unesp

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Rio Claro

Programa de Pós-Graduação em Ciência da Computação

Lucas Pascotti Valem

Contextual Similarity Learning for Image Retrieval and Classification: Applications in Person Re-Identification

Rio Claro - SP

2024

UNIVERSIDADE ESTADUAL PAULISTA
“Júlio de Mesquita Filho”
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Lucas Pascotti Valem

Contextual Similarity Learning for Image Retrieval and Classification: Applications in Person Re-Identification

Tese de Doutorado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos para obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Daniel Carlos Guimarães Pedronette

Rio Claro - SP
2024

V151c

Valem, Lucas Pascotti
Contextual similarity learning for image retrieval and classification: applications in person re-identification / Lucas Pascotti Valem. -- Rio Claro, 2024
261 f. : il., tabs.

Tese (doutorado) - Universidade Estadual Paulista (UNESP), Instituto de Geociências e Ciências Exatas, Rio Claro
Orientador: Daniel Carlos Guimarães Pedronette

1. Ciência da Computação. 2. Inteligência Artificial. 3. Recuperação de Imagens por Conteúdo. 4. Classificação de Imagens. 5. Re-Identificação de Pessoas. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Universidade Estadual Paulista (UNESP), Instituto de Geociências e Ciências Exatas, Rio Claro. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

UNIVERSIDADE ESTADUAL PAULISTA
"Júlio de Mesquita Filho"
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Lucas Pascotti Valem

Contextual Similarity Learning for Image Retrieval and Classification: Applications in Person Re-Identification

Tese de Doutorado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista "Júlio de Mesquita Filho", como parte dos requisitos para obtenção do título de Doutor em Ciência da Computação.

Comissão Examinadora

- Prof. Dr. Daniel Carlos Guimarães Pedronette (Orientador)
  Instituto de Geociências e Ciências Exatas (IGCE)
  Universidade Estadual Paulista - UNESP
- Profa. Dra. Agma Juci Machado Traina
  Instituto de Ciências Matemáticas e de Computação (ICMC)
  Universidade de São Paulo - USP
- Prof. Dr. Hélio Pedrini
  Instituto de Computação (IC)
  Universidade Estadual de Campinas - UNICAMP
- Prof. Dr. João Paulo Papa
  Faculdade de Ciências (FC)
  Universidade Estadual Paulista - UNESP
- Prof. Dr. Wallace Correa de Oliveira Casaca
  Instituto de Biociências, Letras e Ciências Exatas (IBILCE)
  Universidade Estadual Paulista - UNESP

Conceito: Aprovado.
Rio Claro (SP), 28 de junho de 2024.

Acknowledgements

First and foremost, I thank God for the gift of life and health.

My parents and family, for their love and unconditional support.

My advisor, for the guidance, support, and trust.

All the professors, both national and international, who contributed to this research.

The university and all the faculty members of the graduate program.

Friends and colleagues, for the encouragement and support.

The São Paulo Research Foundation (FAPESP), Fulbright, and Petrobras for the financial support.

Resumo

O crescimento exponencial das coleções de imagens produziu um aumento significativo nas aplicações de aprendizado de máquina e recuperação de imagens em diversos cenários. Apesar dos avanços recentes, muitos métodos ainda dependem fortemente de grandes volumes de dados rotulados para treinamento, o que representa um obstáculo importante, uma vez que produzir dados rotulados é geralmente custoso. Para enfrentar esse desafio, várias técnicas foram desenvolvidas. Um aspecto crítico de tais abordagens é definir a similaridade entre imagens de maneira eficaz, o que continua sendo um desafio central em aplicações de recuperação e aprendizado de máquina, tais como classificação. A questão central está intrinsecamente relacionada à forma como a informação é representada e aos métodos usados para comparar essas representações. Uma grande limitação é que a maioria ainda depende de medidas par-a-par e ignoram outras informações significativas presentes na vizinhança que podem ser usadas para melhorar os resultados. Este trabalho foca em melhorar a eficácia da recuperação de imagens por conteúdo visual e tarefas de classificação usando similaridade contextual, indo além das métricas tradicionais par-a-par para explorar as relações entre os elementos. O aprendizado de similaridade contextual é empregado para explorar relações de vizinhança entre os elementos, usando técnicas tais como informações baseadas em ranqueamento, medidas contextuais, grafos e hipergrafos para modelar a informação contextual de forma eficaz. Esta tese propõe sete métodos novos aplicados a cenários de propósito geral e re-identificação de pessoas (Re-ID) abordando diferentes contribuições. Três tarefas principais foram consideradas: estimativa de eficácia de consultas, recuperação e classificação de imagens. Foi realizada uma ampla avaliação experimental, totalizando 17 coleções de imagens e mais de 50 descritores visuais. Os métodos propostos, quando comparados com o estado-da-arte, demonstram resultados que são comparáveis ou superiores aos das abordagens existentes na maioria dos casos.

Palavras-chave: Similaridade Contextual; Recuperação de Imagens; Classificação de Imagens; Estimativas de Eficácia; Re-identificação de Pessoas; Aprendizado de Representações.

Abstract

The exponential growth of image collections has demanded a significant increase in the use of machine learning and image retrieval applications across various scenarios. Despite the relevant advances, many methods still rely heavily on large volumes of labeled data for training, which establishes an important obstacle, once producing labeled data is generally expensive and time-consuming. To address this challenge, numerous techniques have been developed recently. A critical aspect of these approaches is effectively defining image similarity, which remains a central challenge in retrieval and machine learning applications, such as classification. The core of this issue is intrinsically linked to how information is represented and the methods used to compare these representations. A major limitation is that most of them still rely on pairwise measures, ignoring other meaningful information present in the neighborhood that can be used to further increase the results. This work focuses on improving the effectiveness of image retrieval by visual content and classification tasks using contextual similarity, moving beyond traditional pairwise measures to exploit relationships among elements. Contextual similarity learning is employed to capture underlying relationships among elements, using techniques such as rank-based models, contextual measures, graphs, and hypergraphs to model contextual information effectively. This dissertation proposes seven novel methods applied across general-purpose and person re-identification (Re-ID) scenarios addressing different contributions. Three main tasks were considered: query performance prediction, image retrieval, and image classification. A wide experimental evaluation was conducted, totaling 17 datasets and more than 50 visual image descriptors. The proposed methods, when compared with state-of-the-art and recent baselines, demonstrate results that are comparable to or surpass those of existing approaches in most cases.

Keywords: Contextual Similarity Information; Image Retrieval; Image Classification; Query Performance Prediction; Person Re-ID; Representation Learning.

List of Figures

List of Tables

List of Abbreviations and Acronyms

ACC	Color Autocorrelogram Descriptor
ACF	Aggregated Channel Features Detector
AF	Attention Features
AIN	Adaptive Instance Normalization
AIR	Articulation-Invariant Representation Descriptor
ALOI	Amsterdam Library of Object Images
ANML	Adaptive Neighborhood Metric Learning
AP	Average Precision
APPNP	Approximate Personalized Propagation of Neural Predictions
ARMA	Auto-Regressive Moving Average Filter Convolution
ARN	Adaptation and Re-Identification Network
ASC	Aspect Shape Context Descriptor
ATNET	Adaptive Transfer Network
BAS	Beam Angle Statistics Descriptor
BFS	Breadth-First Search
BFSTREE	Breadth-First Search Tree
BIC	Border/Interior Pixel Classification Descriptor
BOVW	Bag of Visual Words
BOW	Bag of Words
CAMEL	Cross-view Asymmetric Metric LEarning
CAP	Camera-aware proxies
CBIR	Content-Based Image Retrieval
CC	Connected Components
CCL	Contextual Contrastive Loss
CCOM	Color Co-Occurrence Matrix Descriptor
CEDD	Color and Edge Directivity Descriptor
CFD	Contour Features Descriptor
CG	Correlation Graph
CIFAR	Canadian Institute For Advanced Research
CLD	Color Layout Descriptor
CMC	Cumulative Matching Characteristics
CNN	Convolutional Neural Networks
COMO	Compact Composite Moment-Based Descriptor
CPRR	Cartesian Product of Ranking References
CPU	Central Processing Unit
CSGLP	Camera Style Generation and Label Propagation
CSRT	Discriminative Correlation Filter with Channel and Spatial Reliability
CUB200	Caltech-UCSD Birds Dataset
CUHK	Dataset from the Chinese University of Hong Kong
DAAM	Domain Adaptive Attention Model
DCNN	Deep Convolutional Neural Networks
DIDAL	Discriminative Identity-Feature Exploring and Differential Aware Learning

DPM	Deformable Parts Model
DPNET	Dual Path Network
DRNE	Deep Rank Noise Estimator
DUKEMTMC	Duke Multi-Tracking Multi-Camera Dataset
EANET	Enhancing Alignment Network
ECN	Exemplar Memory Convolutional Network
EHD	Edge Histogram Descriptor
ELF	Ensemble of Localized Features
EMTL	Enhanced Multi-Dataset Transfer Learning
FBRESNET	Facebook Residual Neural Network
FCTH	Fuzzy Color and Texture Histogram
FN	False Negatives
FOH	Fuzzy Opponent Histogram
FP	False Positives
GAN	Generative Adversarial Network
GAT	Graph Attention Networks
GB	Gigabytes
GBICOV	Covariance descriptor based on Bio-inspired Features
GCN	Graph Convolutional Network
GCN-APPNP	Approximate Personalized Propagation of Neural Predictions GCN
GCN-ARMA	Auto-Regressive Moving Average Filter Convolution GCN
GCN-GAT	Graph Attention Networks GCN
GCN-SGC	Simple Graph Convolution GCN
GDP	Graph Diffusion Process
GIST	Global Image Descriptor for low-dimensional features
GNN	Graph Neural Network
GNN-KNN-LDS	KNN variation of GNN-LDS
GNN-LDS	Learning Discrete Structures for Graph Neural Networks
GOG	Gaussian Of Gaussians Descriptor
GPU	Graphics Processing Unit
GRAD-NET	Graph Diffusion Network
GRID	UnderGround Re-IDentification (GRID) Dataset
GS	Graph Sampling
GSP	Graph Signal Processing
GSSL	Graph-based Semi-Supervised Learning
HACNN	Harmonious Attention Network
HCT	Hierarchical Clustering with Hard-batch Triplet Loss
HHL	Hetero and Homogeneously Learning
HLBP	Histogram of Local Binary Patterns
HQPP	Hypergraph Query Performance Prediction
HRSF	Hypergraph Rank Selection and Fusion
HSV	Color Space: Hue, Saturation, Value
IBN	Instance-Batch Normalization
ICE	Inter-Instance Contrastive Encoding
ICS	Intra-Camera Supervise
ID	Identifier
IDSC	Inner Distance Shape Context
IICS	Intra-inter Camera Similarity

IR	Information Retrieval
ISSDA	Iterative Self-Supervised Domain Adaptation
JCD	Joint Composite Descriptor
JVCT	Joint Generative and Contrastive Learning
KCF	Kernelized Correlation Filters
KISSME	Keep-it-simple-and-straightforward distance learning
KNN	K Nearest Neighbors
LAS	Local Activity Spectrum
LBP	Local Binary Patterns
LCDP	Locally constrained diffusion process
LDA	Linear Discriminant Analysis
LDFV	Local Descriptors encoded by Fisher Vector
LDS-GNN	Learning Discrete Structures for Graph Neural Networks
LGBM	Light Gradient Boosting Machine
LHRR	Log-based Hypergraph of Ranking Reference
LMNN	Large margin nearest neighbor learning
LOMO	Local Maximal Occurrence Descriptor
LS	Label Spreading
LSTM	Long short-term memory
MAM	Memory Access Method
MAP	Mean Average Precision
MAR	MultilAbel Reference Learning
MATE	Multi-Task Multi-Label
MCFS	Muti-cluster Feature Selection
MCRN	Multi-Centroid Representation Network
MGCE-HCL	Multi-Granularity Clustering Ensemble-based Hybrid Contrastive Learning
MGH	Metadata Guided Hypergraph
ML	Machine Learning
MLFN	Multi-Level Factorisation Network
MMCL	Memory-based Multi-label Classification Loss
MOSSE	Minimum Output Sum of Squared Error
MPEG	Moving Picture Experts Group
MR	Manifold Ranking
MSE	Mean Squared Error
MSMT	Multi-Scene Multi-Time Re-ID Dataset
NASNET	Neural Architecture Search Network
NDFS	Non-negative Discriminative Feature Selection
NET	Network
NMF	Non-negative Matrix Factorization
NNCLR	Nearest-Neighbor Contrastive Learning of Visual Representations
NP-HARD	Nondeterministic Polynomial-time Hard
NS	Abbreviation of NS-Score
NS-SCORE	Score for UKBench Dataset
O2CAP	Offline-Online Associated Camera-Aware Proxies
OLDFP	Object Level Deep Feature Pooling
OOD	Out-of-Domain
OPF	Optimum-Path Forest

ORL	Our Database of Faces
OSNET	Omni-Scale Feature Learning Neural Network
OSNET-AIN	OSNET with Adaptive Instance Normalization
OSNET-IBN	OSNET with Instance-Batch Normalization
PAF	Part Association Field
PAUL	Patch-Based Unsupervised Learning Framework
PCA	Principal Component Analysis
PHOG	Pyramidal Histogram of oriented gradients
PIF	Part Intensity Field
PK-SAMPLER	Random Sampling Method in Re-ID
QPP	Query Performance Prediction
RAM	Random Access Memory
RBF	Radial Basis Function Kernel
RBO	Rank-Biased Overlap
RDNN	Residual Dense Neural Network
RDP	Regularized Diffusion Process
RDPAC	Rank Diffusion Process with Assured Convergence
RE-ID	Person Re-Identification
RESNET	Residual neural network
RFE	Rank Flow Embedding
RGB	Red, Green, Blue
RL-SIM	Ranked Lists Similarity Approach
RLCC	Refining Pseudo Labels with Clustering Consensus
RQPPF	Regression for Query Performance Prediction Framework
SCC	Strongly Connected Components
SCD	Scalable Color Descriptor
SCH	Simple Color Histogram
SD	Self-diffusion for Image Segmentation and Clustering
SDC	Scale-invariant Feature Transform Dense Color
SENET	Squeeze-and-Excitation Network
SGC	Simple Graph Convolution
SGD	Stochastic Gradient Descent
SIFT	Scale-invariant Feature Transform
SORT	Simple Online and Realtime Tracking
SOTA	State-of-the-art
SP	Spatial Pyramid
SPACC	Spatial Pyramid Color Autocorrelogram Descriptor
SPCEDD	Spatial Pyramid Color and Edge Directivity Descriptor
SPEC	Spectral Regression
SPFCTH	Spatial Pyramid Fuzzy Color and Texture Histogram
SPGAN	Similarity preserving generative adversarial network
SPJCD	Spatial Pyramid Joint Composite Descriptor
SPLBP	Spatial Pyramid Local Binary Patterns
SS	Segment Saliences
SSL	Softened Similarity Learning Approach
STF	Swin-Transformer
SURF	Speeded-Up Robust Features
SVD	Singular Value Decomposition

SVM	Support Vector Machines
SVR	Support Vector Regression
SWIN-TF	Swin-Transformers
T-SNE	t-Distributed Stochastic Neighbor Embedding
TAUDL	Tracklet Association Unsupervised Deep Learning
TN	True Negatives
TP	True Positives
TPG	Tensor Product Graph
UDA	Unsupervised Data Augmentation
UDLF	Unsupervised Distance Learning Framework
UGAF-RSF	Unsupervised Genetic Algorithm Framework for Rank Selection and Fusion
UKBENCH	University of Kentucky Dataset
USRF	Unsupervised Selective Rank Fusion Method
UTAL	Unsupervised Tracklet Association Learning
VAL-PAT	Framework for Transferable Representations of Pedestrians
VGGNET	Visual Geometry Group Network
VIT	Vision Transformer
VOC	Vocabulary Tree
VRAM	Video Random Access Memory
WHOS	Weighted Histograms of Overlapping Stripes
WSEF	Weakly Supervised Experiments Framework
YOLO	You Only Look Once object detection network

List of Symbols

$A(i)$	Set of all elements in a batch, except the image of index i.
C	Number of virtual classes in the synthetic scenario.
C_n	Set of combinations where each combination is of size n.
E	Set of edges of a graph.
E_h	Set of hyperedges of a hypergraph.
G	A graph.
H	A hypergraph model or the number of feature maps (or hidden units) in the hidden layer of a GCN.
I	The set of indices for all augmented samples in a batch.
L	Size of ranked lists.
M	Confusion matrix of probabilities between classes.
M_c	Confusion matrix of probabilities between elements of the same class.
M_f	Sparse matrix used by RFE to accumulate normalized scores from different rankers.
N	The size of the collection $\mathcal{C}$, i.e., dataset size.
$NN_k(i)$	The set of k nearest neighbors of image i.
$NN_k^{\mathcal{Y}}(i)$	A subset of $NN_k(i)$ containing only images from the same class of image i.
N_b	Number of image pairs in a training batch.
$P(i)$	The set of indices of all positive samples in the batch distinct from image i.
R_i	Ranker of index i.
S	Selection set of all possible combinations of rankers.
S_p	Selection set of pairs of rankers.
T, t	Number of iterations.
V	Set of graph vertices.
V_L	Set of labeled nodes in the graph.
V_U	Unlabeled subset of the node set.
α	Constant for the normalization equation of RFE.
β	Weight or relevance of correlation in the selection measure.
$\boldsymbol{z}_i$	The embedding of the data sample i generated by the metric learning model.
$\circ$	Hadamard (element-wise) product.
δ	A function that computes the distance between two feature vectors.
ϵ	A function that extracts a feature vector from an image.
η_f	Fused affinity measure used for rank aggregation.
$\eta_r(i,x)$	Function that assigns a weight to image x according to its position in τ_i.
γ	An effectiveness estimation measure.
γ_A	Authority effectiveness estimation measure.
γ_R	Reciprocal Density effectiveness estimation measure.
$\hat{\mathbf{A}}$	Normalized adjacency matrix of a graph.

λ	A correlation measure (e.g., RBO).
$\mathbb{R}$	The set of real numbers.
$\mathbf{A}$	Affinity matrix (RFE) or adjacency matrix (GCNs).
$\mathbf{C}$	Similarity measure matrix based on Cartesian product.
$\mathbf{D}$	Distance matrix.
$\mathbf{H_G}$	HRSF Hypergraph model.
$\mathbf{H}$	Incidence matrix for HRSF or matrix encoding the similarity information of h-embeddings for RFE.
$\mathbf{I}$	Identity matrix.
$\mathbf{S}$	Similarity matrix.
$\mathbf{W}$	Affinity matrix (HRSF) or weight matrix in the definition of GCNs.
$\mathbf{X}$	Feature vectors provided as input to the GCN.
$\mathbf{Z}$	Matrix of embeddings learned by the GCN model.
$\mathbf{b}$	Reciprocal neighborhood binary vector used in the computation of RQPPF meta-features.
$\mathbf{b_i}$	Reciprocal neighborhood binary vector for image i.
$\mathbf{c}_i$	Connected component of index i.
$\mathbf{c}_q$	CC-embedding of a connected component q computed by RFE.
$\mathbf{e}_i$	Representation vector (embedding) of the element of index i from the dataset.
$\mathbf{f}$	Contextual Rank-based Feature (Meta-Feature) vector.
$\mathbf{f}_s$	Set of synthetic features used for training the regression model.
$\mathbf{f}_t$	Set of test features used for testing the regression model.
$\mathbf{h}_i$	Row i of matrix $\mathbf{H}$, named h-embedding.
$\mathbf{p}$	Reciprocal rank position vector used in the computation of RQPPF meta-features.
$\mathbf{p_i}$	Reciprocal rank position vector for image i.
$\mathbf{q}$	Effectiveness estimation vector used in the computation of RQPPF meta-features.
$\mathbf{q_i}$	Effectiveness estimation vector for image i.
$\mathbf{x}_i$	Feature vector representing the image o_i, a row of matrix $\mathbf{X}$.
$\mathbf{z}_i$	Row i of matrix $\mathbf{Z}$, embedding representation for the node v_i.
$\mathcal{C}$	Image dataset.
$\mathcal{C}_L$	Set containing the L most similar images to image o_q in the collection $\mathcal{C}$.
$\mathcal{E}_c$	Set of candidate edges defined by RFE.
$\mathcal{L}_{\text{ccl}}$	Proposed contextual contrastive loss.
$\mathcal{L}^{\text{sup}}$	Supervised contrastive loss.
$\mathcal{N}$	Neighborhood set.
$\mathcal{N}(o_q, k)$	Neighborhood set containing the k most similar elements to o_q.
$\mathcal{N}_r(o_q, k)$	Reciprocal neighborhood set for image o_q.
$\mathcal{N}_r$	Reciprocal neighborhood set.
$\mathcal{R}$	Set of rankers.
$\mathcal{S}$	Set of connected components.
$\mathcal{T}$	Set of ranked lists for all the images in the dataset.
$\mathcal{T}_i$	Set of ranked lists produced by ranker R_i.
$\mathcal{T}_j$	The set of ranked lists produced by the ranker R_j.

$\mathcal{X}$	A subset of $\mathcal{C}$.
$\mathcal{Y}$	A set of labels (classes).
$\mathfrak{R}$	Set of rankers provided as input to the method.
$\mathfrak{X}^*$	Selected combination among all sizes.
$\mathfrak{X}^*_n$	Selected combination composed by n rankers.
$\mathfrak{X}_n$	Candidate combination composed by n rankers.
μ	Constant used in RBO correlation measure.
ϕ	Regression model for query performance prediction.
ψ	The Contrastive loss temperature parameter.
ρ	A similarity measure.
σ	Normalization function used in RFE.
τ_n^R	Ordered list of combinations of size n. Also referred to as the selection list.
$\tau_n^R(\mathfrak{X}_n^i)$	Position of the combination $\mathfrak{X}_n^i$ in the selection list τ_n^R.
τ_q	Ranked list of image q.
$\tau_q(i)$	The position of image o_i in the ranked list τ_q.
$\tau_{i,q}$	Ranked list of image of index q calculated by ranker i.
τ_i	Ranked list of image of index i.
$\tau_{q,f}(i)$	Position of image o_i in the ranked list of o_q according to feature f.
$\tau_q(i)$	Position of the image i in the ranked list of query image q.
$\boldsymbol{x}_\ell$	The ℓ-th image in the batch.
$\tilde{\boldsymbol{x}}_\ell$	The ℓ-th augmented image in the batch.
$\boldsymbol{y}_\ell$	The label corresponding to the ℓ-th image.
$\tilde{\boldsymbol{y}}_\ell$	The label corresponding to the ℓ-th augmented image.
$\tilde{\mathbf{A}}$	Adjusted adjacency matrix, $\mathbf{A}+\mathbf{I}$.
$\tilde{\mathbf{D}}$	Degree matrix of $\tilde{\mathbf{A}}$.
$\times$	Multiplication operator.
ξ	The current epoch number.
ξ_{total}	The total number of epochs.
a_{ij}	Entry in the adjacency matrix indicating the presence (1) or absence (0) of an edge between vertices o_i and o_j.
c	Number of classes (or categories).
$c(i,j)$	Element of matrix $\mathbf{C}$.
cp	Pairwise similarity relationship based on Cartesian product.
d	Number of vector dimensions.
d_e	The dimensionality of the RFE embedding space in which each object is represented.
e_i	A hyperedge of index i.
f_g	Function that, given a hypergraph and an incidence matrix, calculates a graph (RFE).
f_h	Function that, given ranked lists, calculates a hypergraph and an incidence matrix by re-ranking through hypergraph embeddings (RFE).
f_m	Manifold learning function that processes a set of ranked lists $\mathcal{T}$.
$f_p(o_q, i)$	Function returning the i-th neighbor of image q.
f_r	Function representing unsupervised similarity learning.
f_s	Function for ranker selection.
f_{gcn}	Function representing the graph convolutional network model.

$h(e_i, v_j)$	Reliance of vertex v_j to belong to a hyperedge e_i.
$h_p(e_q)$	Weight of hyperedge e_q.
h_{ij}	An element of $\mathbf{H}$ representing the similarity of object o_j in the context of the hyperedge e_i.
k	Size of the neighborhood set.
k_d	A variable representing a specific depth for computing a correlation measure.
k_{start}	The initial value of k for the first epoch.
k_v	Size of virtual classes for synthetic data.
m	Number of features, i.e., size of the set $\mathfrak{R}$.
n	Size of a combination.
n_k	Number of candidate edges for RFE graph.
o_i	Indicates any object (element) belonging to the dataset, whose index is i.
obj_i	Object of index i, often abbreviated as o_i.
p	Pairwise relationship function defined by RFE.
p_x	Pixel of position x in a grayscale image.
s_c	RFE Similarity measure attributed to pairs based on the similarity between h-embeddings and confidence of the hyperedge.
t_c	Threshold for edge computation in the connected components stage of the RFE.
th_{end}	Final threshold of the Correlation Graph.
th_{inc}	Correlation Graph threshold increment.
th_{start}	Initial threshold of the Correlation Graph.
v_i	A node in the node set V representing an image o_i.
v_l	A labeled node.
w	Selection measure for combinations of rankers.
$w(e_i)$	A positive weight assigned to a hyperedge e_i.
$w_p(i, x)$	A weight function that assigns relevance to a vertex o_x based on its position in a ranked list.
w_p	Selection measure for pairs of rankers proposed by HRSF.
$\mathcal{T}_h{}^{(T)}$	Set of ranked lists after T iterations of RFE.
y_i	Label (class) of object o_i, i.e., i-th row of $\mathcal{Y}$.

Contents

1 Introduction

Effectively defining the similarity between images is a central challenge in retrieval and machine learning applications. This issue is deeply connected to: *(i)*: how information is represented; and *(ii)*: the measures used to compare these representations [321, 294, 371, 250]. This work presents contributions in both directions, proposing seven novel approaches.

This dissertation discusses and presents contributions aimed at improving the effectiveness of image retrieval by visual content and classification tasks using contextual similarity learning. This introductory chapter outlines an overview of this work and is organized as follows: Section 1.1 discusses the motivations of the conducted research. Section 1.2 presents the challenges and research questions addressed. Section 1.3 states the main hypothesis validated in this dissertation. Section 1.4 discusses the objectives and contributions of the study. Section 1.5 describes the overall structure of this document, including a summary of each chapter's content and an illustration of how concepts and terms relate to the contributions.

1.1 Motivation

In recent years, there has been an exponential increase in the volume of image data, primarily due to advancements in technologies for generating, storing, and sharing visual information [320, 80]. Additionally, there are numerous applications (e.g., surveillance cameras [390, 137, 426], medical imaging [341, 6, 1], remote sensing systems [160], social media [140]) that generate vast amounts of visual data.

In this scenario, image retrieval and machine learning tasks such as image classification are increasingly being utilized in many applications [255]. Remarkable progress has been made in these methods, particularly due to the consistent evolution of deep learning [109, 31]. However, the majority of them are supervised and depend on large volumes of labeled data for training. In contrast, the production of labeled data is challenging since it is often expensive and time-consuming to obtain [91]. It may also require a specialist for labeling, especially according to the specificity of the domain. Aiming at filling this gap, many unsupervised, semi-supervised, and even self-supervised approaches have been proposed to deal with such a challenge [107]. In most of these methods, effectively modeling data is crucial for exploiting the information available in the unlabeled data.

For most approaches, the essence of learning hinges on the ability to model data accurately, which involves different concepts, in particular, representation approaches and distance or similarity measures [321]. This is especially important for Content-Based Image Retrieval (CBIR) systems, which retrieve images based on visual content rather than metadata [316]. These systems usually employ feature extraction and representation methods, which have evolved considerably [294]. Such methods have transitioned from traditional hand-crafted features [254] to more advanced deep learning approaches [294, 438], including Vision Transformers [77, 202]. However, most comparison tasks still rely on pairwise measures [294, 80], which do not exploit contextual information [245]. In general, the term context can be broadly understood as all the relevant information pertinent to an application and its users. This work considers the idea of *contextual similarity* that consists of exploiting the relationships beyond pairwise analysis, involving other elements, such as the neighborhood or more related additional information [246, 245]. The term *contextual similarity learning* is used for the learning process that employs contextual similarity for more effectively capturing the underlying relationships among elements.

An essential aspect is that contextual similarity information can be modeled in many different forms, using different representations and structures [235, 416], among them: *(i)* graphs [86, 366]: they can be used for exploiting the relationships between neighbors, which is a key aspect for understanding the local context and influence among interconnected entities; *(ii)* ranked lists [246, 245]: in image retrieval, each ranked list contains the most similar elements for a given query. The similarities between elements can be redefined according to the analysis of the neighborhoods available in these lists. The position of each element in each list also contains valuable information; *(iii)* clustering [404, 373]: identify and group data points that are similar according to predefined criteria. These groups allow the discovery of inherent patterns or relationships that may not be apparent upon initial observation.

Besides these approaches, there is still a wide range of methodologies that can be proposed to exploit contextual information in numerous scenarios. Similarity learning applied to retrieval is generally explored by re-ranking tasks. Despite the crescent popularity of these methods, more robust structures have not yet been extensively employed in most cases. Structures that represent higher-order similarities, such as relationships among neighbors of neighbors, can be particularly advantageous. Hypergraphs, for example, allow edges to connect multiple vertices, offering a sophisticated technique for capturing these relationships [403, 251]. Additionally, most unsupervised re-ranking approaches [18, 282, 108] provide a new ranked list representation as output, but new features are not produced in return, which could be used to encode contextual information for classifiers, for example.

Another application that could deeply take advantage of the enrichment of contextual information is feature selection and fusion [260, 389, 424, 329, 327]. Among different strategies, the selection of features can be done through effectiveness estimation and correlation measures [329]. They consider the idea that fusion benefits from elements with high effectiveness and that are also complementary. The creation and usage of contextual structures for estimating the effectiveness and measuring correlations is still an area that requires further research.

This work exploits contextual similarity learning for general-purpose image retrieval and person re-identification, usually abbreviated as person Re-ID. Person Re-ID is a type of surveillance application that has been gaining a lot of attention and nowadays is of fundamental importance in many camera surveillance systems. The task consists of identifying individuals across multiple cameras that have no overlapping views [137]. A Re-ID system broadly consists of three main steps [426]: person detection, feature extraction, and person retrieval or matching. This work focuses on the final step, which can be viewed as a specific image retrieval application [137].

Person Re-ID is a complex task that presents numerous difficulties [390, 137, 426], including *(i)* varying angles of view between cameras, *(ii)* low-resolution images, *(iii)* changes in lighting conditions, *(iv)* occlusions blocking part of the view, *(v)* the difficulty of manually labeling images for use in training algorithms, *(vi)* unbalanced classes or classes with very few elements, *(vii)* the complexity of modeling data, and *(viii)* the extensive volume of data that needs to be processed.

Amid these challenges in Re-ID, many approaches introduced more robust deep learning models [390], such as Vision Transformers [111, 163, 221], metric learning [185, 156, 185, 396], and Siamese networks [340, 339]. Other strategies include dataset expansion with augmentations [132, 291] or artificial data considering appearance attributes, body parts, temporal information, and different types of multimodal information. Additionally, metric learning is often employed for Re-ID due to its capacity to be effective when dealing with unseen data [185] since it focuses on learning distances or similarities rather than specific features of the training data. This approach allows the model to generalize better to new examples that were not present in the training set.

Apart from all these advancements, post-processing methods that exploit contextual information have gained significant attention due to their ability to improve results provided by latent features of different deep learning models. Various unsupervised post-processing strategies are based on the idea of exploiting the information of reciprocal neighborhoods and measuring the co-occurrence of elements in ranked lists [429, 174, 225, 165, 211, 96, 388, 95, 108], demonstrating substantial improvements. Although these approaches are becoming increasingly common in Re-ID, methods for selection and fusion remain relatively scarce. This scenario highlights the importance of investigating methods capable of effectively exploiting contextual information.

In addition to retrieval, it is also imperative to address scenarios with limited labeled data for classification [135]. Graph convolutional networks (GCNs) offer a promising solution for semi-supervised classification by learning from both labeled and unlabeled data considering graph structures [412]. Moreover, GCNs can learn node and graph embeddings that capture complex dependencies and structural relationships [141]. However, GCNs are not widely used for image classification since graphs are typically not available in image domains [274, 343, 307]. Therefore, effectively modeling these graphs, which can be utilized to exploit contextual information, is a crucial topic for research.

Another approach that has recently demonstrated continuous advances for improving classification results is the use of contrastive learning [143, 47]. Unlike the commonly used cross-entropy loss, which aims to minimize the difference between the predicted and true class probabilities, contrastive loss focuses on learning similarities and dissimilarities between data points rather than merely categorizing them [47]. Despite this, most contrastive losses consider only pairwise measures [143, 47, 49, 47], with only a few incorporating some type of neighborhood information [441, 82, 183]. Moreover, these approaches often require huge volumes of data for training (i.e., labeled or unlabeled) [47, 49], even in self-supervised scenarios, which is a challenge in circumstances where data is scarce.

In light of the presented discussion and all challenges, the focus of this dissertation is to exploit the use of contextual similarity information with the objective of improving the effectiveness of image retrieval and classification, particularly in cases where labeled data is limited or non-existent. This dissertation primarily concentrates on unsupervised learning, while also proposing semi-supervised and supervised approaches.

1.2 Research Challenges

Contextual similarity information can be applied in a variety of fields. However, appropriately representing and exploiting contextual information in each scenario poses significant difficulties. There are many research challenges related to various applications that can be used to improve the effectiveness of image retrieval and classification tasks. In the following, several topics are discussed and corresponding research questions are presented for each:

- **Selection and fusion in person Re-ID:** The selection and fusion involves choosing the most relevant features from the data and combining them to enhance the retrieval effectiveness [260]. There are various feature extractors and possible combinations between them. Selecting the right features is crucial because manually evaluating all combinations becomes impractical as the number of features increases linearly and the number of combinations increases exponentially. The concept is based on the idea that fusion is most effective when it involves elements that are both highly efficient and complementary. For person Re-ID, the complexity of accurately matching

individuals across different camera views becomes significantly more challenging in unsupervised applications due to the absence of labeled data [137]. Effectively modeling and exploiting patterns in the data is crucial in this scenario.
Research question:

  - *How can contextual similarity information be used for selection and fusion in unsupervised person Re-ID?*

- **Query performance prediction:** Also known as effectiveness estimation, query performance prediction (QPP) encompasses techniques for assessing the quality of ranked lists in scenarios where no labels are provided. In this context, the ability to assess the effectiveness of the retrieval process provides a significant advantage for different tasks, including enabling the selection of more effective ranked lists. However, QPP is very challenging, especially in unsupervised tasks. One of the main difficulties is elaborating an approach that effectively generalizes across diverse scenarios [262]. Bridging this gap represents a major challenge that can be mitigated by incorporating contextual similarity information.
Research question:

  - *How can data be modeled using contextual similarity information for query performance prediction?*

- **Synthetic data:** Recently, self-supervised approaches have been proposed to address scenarios where labeled data is scarce. Among the different means of self-supervision, one of them is by using synthetic data. There are many advantages and benefits of using synthetic data, primarily due to its flexibility and control in generating large volumes of annotated data. In domains where safety and privacy are relevant, using real data can raise privacy concerns and legal issues. Synthetic data does not carry these risks. However, creating representative synthetic data presents many difficulties. One of the primary challenges is to accurately reflect the complexity and variability of real-world data [72]. The generated synthetic data is expected to encompass a wide range of scenarios, including rare events and edge cases, to ensure comprehensive learning.
Research questions:

  - *How can contextual similarity information be used to generate synthetic data?*
  - *How can contextual similarity learning be employed on synthetically generated data?*

- **Unsupervised similarity learning methods:** Despite the potential of unsupervised similarity learning methods to improve retrieval results, effectively representing and encoding the maximum amount of contextual information remains

a challenge. This difficulty is amplified because these methods operate without labels and cannot utilize relevance feedback [312] as supervised algorithms do. These methods usually exploit the relationships among images through ranked lists and similarity among elements [95, 108, 250]. The primary challenge lies in modeling and leveraging this similarity information, which can be approached through various strategies such as graph structures [381], contextual measures [128], and more [382, 384]. Utilizing more complex structures to represent second-order similarity (i.e., relationships such as neighbors of neighbors) can be particularly relevant, for example.
Research question:

  - *How can more complex structures, which encode contextual information more effectively, be applied to unsupervised similarity learning?*

- **Representation learning and embeddings:** Feature learning is of fundamental importance in many retrieval and classification applications [321]. However, the capacity to encode information of an image in an embedding is very challenging. When converting an image into an embedding, some information is inevitably lost. This loss must be minimized to ensure that the most critical features of the image are retained. There is also the semantic gap [24, 115] between the raw pixel data of an image and the human interpretation of the image's content. Unsupervised similarity learning approaches usually post-process ranked lists to enhance image retrieval results but do not provide any form of embeddings that can be used for other tasks, such as classification.
Research question:

  - *How can contextual information from similarity learning approaches be encoded to generate embeddings that are useful for tasks beyond retrieval, such as classification?*

- **Contextual similarity and Graph Convolutional Networks (GCNs):** The GCNs effectively capture relationships and interactions within complex networks, enhancing results in tasks involving structured data. However, graphs are not inherently available for most image datasets, and GCNs heavily rely on these structures to deliver significant results [141, 307]. The main challenge involves accurately modeling the graph for effective use by the GCN.
Research question:

  - *How can contextual similarity information be incorporated into the input graph utilized by Graph Convolutional Networks (GCNs) and improve their classification results?*

- **Correlation measures and manifold learning:** Manifold learning is a technique for uncovering simpler, underlying structures in complex high-dimensional data [133]. Correlation measures quantify the similarity between data points, which is very useful to model relationships in the data. However, this is challenging since data can be complex and heterogeneous, involving multiple variables with nonlinear relationships that are difficult to capture [16]. Also, outliers may present significant challenges in data analysis.
  Research questions:
  - *Can rank-based information be utilized to measure the correlation between images more effectively?*
  - *Can a correlation measure be proposed and applied to enhance image retrieval with manifold learning?*
- **Contrastive learning**: It has been extensively used in self-supervised and supervised learning due to its effectiveness in learning representations that distinguish between similar and dissimilar images. It offers an alternative to cross-entropy by yielding more semantically meaningful image embeddings. However, most contrastive losses rely on pairwise measures to assess the similarity between elements [143, 47], ignoring more general neighborhood information that can be leveraged to enhance model robustness and generalization [441].
  Research question:
  - *How can contextual similarity information be incorporated into metric learning, including its direct integration into losses such as contrastive loss?*

The contributions presented and discussed in this work address those important research challenges.

1.3 Dissertation Statement

Driven by the challenges identified in the literature, primarily the difficulty of obtaining a large amount of labeled data and the increasing need for methods that exploit contextual information, we explore the application of contextual similarity learning in different scenarios. The main hypothesis of the work is briefly stated as follows:

> *Contextual similarity learning can improve the effectiveness of image retrieval and classification tasks across general-purpose and person re-identification (Re-ID) applications. This concept is applicable to unsupervised, semi-supervised, and supervised approaches, particularly in contexts where labeled data is limited.*

The hypothesis is validated by the proposed approaches and a comprehensive experimental evaluation presented in this dissertation.

1.4 Goals and Contributions

The general objective of this work is to investigate and propose new approaches that utilize contextual similarity information to improve the effectiveness of image retrieval and classification, applying it to general-purpose scenarios and person re-identification. Figure 1.1 outlines the goals and contributions and their relationships with each approach.

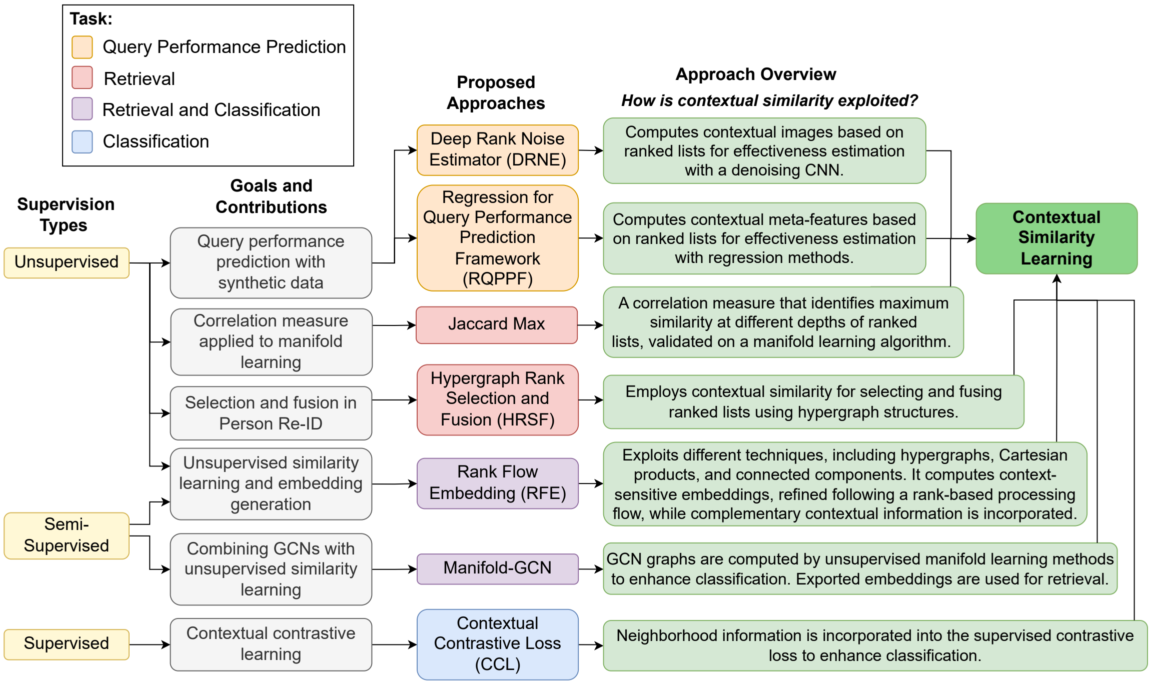

Task:
Query Performance Prediction
Retrieval
Retrieval and Classification
Classification
Supervision Types
Goals and Contributions
Proposed Approaches
Approach Overview
How is contextual similarity exploited?
Unsupervised
Query performance prediction with synthetic data
Deep Rank Noise Estimator (DRNE)
Computes contextual images based on ranked lists for effectiveness estimation with a denoising CNN.
Regression for Query Performance Prediction Framework (RQPPF)
Computes contextual meta-features based on ranked lists for effectiveness estimation with regression methods.
Contextual Similarity Learning
Correlation measure applied to manifold learning
Jaccard Max
A correlation measure that identifies maximum similarity at different depths of ranked lists, validated on a manifold learning algorithm.
Selection and fusion in Person Re-ID
Hypergraph Rank Selection and Fusion (HRSF)
Employs contextual similarity for selecting and fusing ranked lists using hypergraph structures.
Unsupervised similarity learning and embedding generation
Rank Flow Embedding (RFE)
Exploits different techniques, including hypergraphs, Cartesian products, and connected components. It computes context-sensitive embeddings, refined following a rank-based processing flow, while complementary contextual information is incorporated.
Semi-Supervised
Combining GCNs with unsupervised similarity learning
Manifold-GCN
GCN graphs are computed by unsupervised manifold learning methods to enhance classification. Exported embeddings are used for retrieval.
Supervised
Contextual contrastive learning
Contextual Contrastive Loss (CCL)
Neighborhood information is incorporated into the supervised contrastive loss to enhance classification.

Figure 1.1 – Overview of goals and contributions and how contextual similarity is exploited.

Notice that the proposed approaches were investigated to address the research challenges previously discussed. Each method exploits contextual similarity differently. In the following, an overview of each of the goals and contributions (shown in gray in the diagram) is presented:

- **Query performance prediction with synthetic data:** The objective is to predict the quality of ranked lists generated by CBIR systems without labeled data. Two self-supervised methods were proposed; both are trained using synthetic data and utilize the same algorithm for generating this data, but differ in their strategy:
  - **Based on denoising with Convolutional Neural Network (CNN):** The Deep Rank Noise Estimator (DRNE) proposes a new model architecture

for image denoising to perform query performance prediction. The idea is to interpret the incorrectness of a ranked list as noise in a *contextual image*.

  - **Based on regression and feature modeling:** The Regression for Query Performance Prediction Framework (RQPPF) supports diverse features and regression models. It computes *meta-features*, that encode reciprocal neighborhood information, based on unsupervised measures.

- **Correlation measure applied to manifold learning:** Effectively measuring similarity among data samples represented as points in high-dimensional spaces remains a major challenge. A rank correlation measure, the Jaccard Max, robust to such variations is a contribution of this study. The proposed measure is suitable for diverse scenarios and is validated on an unsupervised manifold learning algorithm based on the Correlation Graph (CG) approach.

- **Selection and fusion in person Re-ID:** This contribution addresses the challenging task of unsupervised selection and fusion of different features for more effective person re-identification. A novel Hypergraph Rank Selection and Fusion (HRSF) framework is proposed, which combines an unsupervised rank-based formulation for feature selection with a robust hypergraph model for query performance prediction and rank aggregation based on manifold learning.

- **Unsupervised similarity learning and embedding generation:** A novel manifold learning algorithm named Rank Flow Embedding (RFE) for unsupervised and semi-supervised scenarios. The proposed method is based on ideas recently exploited by manifold learning approaches, which include hypergraphs, Cartesian products, and connected components. The algorithm computes context-sensitive embeddings, which are refined following a rank-based processing flow, while complementary contextual information is incorporated. The generated embeddings can be exploited for more effective unsupervised retrieval or semi-supervised classification based on Graph Convolutional Networks.

- **Combining GCNs and unsupervised similarity learning:** A novel approach, the Manifold-GCN, based on GCNs for semi-supervised image classification. The main objective is to use manifold learning to model the graph structure to further improve the GCN classification. All manifold learning algorithms employed are completely unsupervised, which is especially useful for scenarios where the availability of labeled data is a concern. This method is also evaluated for person Re-ID by utilizing the embeddings exported by the GCNs.

- **Contextual contrastive learning:** Besides the promising results obtained by contrastive learning approaches, they often consider pairwise measures. The proposed Contextual Contrastive Loss (CCL) replaces pairwise image comparison by introducing a new contextual similarity measure using neighboring elements. The CCL yields a more semantically meaningful image embedding ensuring better separability of classes in the latent space. Although supervised, the results show that the CCL provides higher gains in cases with fewer labeled data.

1.5 Organization

The text is structured around the main contributions of this research, which have been either published or submitted to international conferences and journals. To facilitate understanding, the chapters are organized according to the order in Figure 1.1, based on the type of supervision and task. In the following, a brief overview of each chapter is provided:

- **Chapter 2 - Background:** describes the main concepts related to this dissertation, including an overview of each topic and the formal definitions.
- **Chapter 3 - Related Work:** discusses the related work relevant to each topic presented in this study.
- **Chapter 4 - Experimental Protocol:** presents the evaluation measures, datasets, and descriptors considered in the evaluation for both general-purpose and person re-identification scenarios.
- **Chapter 5 - Self-Supervised Contextual Effectiveness Estimation Measures:** presents two approaches for query performance prediction trained on synthetic data that use contextual representations (i.e., *contextual images* and *contextual meta-features*) to encode information from the ranked lists. The content of this chapter can be found in papers published in the proceedings of the *International Conference on Multimedia Retrieval* (ICMR 2021) [330] and in the proceedings of the *Sixth International Conference on Artificial Intelligence and Knowledge Engineering* (AIKE 2023) [336].
- **Chapter 6 - Rank Correlation Measures for Manifold Learning on Image Retrieval:** presents the Jaccard Max correlation measure, which enhances unsupervised manifold learning results on image retrieval. The content of this chapter is available in a paper published in the proceedings of the *International Conference on Image Processing* (ICIP 2022) [324].
- **Chapter 7 - Hypergraph Rank Selection and Fusion (HRSF):** presents the HRSF, an unsupervised approach for selecting and fusing different ranked lists using hypergraph structures. The content of this chapter can be found in a paper published in the journal of *Image and Vision Computing* (IVC 2022) [331].

- **Chapter 8 - Rank Flow Embedding (RFE):** presents an approach that performs unsupervised similarity learning to improve results on image retrieval. Different techniques are exploited, including hypergraphs, Cartesian products, and connected components on graphs. Beyond retrieval, it also generates embeddings to enhance semi-supervised classification with GCNs. The content of this chapter can be found in a paper published in the journal *Transactions on Image Processing* (TIP 2023) [334].
- **Chapter 9 - Contextual Manifold Learning on Graph Convolutional Networks (Manifold-GCN):** describes a method for computing graphs through manifold learning, which serves as input for Graph Convolutional Networks (GCNs). This approach improves the results of semi-supervised classification tasks and is also employed for retrieval in person re-identification. The content of this chapter can be found in a paper published in the journal of *Computer Vision and Image Understanding* (CVIU 2023) [333].
- **Chapter 10 - Contextual Contrastive Loss (CCL):** describes a contrastive loss that replaces pairwise image comparison by introducing a new contextual similarity measure using neighboring elements. The approach is validated on supervised classification, providing higher gains especially when fewer labeled data are provided. The content of this chapter can be found in an article submitted to the *19th International Symposium on Visual Computing* (ISVC 2024) [325].
- **Chapter 11 - Conclusions:** presents a discussion of the results, and an overview of all collaborations, publications, submissions, and contributions. It also discusses potential extensions and directions for future work.

Figure 1.2 presents the overall organization of this dissertation, highlighting the main concepts, contributions, and their relationships. It illustrates the contributions presented in each chapter and their associations. The legend in the upper right corner explains the meaning of each color. The diagram is designed for flexible navigation; readers may begin at any chapter node, marked in light blue, and trace the edges to discover how they connect to this work's key concepts and terms. Notice that all the proposed approaches, in red, are interconnected to different concepts and terms (shown in gray) that are always related to the *contextual similarity learning* in green. All of them were published or submitted to international conferences and journals, as indicated by the yellow and orange nodes, respectively.

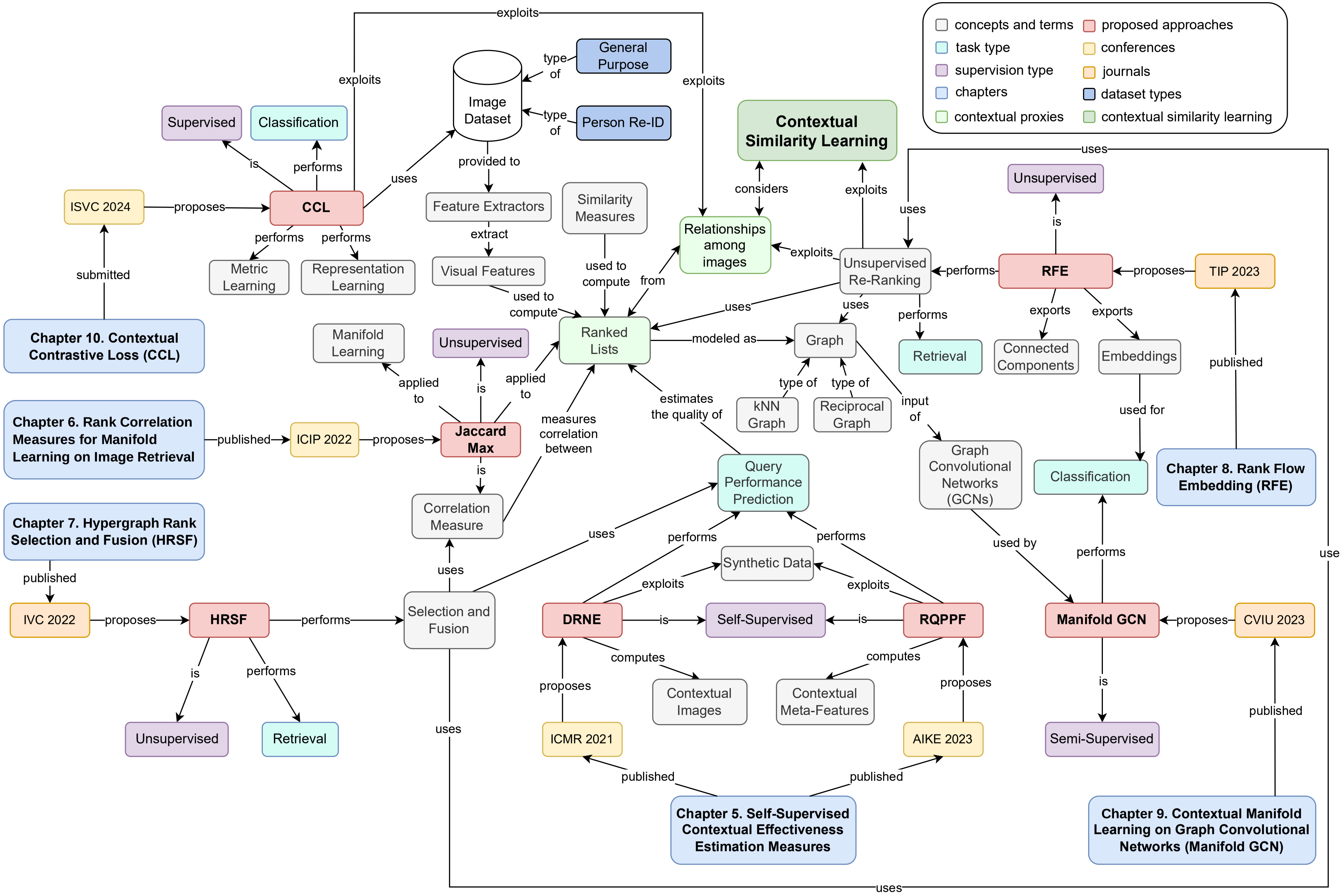


Figure 1.2 – Dissertation structure: organization, main concepts, proposed approaches, and publications.

2 Background

This chapter discusses the main concepts and definitions required for a straightforward understanding of this work, especially the methods presented in subsequent chapters. Section 2.1 introduces definitions of machine learning and the categories of supervision. Section 2.2 defines information retrieval and describes a content-based image retrieval (CBIR) system and its main steps. Section 2.3 discusses feature selection and fusion in the context of image retrieval. Section 2.4 outlines the process of person re-identification and its workflow. Section 2.5 defines the concepts related to graph convolutional networks and semi-supervised classification. Section 2.6 defines hypergraphs and discusses their potential in modeling high-order relationships between elements.

2.1 Machine Learning and Categories of Supervision

Machine learning (ML) is a rapidly evolving technology with the potential to significantly impact society in various ways [287], with multiple applications [270], especially as deep learning continues to advance [255]. A widely accepted and general definition of machine learning is:

> “Machine learning enables computers to learn from data and make decisions without explicit programming. It involves algorithms and statistical models that computer systems use to perform specific tasks by relying on patterns and inference instead of direct instructions.” [14]

An important aspect of machine learning is the use of labeled data. Labeled data refers to any set of data that has been annotated with one or more classes to describe certain characteristics relevant to the data. In other words, labels in the dataset can be understood as the correct responses, instructing the learning model on the associations it needs to learn. Unlabeled data, on the other hand, consists of data points without any corresponding labels, requiring the model to find patterns and structures within the data. The type of supervision defines how this data is used in the training process. There are three main broad categories of supervision [14, 255], summarized as follows:

- **Supervised learning:** Learns from labeled data and makes decisions for unlabeled data. The majority of methods belong to this category.
- **Unsupervised learning:** Does not require any labeled data or user intervention.
- **Semi-supervised learning:** Learns using both labeled and unlabeled data.

Each type of supervision has its subcategories. For example, self-supervised learning [107] is a subcategory of unsupervised learning. In self-supervised learning, the system generates its own labels for training. Among the possibilities to generate these labels, can be mentioned: *(i)* using synthetic data; *(ii)* learning one part of the input from another part of the input, as data within the same input (e.g., an image) can be interpreted as belonging to the same class.

This study considers the three broad types of supervision but mainly focuses on unsupervised learning. All the proposed retrieval approaches are unsupervised, with only minor exceptions discussed in the text. The approaches proposed for query performance prediction train themselves on their generated synthetic data, being completely unsupervised. The proposed classification methods are mostly semi-supervised, with only Contextual Contrastive Learning (CCL) being supervised. However, even in semi-supervised and supervised scenarios, the proposed approaches present advantages in scenarios with few labeled data, which is one of the key points of this dissertation.

2.2 Content-Based Image Retrieval (CBIR)

Content-Based Image Retrieval (CBIR) [80, 294, 438] is a specialized area within the broader field of Information Retrieval (IR) [316]. IR encompasses various approaches in Computer Science aimed at indexing and searching for information. It can be defined as:

> "Information Retrieval deals with the representation, storage, organization, and access to information items such as documents, web pages, structured and semi-structured records, multimedia objects, etc. The representation and organization of information items should provide users with easy access to information of their interest." [15]

With the high volume of image data available due to the evolution of technologies to store and retrieve this content, searching for images in a database is now a crucial and imperative task [320, 80, 60]. Originally, most information retrieval systems employed image search based only on textual metadata and keywords [260, 284]. However, searching based on keywords presents many limitations. Among them, it may lead to ambiguities, that do not exist when visual content is considered, for example. Therefore, Content-Based Image Retrieval (CBIR) [284] systems were proposed to compare and retrieve images based on their visual information.

In these systems, images are commonly represented as feature vectors [80], which are numerical representations of their characteristics. These vectors can capture key visual attributes such as color [119], texture [308], and shape [103], enabling the comparison and matching of images within a database. The process of extracting these features is done by a method known as a descriptor, which consists of a feature extractor and a distance measure [316]. Initially, most CBIR systems utilized manually designed (i.e., hand-crafted) descriptors. However, modern descriptors predominantly rely on deep learning techniques [294].

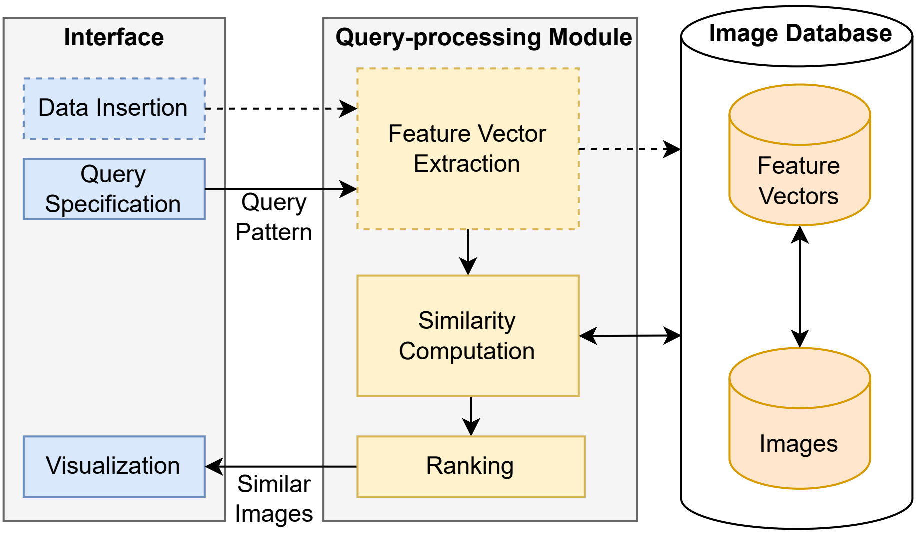


Figure 2.1 – Typical architecture of a CBIR system. Figure adapted from [316].

Using these extracted features, CBIR systems store them in a database. When a new image query is submitted, features are extracted from the query and compared to those in the database. The system then returns a ranked list of images, sorted in descending order of similarity to the query image. Figure 2.1 illustrates a typical architecture of CBIR system [316] and its main steps. There are three main modules:

- **Interface:** This module is visible to the user and is used to submit a query image and visualize the results. The results are presented as a ranked list, with the most similar images to the query image appearing at the top.
- **Query-processing module:** It involves all the steps related to processing the query and returning the most similar images. The first step is to extract the features from the image provided. These features are then compared to those in the database. Various distance or similarity measures can be used in this process, with Euclidean distance and cosine similarity being the most common. It can also involve the use of Memory Access Methods (MAMs) to efficiently access and retrieve relevant data from memory, especially when dealing with large databases. After computing these measures, the images are ranked according to the similarity or distance values. The ranked lists are then returned to the interface module, which displays the results to the user.

- **Image Database:** The database includes all the images along with their respective stored features, eliminating the need to recompute the features each time they are accessed. If more than one descriptor is considered, the database may need to store more than one feature per image. This module communicates with the query-processing component when requested.

Over the last decade, various approaches have been proposed to improve the effectiveness of feature extractors [260, 77, 202]. Section 2.2.1 further discusses feature extraction, the types of descriptors, and the process of obtaining ranked lists through distance measures. In addition to the continuous evolution and advancements in feature extraction and deep learning techniques, unsupervised similarity learning has significantly improved the ranking results in various scenarios by post-processing the ranked lists [240, 96]. Section 2.2.2 defines and discusses these methods, which are among the topics investigated in this work. Section 2.2.3 presents the formal definition and notations related to CBIR which are relevant to this study.

2.2.1 Feature Extraction and Ranking

In image retrieval systems, raw images are typically not directly used due to their high dimensionality and the possible presence of redundant information they often contain, which can significantly decrease the performance of the retrieval process and even make it less effective. Instead, extractors are employed to compute feature vectors [60], which are derived from raw images and are designed to concisely represent the visual content through essential attributes (e.g., shape, color, texture). For instance, a 256x256 image, which has 65,536 pixels, can be expressed as a single vector of approximately 1,000 positions. The size of the vector varies depending on the chosen approach [254].

Therefore, effectively describing images as feature vectors is a crucial task. However, it is also very challenging since images are inherently complex and may provide a wide variety of information depending on the context. When extracting features, the representation of data into a vector can result in the omission or loss of relevant information, for example. The main challenge is known as the *semantic gap* [24, 115] which refers to the discrepancy between the low-level visual features and the high-level semantic meanings that users actually perceive and are interested in. For example, a computer analyzing an image of a picnic can identify objects and perhaps categorize the scene, but it lacks the ability to grasp the emotional resonance, social interactions, or cultural significance that a human might immediately recognize.

In this scenario, multiple feature extraction methods are available [294, 260]. An effective feature extractor should provide discriminative features. This means it should capture the crucial characteristics of the data that not only encapsulate its essence but also sharply differentiate between various classes or categories in the latent space. The term latent space refers to an abstract space where the intrinsic properties and relationships of data are represented.

The most traditional extractors, known as hand-crafted approaches consist of manually designed methods based on human understanding and intuition about what constitutes important and distinctive features in images for a given scenario. These often involve specific algorithms to detect relevant visual aspects and attributes, such as colors, textures, shapes, and key points of interest. They can be broadly categorized into three main categories depending on how they process the input images:

- **Global:** They consider a holistic view, processing the entire image to capture overall patterns and structures. This approach contrasts with local feature analysis, which focuses on specific parts or details of the image. By evaluating global features, algorithms can encode visual attributes, such as the overall shape, color distribution, and texture. An example of how a global color feature vector is computed can be understood as a process where color histograms are created by counting the number of pixels within an image that corresponds to each of several color bins, summarizing the distribution of colors present in the entire image. A similar process can be applied to texture and shape as well. Some examples of methods in this category are Color Autocorrelogram (ACC) [119], Segment Saliences (SS) [317], Local Binary Patterns (LBP) [187], and Histogram of Oriented Gradients (HoG) [59].

- **Local:** Local features in image analysis refer to distinct elements within an image that capture important information about specific regions rather than the entire image. Analyzing local features involves extracting key points and using techniques to encode their characteristics, allowing for more precise and efficient comparison and manipulation of images. One of the most well-known methods for detecting and describing local features is the Scale-Invariant Feature Transform (SIFT) [205, 427]. It identifies key points in an image that are invariant to scale and rotation, making it highly effective for matching features across different images. Local descriptors are often used in conjunction with a Bag of Visual Words (BoVW) model to generate mid-level representation features [3]. The BoVW model represents an image as a collection of local feature descriptors, which are then quantized into a finite number of visual words. To create a histogram for the feature vector, each local feature descriptor is assigned to the nearest visual word in the vocabulary. The histogram is then computed by counting the frequency of each visual word in the image, resulting in a feature vector.

- **Deep Learning:** Due to continuous advancements in deep learning, most features are now extracted using these methods rather than hand-crafted algorithms. Some approaches even combine both strategies [222]. These models are commonly trained on extensive datasets like ImageNet [70] to ensure more robust generalization. Rather than making predictions, the output from the last fully connected layer of a trained model can be used as feature representations. This layer is often preferred because it provides a compact, high-level representation and is close to the final output. Such a process can be applied to different deep learning methods, including Convolutional Neural Networks (CNNs) [110] and Transformers [77, 202].

After computing the features, the distance between them is calculated to obtain ranked lists. Table 2.1 presents a list of some traditional pairwise distance measures commonly used in this process along with their equations, where δ denotes a distance function. The variables $\mathbf{x}_1$ and $\mathbf{x}_2$ represent two feature vectors in a multidimensional space with d dimensions, where d is also the size of the vectors. Here, $\mathbf{x}(i)$ denotes these i-th component of vector $\mathbf{x}$. Once distances are computed, the resulting values are sorted to generate ranked lists.

Table 2.1 – Examples of traditional distance measures.

Distance	**Equation**
Chebyshev	$\delta(\mathbf{x}_1, \mathbf{x}_2) = \max_{i=1}^{d} \lvert \mathbf{x}_1(i) - \mathbf{x}_2(i) \rvert$
Cosine	$\delta(\mathbf{x}_1, \mathbf{x}_2) = 1 - \frac{\sum_{i=1}^{d} \mathbf{x}_1(i)\mathbf{x}_2(i)}{\sqrt{\sum_{i=1}^{d} \mathbf{x}_1(i)^2}\sqrt{\sum_{i=1}^{d} \mathbf{x}_2(i)^2}}$
Euclidean	$\delta(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{i=1}^{d} (\mathbf{x}_1(i) - \mathbf{x}_2(i))^2}$
Manhattan	$\delta(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^{d} \lvert \mathbf{x}_1(i) - \mathbf{x}_2(i) \rvert$

Since feature extractors and distance measures are generally used together, the pair of a feature and a distance measure is called a descriptor [316]. For ease of reading, in this study, the terms descriptor and features are used interchangeably.

In this work, a wide variety of descriptors from different categories were considered. All features are extracted through an unsupervised process, without using any labeled data from the target dataset since deep learning descriptors always perform transfer learning. A complete list of all descriptors used and other details are presented in Section 4.2.

2.2.2 Unsupervised Similarity Learning for Re-Ranking

Despite the significant evolution in descriptors and more sophisticated methods for computing similarity between features, the semantic gap [24, 115] continues to be one of the main challenges. To address this and other issues, post-processing approaches have been employed to improve the retrieval results. Initially, most of these algorithms were based on relevance feedback [312]. However, relevance feedback methods require considerable amounts of supervision, which is often costly or not possible in many scenarios.

As an alternative in scenarios with limited supervision, unsupervised similarity learning approaches have been proposed. These methods are highly promising because they can provide gains in retrieval results by post-processing similarity matrices or ranked lists without requiring labeled data. Even with the advancement of deep learning, these methods continue to demonstrate significant effectiveness gains by exploiting the similarity relationships in the neighborhood and considering the geometry and underlying structure of the datasets [250]. These approaches can employ various strategies such as graph transduction [381], diffusion processes [382], affinity learning [384], manifold learning [241], re-ranking [237], and contextual measures [128]. It is important to note that a single method, though not necessarily common, can integrate multiple of these techniques [252].

In this work, the term re-ranking is often used as a synonym for unsupervised similarity learning as our focus is on rank-based approaches that utilize contextual similarity information. Also, not all methods consider similarity measures, some operate with distances instead. For re-ranking, using similarities in the learning process is generally preferred since it enhances scalability and allows the use of sparse matrices, which require significantly less memory. Therefore, for convenience, the term similarity is used in most cases. Figure 2.2 provides an overview of the main steps involved in utilizing unsupervised similarity learning for image retrieval.

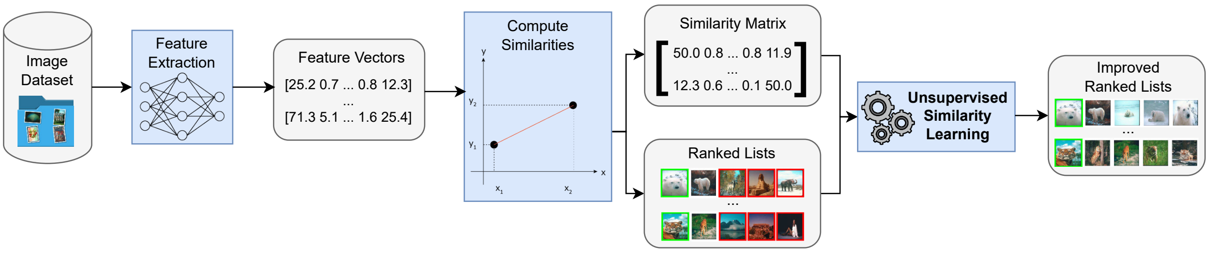


Figure 2.2 – Overview of unsupervised similarity learning workflow for image retrieval.

First, features are extracted from an image dataset, and similarities are computed based on these features. This can be efficiently performed by optimized indexing algorithms, such as the BallTree [234]. There is a great variety of indexing approaches [283], and a discussion about them is beyond the scope of this work. The input for unsupervised similarity learning approaches varies depending on the specific method. Some only handle similarity or distance matrices, others only work with ranked data, and some methods allow any of the two. As output, improved ranked lists are obtained, which are generally more effective than the original ones.

Different techniques are employed by re-ranking approaches: graphs [249, 251], Cartesian products [332], search trees [253], correlation measures [240, 249], and others. For example, RL-Sim [240] assesses the correlation between ranked lists. This correlation data is then utilized to create a similarity matrix, which is part of the method during the learning phase. The underlying concept is that if two items have ranked lists with high similarity, it is likely that the items themselves are similar, and vice versa.

Another example is the Correlation Graph (CG) [249], which creates a graph where each node represents an image, and edges are formed between images if their correlation is above a specified threshold. The method iterates over various thresholds and updates an internal similarity matrix. This matrix is then used to reorder the ranked lists.

2.2.3 Formal Definitions and Notations

This section presents the formal definition and the main concepts for the image retrieval problem addressed in this work. The definitions adopted are similar to those used in the literature and in other re-ranking approaches [316, 329, 328].

- **Retrieval and Rank Model**

Let $\mathcal{C}$=$\{o_1, o_2, \ldots, o_N\}$ be an object collection, where $N = |\mathcal{C}|$ denotes the collection size. In this work, an object refers to an image. Let us consider a retrieval task where, given a query image, returns a list of images from the collection $\mathcal{C}$.

Formally, given a query image o_q, a ranker denoted by R_j computes a ranked list τ_q=$(o_1,\ o_2,\ \ldots,\ o_k)$ in response to the query. The ranked list τ_q can be defined as a permutation of a set $\mathcal{C}_L$ which contains the L most similar images to image o_q in the collection $\mathcal{C}$. The permutation τ_q is a bijection from the set $\mathcal{C}_L$ onto the set $[L] = \{1, 2, \ldots, L\}$. The $\tau_q(i)$ notation denotes the position (i.e., rank) of image o_i in the ranked list τ_q, such that $\tau_q(i) \in \mathbb{Z}^+_{>0}$.

The ranker R_j can be defined based on diverse approaches, including feature extraction or learning methods. In this work, feature-based approaches are considered, defining R as a tuple (ϵ, δ), where $\epsilon : \mathcal{C} \rightarrow \mathbb{R}^d$ is a function that extracts a feature vector v_x from an image $o_x \in \mathcal{C}$; and δ: $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$ is a distance function that computes the distance between two images according to their corresponding feature vectors. Formally, the distance between two images o_i, o_j is defined by $\delta(\epsilon(o_i),\ \epsilon(o_j))$. The notation $\delta(i, j)$ is used for readability purposes.

A ranked list can be computed by sorting images in a crescent order of distance. In terms of ranking positions, we can say that if image o_i is ranked before image o_j in the ranked list of image o_q, that is, $\tau_q(i) < \tau_q(j)$, then $\delta(q, i) \leq \delta(q, j)$. Taking every image in the collection as a query image o_q, a set of ranked lists $\mathcal{T} = \{\tau_1, \tau_2,\ \ldots,\ \tau_N\}$ can be obtained.

Different features and distance functions give rise to different rankers which, in turn, produce distinct sets of ranked lists $\mathcal{T}$. Let $\mathcal{R} = \{R_1,\ R_2, \ldots,\ R_m\}$ be a set of rankers and $R_j \in \mathcal{R}$, we denote by $\mathcal{T}_j$ the set of ranked lists produced by R_j. A ranked list computed by the ranker R_j in response to a query o_q is denoted by $\tau_{j,q}$.

- **The Neighborhood Set**

Based on the rank model, the neighborhood set can also be defined. Let o_q be a multimedia object taken as query, a neighborhood set $\mathcal{N}(q,k)$ that contains the k most similar multimedia objects to o_q can be defined as follows:

$$\mathcal{N}(q,k) = \{\mathcal{S} \subseteq \mathcal{C}, |\mathcal{S}| = k \wedge \forall o_i \in \mathcal{S}, o_j \in \mathcal{C} - \mathcal{S} : \tau_q(i) < \tau_q(j)\}. \tag{2.1}$$

- **Distance and Similarity Matrices**

Besides ranked lists, the distance $\delta(i,j)$ between all objects obj_i, $obj_j \in \mathcal{C}$ can also be calculated to obtain a square distance matrix $\mathbf{D}$, such that $\mathbf{D}_{ij} = \delta(i,j)$. Analogously, a similar process can be employed to obtain a square similarity matrix $\mathbf{S}$. Let $\rho : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a function that computes the similarity between images, such that, $\mathbf{S}_{ij} = \rho(i,j)$ corresponds to the similarity between images o_i and o_j. The distance matrix $\mathbf{S}$ or the similarity matrix $\mathbf{S}$ are used as input for various post-processing methods, which often learn more effective similarity or distance measures. These matrices can also be used to compute ranked list representations, as previously described.

2.3 Feature Selection and Fusion

Since different descriptors and methods provide complementary retrieval results, an interesting approach is to combine them [260]. Various works have shown that by fusing diverse features, effectiveness can significantly improve [260, 322, 329, 328]. This scenario leads to two challenging and complex tasks. The first is the selection: *how to select a combination of features?* The second is the fusion: *how to fuse the selected features?*

In terms of efficiency, the selection is imperative. While the number of features increases linearly, the number of combinations grows exponentially. For example, for 10 features, there are 1024 combinations of any size. However, with 20 features, the number of possible combinations increases to 1,048,576.

There are many types of selection algorithms. This section discusses the most representative and relevant ones in the context of this study. For feature-level selection, one notable algorithm is Relief [322], which estimates the importance of features by assessing how well their values distinguish between nearby instances. There are other approaches, that also work with features such as: Laplace [112], Spectral Regression (SPEC) [421], Muti-cluster Feature Selection (MCFS) [36], Non-negative Discriminative Feature Selection (NDFS) [182]. These algorithms are mostly exploited for classification.

Another type of selection, which is the focus of this study, is rank-based selection [329]. One advantage of using ranked lists is the ability to directly exploit contextual similarity information and relationships among images. The idea is that ranked lists that are highly complementary and also effective provide good potential for combination. In unsupervised scenarios, rank-based correlation measures can be utilized to identify the most complementary ranked lists. To determine the most effective lists, effectiveness estimation measures, also known as query performance prediction (QPP), can be employed. After choosing a combination using a criteria, the next step is fusion.

The fusion process involves taking a set of data from multiple sources and using an algorithm that combines these features to produce a more effective output. In the context of this work, the terms combination and fusion can be used interchangeably in many cases. However, there is a subtle difference in their meanings. While combination refers to the set of features or descriptors, fusion is the process an algorithm uses to learn from the features in the combination. The fusion approaches are categorized into two main groups [260]:

- **Early Fusion:** This term is broadly used for feature-level fusion. It involves fusing the raw feature vectors, which are the output of the feature extractors. Some simple examples of early fusion are the concatenation, weighting, multiplication, or summation of feature vectors. For image retrieval, this fusion is performed prior to the computation of the similarity matrices or ranked lists.
- **Late Fusion:** Contrary to early fusion, late fusion performs combination using structures derived from the feature vectors, such as the similarity matrices or ranked lists. Some examples in this category are the multiplication or summation of matrices. There are also unsupervised similarity learning approaches that compute fusion based on the ranked lists, this strategy is known as rank-aggregation.

It should be noted that a system can utilize both early and late fusion. While some researchers identify additional categories, such as intermediate fusion, which occurs between early and late, they are beyond the scope of this study. This work specifically concentrates on unsupervised rank-based late fusion.

A more detailed description of query performance prediction and correlation measures, that can be used for rank-based selection are presented in Sections 2.3.1 and 2.3.2, respectively. Re-ranking methods can be used for fusion, with a strategy called rank-aggregation, discussed in Section 2.3.3.

2.3.1 Query Performance Prediction

The task of query performance prediction (QPP) involves estimating the effectiveness of a ranked list without relying on labeled data. Also commonly referred to as query difficulty prediction, query difficulty estimation, and effectiveness estimation; these methods were initially developed for traditional text-based information retrieval systems.

Despite recent advancements, applying QPP to image retrieval is still a largely unexplored area [262]. However, QPP in text and image retrieval is equally important.

For selecting ranked lists, QPP is crucial in distinguishing poorly performing queries from effective ones. It helps to filter out low-quality results and select only the most efficient ranked lists for combination. Moreover, QPP has several other applications, including:

- Controlling the convergence of a retrieval algorithm by measuring the quality of ranked lists. When the quality no longer varies significantly, it indicates that convergence has been achieved, for example.
- Enhancing search optimization by predicting query performance, thereby ensuring that the most relevant information is presented to the user.
- Providing visualization of the cases where a retrieval system is not performing as expected, these cases work as possible suggestions for enhancements of the algorithms employed.

In terms of supervision, there are both supervised and unsupervised QPP approaches. A QPP method can be trained on one dataset before being applied to a target dataset, which is an example of transfer learning. In this work, only completely unsupervised approaches were considered.

Two examples of QPP measures that estimate the effectiveness of ranked lists are the Authority [243] and Reciprocal Density [248]. Both yield a score in the range of $[0, 1]$, where a higher score indicates a ranked list of higher effectiveness and vice versa. They are both based on the cluster hypothesis [155], considering that the images belonging to a highly effective ranked list should appear in the ranked lists of each other. Both count the number of reciprocal neighbors in the top-k positions, which can be understood as measuring the density of a neighborhood graph. In contrast to Authority, Reciprocal Density assigns a weight based on the positions in which the elements appear.

2.3.2 Rank Correlation Measures

Measuring the correlation is essential to know the degree of similarity between different data points. Diverse types of measures assess the correlation between features, scores, and other types of variables. In the context of this work, rank-based correlation measures are employed for ranked list selection and improving the effectiveness of re-ranking approaches.

The concept of rank-based correlation involves measuring the similarity between two ranked lists [16]. For fusion, the correlation is used to assess the similarity between ranked lists of the same query from different descriptors. This measurement helps determine the level of complementarity or redundancy among descriptors. A high degree of complementarity is generally preferred in fusion processes [329]. There are also many other applications of rank-based correlation, including:

- If the ranked list of two images shows high similarity, these images are likely similar [155]. Therefore, the similarity learning approach can bring these two images closer [240] in the learning space.
- Graph approaches can model networks in which nodes depict images and edges represent correlations between them [249]. These correlations can be used to weigh the edges or to create edges only between the most similar nodes. More effective correlation measures can lead to more accurate results.
- For pseudo-label generation [264]. If a labeled image belongs to the same class as an unlabeled image and both have ranked highly correlated ranked lists, they probably belong to the same class.
- It can be used to detect and filter outliers. If an image has a lower correlation with all of the other elements in the dataset, it is probably an outlier.

Various properties can be considered when measuring the similarity between ranked lists, such as: *(i)* the number of elements that the lists have in common; *(ii)* the order that elements appear in both lists; *(iii)* by counting the number of overlaps in different depths.

The Jaccard index, a traditional and widely used method, takes into account the number of elements that the lists have in common [16]. It calculates a score by dividing the length of the intersection of the two lists by the length of their union. Other traditional measures, such as KendallTau [87], consider the order in which elements occur by counting the number of concordant and discordant pairs. A pair is concordant if the ranks for both elements agree in order and discordant if they disagree.

Some consider overlaps in different depths such as the case of Rank-Biased Overlap (RBO) [358]. It measures the similarity between two ordered lists considering their order and partial overlaps. It employs a persistence parameter to weigh the top of the lists more heavily, calculates overlap at each depth, and sums these overlaps with decreasing weights to produce a final similarity score.

2.3.3 Rank-Aggregation

The rank-aggregation deals with combining multiple ranked lists into a single aggregated one [87], a classic and challenging task that has been investigated for a long time [38, 260]. Any type of fusion that combines ranked lists can be understood as a rank-aggregation technique. It can be used to combine ranked lists from different descriptors or sources aiming at providing a more effective ranked list as output. This research is dedicated specifically to rank-aggregation in unsupervised image retrieval, but there are various other applications in different fields (e.g., social choice theory, collaborative filtering, web search, statistics, databases, sports, and admission systems) [38, 7].

An example of a traditional approach in this category is the Borda Count [394], where each item receives points based on its position in each ranked list: the top-ranked item gets the most points, decreasing down to the lowest-ranked item. The points across all ranked lists are summed for each item, and items are then ranked from highest to lowest total score. Other examples include MedianRank [87], where items are ranked based on the median of their positions in all rankings; and Copeland's Method, where each item competes against every other item in pairwise comparisons across all lists. An item scores a point for each pairwise win, and items are ranked based on their total scores.

In addition to conventional approaches, several re-ranking methods support the input of one or more ranked lists to perform rank-aggregation. Figure 2.3 illustrates the workflow of a re-ranking algorithm employed for rank-aggregation. From a single dataset, various descriptors can be used to generate different ranked lists. These lists are then combined through rank aggregation to produce a single, consolidated ranked list as output.

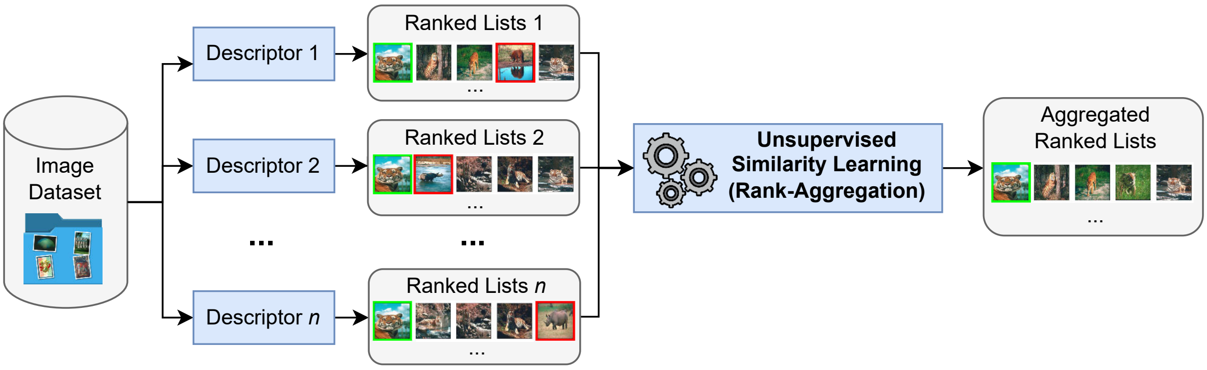


Figure 2.3 – Overview of unsupervised similarity learning applied for rank-aggregation in image retrieval. Complementary information from multiple descriptors is combined.

Re-ranking methods typically consider an internal similarity matrix learned during the algorithm execution [326]. This matrix is also used to reorder ranked lists through a sorting process, such as insertion sort, heap sort, or merge sort. This matrix is generally initialized considering the positions of the ranked lists provided as input. For efficiency and scalability purposes, a sparse matrix can be utilized. Such a process is executed for a single set of ranked lists (i.e., ranker). When multiple rankers are provided as input (i.e. rank-aggregation), there are multiple ways that the method can handle various inputs, among them, the most common:

1. The matrix initialization is performed individually for each ranker, and the matrices are then combined through an arithmetic operation (e.g., sum or multiplication).

The rest of the algorithm proceeds unchanged, in the same manner as it would for a single input ranker.

2. The algorithm is executed separately for each ranker, and the individual matrices from each ranker's execution are subsequently combined through arithmetic operations.

Among the approaches used in this work, the Correlation Graph (CG) [249], considered in Chapter 6, and the proposed Rank Flow Embedding (RFE) [334], presented in Chapter 8, employ strategy (2). All other approaches [251, 252] utilize strategy (1).

It is important to note that, depending on the task, aggregation can be performed in multiple steps. For example, it is possible to aggregate rankers pairs and merge them individually. However, these are beyond the scope of this work. This study considers aggregation where all rankers are simultaneously combined as input.

2.4 Person Re-Identification

Person re-identification, usually abbreviated as Re-ID, is crucial for enhancing security and surveillance systems by enabling the identification of individuals across multiple camera feeds [25, 390]. Among the various applications, it can improve public safety by assisting in crime prevention, suspect tracking, crowd control, and finding missing persons in different environments (e.g., shopping malls, railway stations, airports, universities, and huge public events) [288]. The task of person re-identification is defined in the literature as follows:

> "Given an image or video of a person from a camera, the re-identification process involves identifying the same individual from images or videos taken from different cameras, which may or may not have overlapping fields of view. Re-identification is indispensable in establishing consistent labeling across multiple cameras or even within the same camera to re-establish disconnected or lost tracks." [25]

However, there are variations of definitions in the literature. Some authors consider that the cameras cannot have overlapping fields of views [176, 422, 428, 137], for example. The Re-ID process is generally complex and consists of a sequence of main steps [426], which are illustrated in Figure 2.4 and described as follows: **(1) Detection**, where, given an image or a video frame, the regions where the people of interest are present are segmented; **(2) Tracking**, where, in the case of a video, it consists of the task of following the movement of the people detected; **(3) Retrieval or Classification**, where, from the segmented person, the task is to return the individuals most similar to the person of interest.

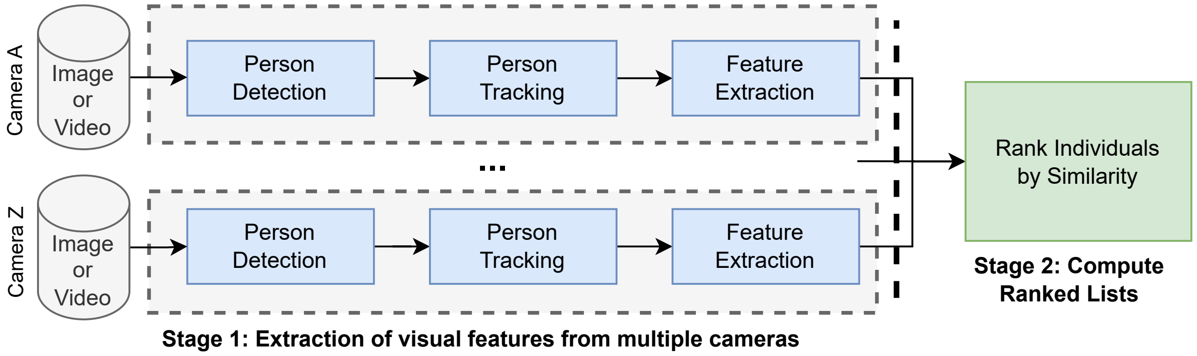


Figure 2.4 – General diagram of a Re-ID system. Figure adapted from [25].

Regarding the detection stage, one of the major challenges often lies in the significant variation in positions between images of the same person. In some datasets, the process of separating images that belong to the same person is performed manually. However, currently, there are various approaches in this regard, some of which are based on the calculation of similarities between parts (patches) of the images. Although many patches show large differences for images of the same person, some regions may indicate great similarity, such as the facial area, unless there are occlusions, for example. In general terms, the DPM (Deformable Parts Model) [173] method applies the extraction of patches from images through a pre-trained network that extracts patches of different sizes, which are grouped based on the positions they belong to on the human body, assigning a score to each one of them. The CUHK03 [176] and Market1501 [422] datasets contain images cropped with the DPM detector, for example. Another widely used detector is the ACF (Aggregated Channel Features) [73], where different channels are calculated for the input image and a decision tree provides the segmentation of the person. There are also recent approaches that use neural networks for object detection, which can also be used to identify pedestrians, such as in the case of the YOLO (You Only Look Once) [272] network.

In the vast majority of cases, cameras produce various videos, which have different frames featuring the same person in motion. The tracking phase involves monitoring an individual's movements across a sequence of video frames following their initial detection. Among the different approaches applied for detection, most methods search for intersections or similarities between patches from different frames to find a match, as is done by KCF (Kernelized Correlation Filters) [113], for example. The idea is that similar patches from different frames usually refer to the same object. Still, there are various challenges, such as the difficulty of tracking in cases of occlusion or intense movement, for example. Among some of the prominent object-tracking methods known in the literature, we can mention: BOOSTING [105], KCF (Kernelized Correlation Filters) [113], CSRT (Discriminative Correlation Filter with Channel and Spatial Reliability) [207], MOSSE (Minimum Output Sum of Squared Error) [32], SORT (Simple Online and Realtime Tracking) [29]. Recently, deep learning methods have also been applied to tracking, an example of this is DeepSORT [361], which is a variant of SORT [29], but uses deep learning.

The person re-identification problem is addressed by many authors as an image retrieval task [137], which is the final stage of a Re-ID system. Section 2.4.1 presents the concepts and terminologies commonly used for Re-ID.

2.4.1 Concepts and Terminologies

This section aims to present some concepts and categories commonly used in the Re-ID literature. The methods are categorized according to the protocols considered for both training and testing with queries.

Currently, a large portion of Re-ID methods are based on machine learning techniques. In general, machine learning algorithms are those that have the task of analyzing a certain volume of data and automatically extracting patterns and information about them, so that it is possible to make data-based decisions instead of being explicitly programmed for a specific task [154]. Due to the increasing volume of data, these methods have gained much attention for the possibility of automating various tasks.

For person Re-ID, this work focuses on unsupervised methods, which, despite often being more challenging, present themselves as a promising solution in many scenarios where there is a lack of labeled data. Among the Re-ID methods within this category, we can mention: ARN (Adaptation and Re-Identification Network) [181]; EANet (Enhancing Alignment Network) [118]; ECN (Exemplar Memory Convolutional Network) [431]; MAR (MultilAbel Reference Learning) [397]; TAUDL (Tracklet Association Unsupervised Deep Learning) [170]; UTAL (Unsupervised Tracklet Association Learning) [171].

In addition to the mentioned categories, there is also the concept of cross-domain learning [153], which consists of training on labeled data from one or more image datasets (training datasets are called source domains) and making predictions on a dataset where the labeled data is completely unknown (test datasets are called target domains). Cross-domain learning is considered a type of transfer learning. It is usually referred to as multi-source when training is conducted on more than one dataset and as single-source for a single dataset. Among the Re-ID methods within this category, we can mention: HHL (Hetero and Homogeneously Learning) [430]; ATNet (Adaptive Transfer Network) [197]; CSGLP (Camera Style Generation and Label Propagation) [273]; ISSDA (Iterative Self-Supervised Domain Adaptation) [306].

From cross-domain learning, there are also proposals for domain-adaptive methods (domain adaptation) [153]. In this case, the learning is very similar to cross-domain, with the difference that a final training step is performed on the test dataset (target domain), but without using the labeled data from this dataset. Only the labeled data from the training dataset (source domains) are used. Among the Re-ID methods within this category, we can mention: EANet (Enhancing Alignment Network) [118]; SPGAN (Generative Adversarial Network) [71]; DAAM (Domain Adaptive Attention Model) [121]; PAUL (Patch-Based Unsupervised Learning Framework) [380]; CAMEL (Cross-view Asymmetric Metric LEarning) [396]. Some methods may belong to more than one category, defined by the training method used during evaluation.

Furthermore, the images in Re-ID datasets are generally subdivided into three sets [176, 429, 422, 428]: *(i)* training images (training set), a set of images used for training the methods; *(ii)* query images (probe set), which refer to the images that should be used to perform retrieval against the gallery images; *(iii)* gallery images (gallery set), the images that will be ranked according to their similarity to the query image. Thus, in most cases, the method is trained on the training set and then generates a ranked list for each query image. A ranked list contains the query image (belonging to the probe set) as the first image, with the remaining images belonging to the gallery set ranked according to similarity. In many cases, the subdivisions are made by the authors of the dataset, but it should be noted that, in some cases, these protocol definitions can be altered, and there can be different protocols for the same dataset, as in the case of CUHK03 [176, 429]. Additionally, unsupervised methods do not always perform training on the training set of the evaluated dataset, and when they do, they do not consider the labels of this set in the process [121, 171].

Regarding the retrieval methodology, there are two distinct ways for a Re-ID method to perform a query: single-query or multi-query. When a single image of an individual is used as a query, it is a single-query retrieval. On the other hand, when two or more images are provided as input, it is a multi-query case. In some cases, both protocols may be used, while in others only one is used. Some datasets may be used exclusively for single-query, such as VIPeR [106], GRID [206], and CUHK01 (images from the Chinese University of Hong Kong) [175], as these have only two images per individual. Others may be considered in both cases, as long as there is more than one image per individual.

Additionally, there is the concept of single-shot and multi-shot in re-identification. Single-shot refers to the scenario where only one image per person is available in the gallery and query set, making the task challenging due to the limited number of examples. Conversely, multi-shot involves having multiple images of each person, which provides diverse perspectives and poses of the individuals. While multi-shot offers more samples per person, it can also be more difficult due to the increased complexity of managing and comparing multiple images per individual.

Table 2.2 summarizes all the concepts and terminologies discussed for person Re-ID. There are four main categories of concepts: *(i)* Re-ID dataset split, where datasets are divided into three sets provided by the dataset authors; *(ii)* query type, which refers to how the queries are provided; *(iii)* capture frequency, which refers to the number of images per individual in the dataset; *(iv)* dataset usage, describing how the usage of datasets affects their terminology; *(v)* methods category, where all methods are unsupervised but include subcategories specific to Re-ID scenarios. Although all listed method categories are unsupervised, the Re-ID literature often differentiates by using the terms *fully unsupervised* or simply *unsupervised* for methods that do not involve any form of transfer learning [367]. In contrast, domain adaptive and cross-domain consider transfer learning.

Table 2.2 – Summary of concepts and terminologies discussed for Re-ID.

Category	**Term**	**Description**
Re-ID Dataset Split	Training Set	Used to train the Re-ID model. This set contains labeled images which help the model learn how to differentiate between different individuals.
	Query Set	Also known as the probe or test set, it contains the images used as queries, which are compared against the images in the gallery set.
	Gallery Set	When a query image is presented, the method searches through the gallery set to find images similar to the query. This set is sometimes referred to as the reference set.
Query Type	Single-Query	A single image or a frame of an individual is used as the query. It relies on a one-to-many comparison, where the query image is compared against all gallery images.
	Multi-Query	Multiple images or frames of the same individual are used as the query. This method often aggregates the results from all query images to return a single ranked list.
Capture Frequency	Single-Shot	Each individual is represented by a single image in the gallery and query sets.
	Multi-Shot	Each individual is represented by multiple images in the gallery and query sets.
Dataset Usage	Source Dataset	Dataset used to train a Re-ID model.
	Target Dataset	Dataset used to test a Re-ID model. For testing, query and gallery sets are considered.
Methods Category	Unsupervised	These methods do not rely on labeled data. They utilize unlabeled datasets to learn discriminative features or patterns inherent in the data.
	Cross-Domain Methods (Single-Source)	These methods perform transfer learning. They are trained in a single dataset known as the source and are tested in a different target domain.
	Cross-Domain Methods (Multi-Source)	In contrast to single-source methods, multi-source cross-domain methods utilize multiple source domains to improve the adaptability of the model to the target domain.
	Domain Adaptative Methods	They are similar to cross-domain, with the difference that a final training step is performed on the target dataset, but without using the labeled data from this dataset. Only the labeled data from the source datasets are used.

2.5 Graph-Based Semi-Supervised Classification

Graphs have a wide range of applications, as they facilitate the identification and analysis of patterns and relationships among items. Since these aspects are fundamental to many tasks, particularly in machine learning, graph learning has emerged as an important research field. Graph learning is a term that broadly defines machine learning methods that utilize graphs [369]. As shown in Figure 2.5, these approaches are divided into four major categories [369]:

- **Graph Signal Processing (GSP)**: These methods extend classical signal processing techniques to graphs. In GSP, the data is represented as signals on the nodes of a graph, and the graph structure itself is used to analyze these signals. Key concepts include the graph Fourier transform, graph filters, and spectral analysis. These methods are particularly useful for tasks such as smoothing, and denoising.
- **Random Walk Based Methods**: They leverage random walks to capture the structure of the graph. A random walk is a stochastic process where a node is visited based on a probability distribution over its neighbors. These methods, like DeepWalk and Node2Vec, generate node embeddings by simulating random walks and using the sequences of visited nodes to learn latent representations. These embeddings can be used for various tasks, including node classification and link prediction.
- **Matrix Factorization Based Methods**: They decompose a graph's adjacency matrix or other related matrices into lower-dimensional matrices that capture latent features of the nodes or edges. Techniques like Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF) are common. These methods are used to find latent patterns and relationships in the graph and are useful for tasks such as recommendation systems, clustering, and community detection.
- **Deep Learning Based Methods**: These methods apply neural network architectures to graph data. This category includes methods like Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs), and Graph Autoencoders. These models can learn complex, non-linear representations of graph-structured data and are highly effective for tasks like node classification, graph classification, and link prediction. Deep learning methods can capture both local and global graph structures through multiple layers of processing.

In semi-supervised learning, the use of graphs is especially advantageous [293]. Graph-based Semi-Supervised Learning (GSSL) methods generally begin by creating a graph where the nodes represent all the samples and the weighted edges indicate the similarity between pairs of nodes. This graph construction suggests that nodes connected by edges with large weights are likely to have the same label, reflecting the manifold assumption. The manifold assumption states that samples situated close to each other on a low-dimensional manifold should have similar labels.

This work centers on semi-supervised node classification employing Graph Convolutional Networks (GCNs), a type of Graph Neural Networks (GNNs) [141]. Section 2.5.1 discusses about GCNs and Section 2.5.2 presents the formal definitions.

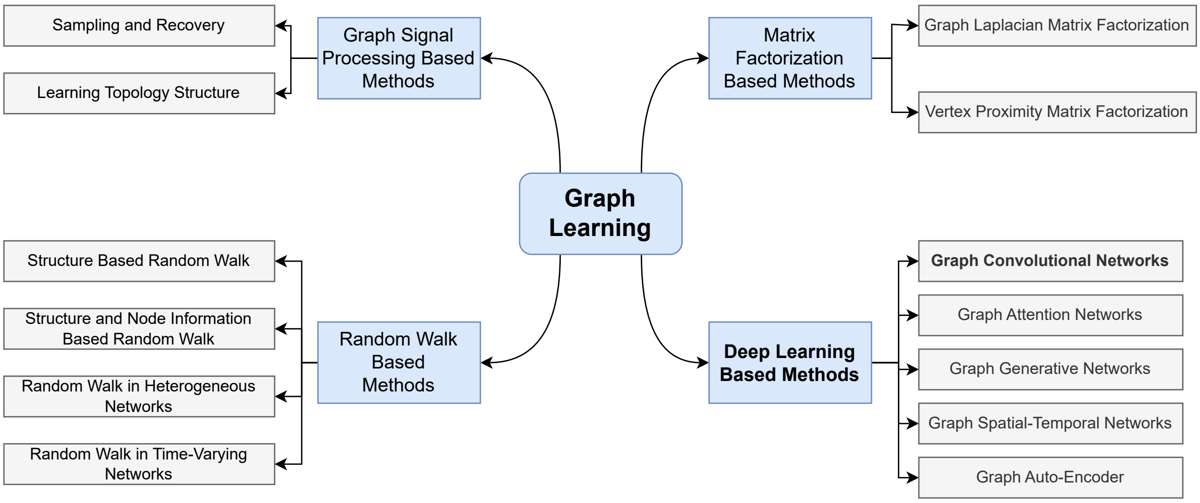

Sampling and Recovery
Learning Topology Structure
Graph Signal Processing Based Methods
Matrix Factorization Based Methods
Graph Laplacian Matrix Factorization
Vertex Proximity Matrix Factorization
Graph Learning
Structure Based Random Walk
Structure and Node Information Based Random Walk
Random Walk in Heterogeneous Networks
Random Walk in Time-Varying Networks
Random Walk Based Methods
Deep Learning Based Methods
Graph Convolutional Networks
Graph Attention Networks
Graph Generative Networks
Graph Spatial-Temporal Networks
Graph Auto-Encoder

Figure 2.5 – Categorization of graph learning approaches. Figure adapted from [369].

2.5.1 Graph Convolutional Networks (GCNs)

A convolution is a mathematical operation widely used in fields like image processing, classification, and deep learning [2]. Convolutional layers can enhance the ability of neural networks to capture patterns in image data. In CNNs, convolutional layers help in capturing the spatial hierarchy in images. Usually, lower layers can detect simple features like edges, corners, and textures, while higher layers can detect more complex aspects like objects and scenes [393].

While CNNs perform convolution in the spatial domain, applying convolution in graph domains presents considerable challenges. Unlike images with a regular grid structure, graphs have an irregular structure [100]. Nodes in a graph can have a variable number of neighbors, making it difficult to apply a consistent convolution operation. In this scenario, GCNs have been proposed to apply convolutions in the graph domain [412, 141, 135].

The input of a GCN consists of a graph, a set of features for each node, and the labels for the nodes that are part of the training set. In most image datasets, a graph is not readily available. Therefore, the graph must be constructed using methods such as computing the k-nearest neighbors (kNN) or other strategies, as explored in this work. In the following, each of the main steps of a GCN are outlined:

- **Nodes initialization**: Each node is initialized with the raw (i.e., original) features provided as input. The edges are defined according to the input graph.
- **Graph message passing**: This step refers to the convolution operation, which is divided into two steps:
  - **Neighborhood aggregation**: In each layer of the GCN, every node collects features from its immediate neighbors. The method of aggregation can vary.

  Common approaches include summing up the features, taking the mean, or using a more complex function like a neural network to aggregate these features.
  - **Transformation**: After aggregation, the aggregated feature vector is usually combined with the node's own features. This combined feature vector is then passed through a transformation function, typically a linear transformation followed by a non-linear activation function (e.g., ReLU). This step effectively updates the feature representation of the nodes.

- **Layer stacking**: The output of the previous layer becomes the input to the next layer. The number of layers is closely related to the neighborhood depth that the network can analyze. Multiple layers can be stacked to allow the network to capture higher-order dependencies in the graph (i.e., features from extended neighborhoods beyond just immediate neighbors). With two layers, for example, a node can gather information from its neighbors and also from the neighbors of its neighbors (2-hop neighbors). Thus, a three-layer GNN can access information from up to 3-hop neighbors, and so on.

- **Normalization**: Often, normalization techniques such as dividing by the square root of the degree of the node and its neighbors are applied during aggregation. This helps in stabilizing the learning process by keeping feature magnitudes in a reasonable range.

- **Output layer**: For node classification, the embeddings produced by the final GCN layer are processed through a classification layer (such as a softmax) to predict the label of each node.

All the GCNs in this work are transductive, they require access to the entire graph during training to effectively learn node representations. The training and inference (i.e., testing) procedures can be executed based on the defined steps, as follows:

- **Training**: The network is trained using a suitable loss function, like cross-entropy for classification tasks. This process involves backpropagation to adjust the weights of the network based on the difference between actual and predicted values.

- **Inference**: The trained model is used to predict the classes of nodes, focusing on sections of the graph that were not exposed during training and validation.

In this study, each image is represented by a node, allowing for image classification by representing the image dataset as a graph.

2.5.2 Formal Definitions and Notations

In this section, we first discuss a formal definition of the semi-supervised learning setting for classification tasks using GCNs, mostly following the notation from [146, 242].

Let $\mathcal{C}$=$\{o_1, o_2, \ldots, o_n\}$ be an object collection, where $o_i \in \mathcal{C}$ denotes an image and N denotes the collection size. The collection is represented by an undirected graph G. The graph can be formally defined as tuple $G = (V, \mathbf{X}, E)$, where V denotes the node set, $\mathbf{X}$ is a feature matrix, and E denotes the edge set.

The node set is defined by $V = \{v_1, v_2, \ldots, v_n\}$ where each node $v_i \in V$ represents an image $o_i \in \mathcal{C}$. Labels can be assigned to nodes $v_i \in V$, such that a set of labels can be defined as $\mathcal{Y} = \{y_1, y_2, \ldots, y_c\}$. According to the labels, the node set can be more specifically defined as $V = \{v_1, v_2, \ldots, v_L, v_{L+1}, \ldots, v_n\}$, which denotes a partially labeled data set, where $V_L = \{v_i\}_{i=1}^{L}$ is the labeled data items subset and $V_U = \{v_i\}_{i=L+1}^{n}$ is the unlabeled data items subset. Formally, the training set can be seen as a labeling function $f_l : V_L \rightarrow \mathcal{Y}$, where $y_i = f_l(v_i) \forall v_i \in V_L$. In general, on semi-supervised scenarios, we have $|V_L| \ll |V_U|$.

The feature matrix can be defined as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$, where $\mathbf{x}_i$ is a d-dimensional feature vector which represents the image o_i, or equivalently, the node v_i. The vector $\mathbf{x}_i$ is obtained by a feature extraction approach, which can be defined as function $\epsilon : \mathcal{C} \rightarrow \mathbb{R}^d$, such that $\mathbf{x}_i = \epsilon(o_i)$.

The edge set E is a set of nodes pairs (v_i, v_j), formally defined as $E \subseteq \{(v_i, v_j) | (v_i, v_j) \in V^2 \land v_i \neq v_j\}$. For graph-structured content, the set E is intrinsically defined by the data. For general image data, we propose to define the set E based on the feature matrix $\mathbf{X}$. How to define an effective graph is a central challenge addressed by our approach, discussed in the next section.

Once defined the graph G, a GCN model denoted by a function f_{gcn} can be used to learn an embedded representation $\mathbf{z}_i$ for each node v_i. The learned representation is exploited to perform classification tasks. Formally, the classification goal is to learn a function $\hat{f}_l : V_U \rightarrow \mathcal{Y}$ to predict the labels of unlabeled nodes in V_U.

2.6 Hypergraph Model

A hypergraph is a generalization of a graph where edges, called hyperedges, can connect any number of vertices [34]. In a standard graph, an edge connects exactly two vertices. This makes hypergraphs a powerful tool for modeling complex relationships and interactions in many applications [11, 102]. Despite their robust expressiveness, hypergraphs have been relatively unexplored in the literature compared to graphs [11].

Hypergraphs allow for a more natural and accurate representation by capturing high-order interactions directly [11]. For example, in a co-authorship network, a single hyperedge can link all authors of a paper, precisely reflecting their collaborative relationship, which would be less accurately represented by multiple pairwise edges [19].

When attempting to model these complex interactions with traditional graphs, significant information about the group dynamics and the interdependencies among the nodes can be lost [11]. Hypergraphs preserve this information by inherently representing these high-order relationships without the need for transforming them into simpler pairwise interactions, which often require additional nodes and edges [34].

Figure 2.6 presents a hypergraph example, where the nodes and hyperedges are denoted by v_i and e_i, respectively. The i is used to indicate the index. Notice that hyperedges can be viewed as groups consisting of one or more vertices. These groups may also overlap with one another. This can provide valuable information, which is particularly useful in this work for contextually modeling image data.

Another important property of hypergraphs is that two different types of weights [251] can be considered:

- **Vertex weights:** The weight or association of node v_j to the hyperedge e_i is denoted by $h(e_i, v_j)$. This weight represents the strength or significance of the relationship between the node and the hyperedge. A hypergraph is commonly represented by an incidence matrix $\mathbf{H}$ that encodes the $h(e_i, v_j)$ values as exemplified in Figure 2.7.
- **Hyperedge weights:** The weight of hyperedge e_i is denoted by $h_p(e_i)$. These weights are commonly associated with the importance of the hyperedge. They can be computed based on various quantities depending on the application, such as the strength of the relationships within the hyperedge and the cost associated with its connections.

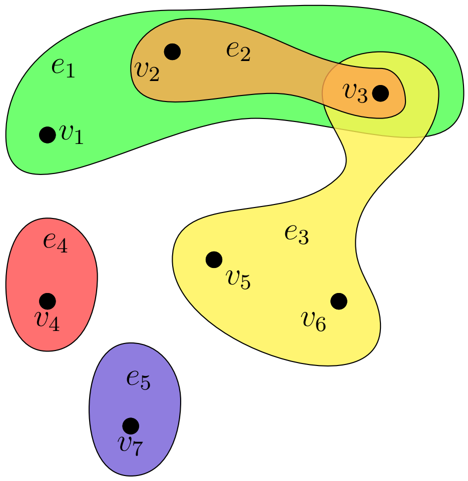


Figure 2.6 – Hypergraph illustration.

$$\mathbf{H} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \left[\begin{array}{ccccc} 2.9 & 0 & 0 & 0 & 0 \\ 5.2 & 1.1 & 0 & 0 & 0 \\ 3.7 & 2.5 & 7.1 & 0 & 0 \\ 0 & 0 & 0 & 5.9 & 0 \\ 0 & 0 & 9.8 & 0 & 0 \\ 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8.9 \end{array}\right] \end{array}$$

Figure 2.7 – Incidence matrix $\mathbf{H}$ example.

Following this discussion, Section 2.6.1 provides a brief overview of the formal definitions and notations of the hypergraph model considered in this work.

2.6.1 Formal Definitions and Notations

As defined in [251], a hypergraph can be described as a tuple $\mathbf{H_G} = (V, E_h, h_p)$, where V is a set of vertices and E_h represents the set of hyperedges. The hyperedge set E_h can be described as a collection of subsets of V such that $\bigcup_{e \in E_h} = V$. Each hyperedge e_i is assigned a positive score $h_p(e_i)$, indicating the confidence in the relationships among the vertices connected by the hyperedge e_i.

Unlike graphs, which are commonly represented by adjacency matrices, hypergraphs are represented by incidence matrices. In this work, the incidence of a hyperedge e_i on a vertice v_j is represented by an incidence matrix $\mathbf{H}$, where $h(e_i, v_j)$ denotes the reliance of the vertex v_j to belong to a hyperedge e_i.

3 Related Work

This chapter presents the related work for the main topics of this study. Section 3.1 discusses recent works that employ similarity learning for image retrieval in general-purpose scenarios. Section 3.2 presents an overview of the literature for unsupervised person Re-ID, discussing the methods for feature extraction, metric learning, and the state-of-the-art. Section 3.3 describes related works for query performance prediction in image retrieval and investigates denoising convolutional networks that can exploit contextual information in these tasks. Section 3.4 categorizes and summarizes the main areas of semi-supervised classification and discusses the importance and advantages of GCNs compared to other semi-supervised approaches. Section 3.5 reviews recent contrastive learning methods and mentions strategies that use contextual similarity information in these scenarios.

3.1 Similarity Learning in Image Retrieval

Content-based Image Retrieval (CBIR) is a central tool behind a diversified range of applications. In fact, it can be seen as technology that helps to organize digital picture archives by their visual content [60], including a broad spectrum of approaches, from general object retrieval to medical diagnostics support and person re-identification [60, 48, 438]. A traditional task is given by a query-by-example arrangement, which consists of retrieving the most similar images to a query image defined by the user from an image collection [420]. While involving various challenges and the fundamental open problem of robust image understanding [60], it can also be seen as a rank-centered task, once the retrieved images are expected to be ranked according to the user needs.

The ranking tasks performed by CBIR approaches typically rely on two basic steps: the image content representation itself and the similarity measurement of collection images to the query. The image representation is concerned with mapping an image to a point in a high-dimensional feature space. The similarity measurement, in turn, relies on assessing how close representations of collection images are from the query point in the feature space [253]. Conventionally, it is accomplished by computing the pairwise dissimilarity between feature representations in the Euclidean space [18].

Extensive advances have been made in image representation techniques over the last decades. Originally, the extraction of global features defined the dominant approach, where a myriad of features were proposed, mainly based on visual properties such as shape, texture, and color. The global features gave rise to local feature strategies, based on Bag-of-Words (BoW) model, largely studied over a decade [427]. More recently, the success of deep neural networks in feature representation has made them a fundamental tool in image retrieval. Models pre-trained on huge datasets are broadly used through transfer learning to extract features of images [378, 48].

Despite the huge advances in representation strategies, especially supported by recent deep features given by Convolutional Neural Networks (CNN) and Vision Transformers (ViT) models, a major limitation is associated with the pairwise formulation of similarity measurements. In fact, both traditional and deep-based representations lie on manifolds in a high-dimensional space [123] such that pairwise similarity measures are insufficient to reveal the intrinsic relationship between images. Instead, similarities can be estimated more accurately along the geodesic paths of the underlying data manifold [18]. The goal of such strategies is to somehow mimic human behavior in judging the similarity among objects; i.e., by considering the context of other objects.

In this research direction, different approaches have been proposed to post-process pairwise measures in order to compute more global and effective similarity measures [74, 382, 253, 247, 385, 251]. Different techniques and comprehensive terminology have been employed, all following the common objective of capturing the structural similarity information encoded in the datasets through unsupervised contextual analysis. Such contextual-sensitive similarity measures have been successfully applied to capture the geometry of the underlying manifold in order to improve retrieval tasks.

Diffusion processes demonstrated high potential in capturing the underlying manifold structure [18, 125]. Diffusion processes use a weighted graph, where each image is represented by a node, and edge weights are defined by pairwise affinity values. The pairwise affinities are re-evaluated in the context of other images, by spreading the similarity values across the graph. Affinities are spread on the manifold, which in turn improves the retrieval scores [74]. Several variants have been proposed [74], including methods capable of analyzing high-order similarity relationships [18]. Another example in this category is the Graph Diffusion Networks (GRAD-Net) [78], which utilizes graph neural networks to learn semantic representations that incorporate local and global manifold structures in an unsupervised manner. Despite their robust mathematical foundation and background, such approaches are often associated with high computational costs [252], lack of flexibility for new instances, and poor scalability for large datasets [78].

Re-ranking and rank-based manifold learning methods constitute another representative category of unsupervised post-processing methods [267, 21, 241, 247, 253]. In fact, ranked lists provide a rich source of contextual information once they establish a similarity relationship among a set of images, in contrast to pairwise relations. Additionally, the most relevant information in the ranked lists is located at top positions, which enables the development of efficient algorithms [240]. Reciprocal similarity relationships [241, 69, 243] and rank correlation measures [247, 21, 323] have been successfully applied by various approaches. Recently, some approaches have begun to apply transformers in the re-ranking and rank-aggregation pipeline to improve feature representations considering both local and global features [305, 442, 279, 406], although not all of them are unsupervised [305].

Graphs and embeddings are modeling tools that also have demonstrated a high potential for contextual similarity analysis. The shortest path in the graph is used to define the similarity between images in [351]. Connected Components are exploited in [249, 241] for spreading confident similarity relationships. Lately, hypergraphs have been exploited, mainly due to their capacity to represent high-order similarity information [251, 18]. More recently, approaches that learn a mapping function to an embedded space have been proposed that exhibit the capacity of generalizing to new data [124], but such approaches are still rarely considered in the literature.

On the other hand, unsupervised image retrieval and semi-supervised classification are well-known and largely studied tasks. However, they remain challenging and interconnected, with many applications in diverse scenarios (person re-identification [137], remote sensing [355], medical imaging [1], and many others). Despite significant advances, most approaches tend to focus on solving only one of these tasks, lacking the versatility to generalize across multiple of them. Therefore, a unified method for unsupervised similarity learning, contextual embedding, and semi-supervised classification would be desirable and innovative.

3.2 Person Re-Identification

Person Re-ID is of critical importance in the majority of modern security and surveillance applications [37, 55, 180, 426]. The task consists of, giving a query image of one person, to identify the same individual across different cameras that have no overlapping views. There are many difficulties for Re-ID retrieval [390], among them: *(i)* different viewpoints, *(ii)* possible low-image resolutions, *(iii)* illumination changes, *(iv)* occlusions, *(v)* difficulty of manually labeling data for training, *(vi)* large amount of data to be processed. The challenge of improving the effectiveness of these systems, especially in open-world scenarios, has attracted a lot of research efforts from the scientific community [137, 426, 390].

Initially, person Re-ID retrieval systems were mainly based on the use of hand-crafted feature representations [387, 106, 213, 217, 184, 192, 422, 375, 176]. Besides that, other strategies commonly used for generic image retrieval, like the bag of visual words [419, 422], have also been employed. Aiming at further improving the quality of the results, rather than using traditional distance measures, researchers have proposed metric learning approaches for Re-ID [184, 156, 94, 216, 411], most of them based on supervised models. In [137], an extensive evaluation of multiple combinations of feature extractors and metric learning approaches is discussed.

Due to the significant impact of deep learning on common image retrieval and machine learning [390], the Convolutional Neural Networks (CNN) also have gained a lot of attention and have been widely employed to solve Re-ID tasks in the recent years [390, 437, 436, 39, 177, 301, 153, 118, 197, 310]. Among the works, existing networks have been trained for Re-ID [110, 275], and some new architectures have been proposed solely focusing on Re-ID [437, 436, 39, 177, 301].

Despite the success of deep learning, one of the main difficulties resides in the lack of large publicly available Re-ID datasets for training [390], mainly because manually labeling images is a very difficult task, especially in open-world scenarios [137], where the datasets may increase dynamically. In order to mitigate this problem, many authors have proposed multi-source training, which consists of joining multiple available datasets for training [370, 396, 153], usually increasing the network capacity for generalization.

However, despite several advances in multi-source approaches, there is an inevitable need for intensive manual annotation to obtain training data. In practice, the demand for extensive training data restricts the generalization and scalability of supervised approaches, especially on person Re-ID tasks, which are not only resource-intensive to acquire identity annotation but also impractical for large-scale data [188, 189].

Usually, there are two main steps associated with the Re-ID labeling process [443, 390, 444]: *(i)* intra-camera annotation that requires comparing a person with all the other unlabeled persons in a single camera with multiple views; and *(ii)* inter-camera that requires to match a person across different cameras with multiple views. Let P be the number of persons and S the number of camera views. The intra-camera annotation complexity is $O(S \times P^2)$ and inter-camera annotation complexity ranges from $O(S \times P^2)$ and $O(S^2 \times P^2)$. The worst case of inter-camera occurs when not all persons appear in every camera view in the majority of cases, which makes the association required to repeat for all V camera views [443, 444]. Commonly, inter-camera association significantly increases standard annotation costs. There are approaches that mitigate these issues; among them there is the Intra-Camera Supervised (ICS) [443, 444], Multi-Task Multi-Label (MATE) [443, 444], and Cross-camera Feature Prediction [97], which were recently proposed.

Due to the challenge of obtaining large amounts of strongly labeled data, semi-supervised methods have been employed, which is a typical strategy for supervision minimization. Based on information learned from a small set of labeled data, the idea is to generate labels from unlabeled training data. Some research has been made in this direction [88, 200, 350, 373]. However, these methods often suffer from performance degradation and often require a large proportion of expensive cross-view pairwise labeling [443].

There are also weakly supervised strategies that replace accurate labels with inaccurate annotations. In [219], the authors proposed the idea of obtaining multiple bounding boxes of the same person from untrimmed videos. This is done by training a deep learning model capable of extracting multiple bounding boxes of the same person in a video. Recently, [349] proposed to replace image-level annotations with bag-level annotations. Weakly supervised Re-ID is very challenging since it is rather difficult to model the considerable variances across camera views (e.g. occlusion and illumination) without using strong label data [219, 443].

As an alternative solution, unsupervised approaches [181, 431, 397, 170, 171, 168, 395, 188, 189] have been attracting a lot of attention from the research community, especially because, once labeled data are not required, the methods become more suitable for real-world scenarios. In a promising research direction, to address the lack of labels issue, there are works proposed to post-process person Re-ID results by analyzing similarity relationships encoded in the datasets. Several authors have proposed unsupervised re-ranking approaches for Re-ID [429, 174, 225, 165, 211, 96, 388, 95, 108]. In [429], the original distances among images are improved by calculating the Jaccard correlation scores of each ranked list. Various approaches exploit the reciprocal neighborhood information and other co-occurrence indexes aiming at improving the ranking results.

Another strategy commonly employed on generic unsupervised image retrieval and few exploited on Re-ID tasks consists in fusion approaches [260]. In general, both broad categories of fusion have been successfully used in generic image retrieval: *(i)* early fusion, which combines the feature vectors; and *(ii)* late fusion, which usually combines ranked lists. Significant results have been achieved based on the fusion of different ranked lists and features [329, 352], with the purpose of obtaining more effective results by exploiting the complementarity of each input.

On person Re-ID tasks, some authors have proposed late fusion strategies based on rank aggregation [389, 424]. In [424], the aggregation is performed by attributing weights for each query of each ranker, but no pre-selection of features is performed, and all the features are used as input for the fusion step. There are also early fusion approaches for Re-ID, as [9], that propose a supervised multi-hypergraph fusion model for early fusion of feature extractors. It learns a hypergraph for each feature through a star expansion strategy and they are fused according to weights that the method has learned from training. In [403], a hypergraph structure is used with a deep learning model to improve the performance of the acquired features for Re-ID.

Although fusing different features can represent a significant advantage due to extra information available, how to choose what features to fuse can be a challenging task. Even for supervised approaches, selecting highly effective combinations of visual features remains a complex task, since it is necessary to consider various aspects, such as diversity and complementarity of results. Therefore, selecting features in an unsupervised way, without any labeled data is even more challenging since no information about the effectiveness of individual visual features is available.

The remaining of this section focuses on specific topics related to Re-ID and is organized as follows. Section 3.2.1 outlines the main feature extraction approaches for Re-ID. Section 3.2.2 presents some metric learning approaches and their applications. Section 3.2.3 discusses state-of-the-art methods for person re-identification and their categories.

3.2.1 Feature Extraction

Different works propose various methods for feature extraction, aiming to extract the most discriminative (relevant) information from the data. Among the most commonly traditional ones used in the context of Re-ID are:

- ELF [106], which calculates color histograms in different types of color spaces (RGB, YCbCr, and HSV), texture histograms using Schmid [276] and Gabor [89] filters, and finally concatenates this information into a single vector;
- LDFV [212], composed of local descriptors that store spatial information of pixels, intensity, and gradient information using Fisher vector representations [303];
- gBiCov [213], where biologically inspired features are stored in covariance descriptors;
- SIFT-DenseColor (SDC) [419], in which each image is subdivided into different regions, applying color histograms and SIFT feature extractors to each region, constructing something similar to a visual dictionary;
- LOMO [184], where scale-invariant HSV color and LBP [187] histograms are extracted by a multi-scale algorithm known as Retinex and then subjected to a horizontal max-pooling step;
- GOG [217], where the image is subdivided into horizontal strips, and the regions are separated such that each part is modeled according to a Gaussian distribution and then condensed into a single Gaussian distribution.

Moreover, due to the evolution of deep learning, most researchers started using different convolutional neural network architectures (AlexNet [151], ResNet [110], VGGNet [198], among others) to extract feature representations of the individuals, since these methods showed better generalization than the traditional ones. Generally, pre-trained networks on the ImageNet [70] dataset, which are general-purpose, are considered and subsequently retrained for the specific Re-ID task. There are also proposals of networks specifically designed for Re-ID, such as OSNET [437, 436], MLFN [39], and HACNN [177], for example. Another example is a two-branch CNN architecture [377] introduced for person re-identification in video surveillance, extracting both global and local features, which employs an adaptive triplet loss function to improve learning efficiency during network training.

Some recent works employ the use of Vision Transformers, such as the Transformer Re-ID (TransReID) [111] that uses a transformer-based framework to capture robust and discriminative features by processing images as sequences of patches, overcoming information loss seen in CNN-based methods. Similarly, the Domain Generalization Person Re-ID (DGReID) [226] proposed a part-aware transformer model that enhances domain generalization by leveraging a transformer architecture that focuses on local parts of images. This is achieved through a proxy task that helps the model learn generic features by comparing local parts of images regardless of their ID labels, thereby reducing domain-specific biases.

3.2.2 Metric Learning

Distance measures are fundamental and aim to present high distances for feature vectors of distinct individuals and low values for representations of the same individual [134]. Initially, most systems used the Euclidean distance (also known as l_2 distance). However, the results using this approach are not always effective.

As a way to incorporate training techniques that consider labeled data, metric learning methods have been proposed. The most common formulation is based on Mahalanobis distance functions, which generalize the Euclidean distance using linear scales and rotations in the feature space. In this case, the squared distance δ between two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ can be written as:

$$\delta(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 - \mathbf{x}_2)^T M_p (\mathbf{x}_1 - \mathbf{x}_2), \tag{3.1}$$

where M_p is a positive semi-definite matrix. Equation 3.1 can even be used in convex optimization problems as proposed in [156].

In the context of Re-ID, one of the most well-known methods for metric learning is the keep-it-simple-and-straightforward (KISSME) [156], which is based on Equation 3.1. The idea involves determining how similar the elements within a pair (i, j) are using a likelihood ratio test. The pairwise difference is applied ($x_{i,j} = x_i - x_j$) and modeled according to a Gaussian distribution with zero mean.

Moreover, several other techniques are proposed based on Equation 3.1, such as large margin nearest neighbor learning (LMNN) [360], which is based on nearest neighbor classification. In this method, a perimeter for the neighbors is defined. Those within the region are considered neighbors, and those outside the region are considered as not belonging to the same class.

In addition to distance learning methods, there are also those based on feature subspace learning. In [184], the learning of a projection w to a lower-dimensional space is proposed, following a procedure similar to what occurs in linear discriminant analysis (LDA) [126]:

$$J(w) = \frac{w^T S_b w}{w^T S_w w}, \tag{3.2}$$

where S_b and S_w are the between-class and within-class scatter matrices. After that, a distance learning step is performed with KISSME.

In other cases, some methods, instead of using Mahalanobis distance, use classifiers such as support vector machines (SVM). Besides these traditional approaches, there are also techniques based on deep learning [446], such as in [391], where the input image is partitioned into three overlapping horizontal parts that are subjected to a convolutional layer and subsequently a fully connected layer that fuses the data and returns a vector to represent the image. Another example is in [386], which performs metric learning by minimizing the distances of similar pairs and vice versa using a network with Inception architecture [302], employing a global loss function in addition to the local loss to regularize the network.

Recently, most research has focused on improving the generalization and efficiency of person re-identification through deep metric learning [4, 81, 446, 400]. Current methods often face limitations such as high memory and computational costs associated with classification parameters or class memory, and the inefficiency of random sampling methods like the PK sampler [134, 400, 114], a popular random sampling method Re-ID. Among the recent approaches, there is a new mini-batch sampling method called Graph Sampling (GS) [186]. GS constructs a nearest neighbor relationship graph for all classes at the beginning of each epoch and forms mini-batches from a randomly selected class and its nearest neighbors. This approach aims to provide more informative and challenging examples, enhancing learning efficiency and performance. However, as with most of the approaches in this category, GS requires supervised training.

3.2.3 Evolution of the State-of-the-Art

It is important to highlight that the Re-ID literature comprises an extensive set of relevant works, which are being produced at an increasingly rapid pace, often making it difficult to provide a comprehensive view of all existing methods. This is mainly due to it being an area of intense research and also to the significant advances in the fields of machine learning and CBIR systems in recent years [137].

Among the main approaches mentioned, a large part of the current state-of-the-art consists of Convolutional Neural Networks (CNNs) [437] and, more recently, Vision Transformers (ViT) [111]. Re-training of architectures previously trained on the ImageNet dataset [70] is commonly performed for Re-ID applications [390, 304]. However, one of the biggest bottlenecks for training neural networks in Re-ID is the lack of labeled data, so networks that require few labels, such as some implementations of siamese networks [426], are applied in most cases.

There are different variations of approaches that employ CNNs in this scenario [426]. In [391], the input image is partitioned into three overlapping horizontal parts that are subjected to a convolutional layer and then a fully connected layer that fuses the data and returns a vector to represent the image. In the case of [340], long short-term memory (LSTM) is used in conjunction with a Siamese network. The LSTM processes parts of the image sequentially so that spatial connections can be memorized to increase the discriminative capacity of the networks.

In [339], it is proposed to use a gating function after each convolutional layer to capture subtle variations when a pair of images is provided to the network. In [196], it is proposed to integrate an attention model based on siamese networks to emphasize the local features of the images. In [365], a method based on low-level features is presented, including color histograms, texture, and bag of visual word approaches (such as SIFT and SURF), which are aggregated into a Fisher vector for each image. The obtained vectors are subjected to dimensionality reduction and subsequently used for training convolutional neural networks.

Furthermore, as an alternative to deep learning, re-ranking methods have started to be applied to post-process the results obtained in the retrieval stage, and promising outcomes have been reported in the literature [426, 137, 429]. Besides the crescent use of re-ranking for Re-ID, such approaches have been little explored in Re-ID when compared to deep learning techniques and require further study [390]. These methods are advantageous due to their capacity to improve the results provided by deep learning models. In general terms, in [429], re-ranking is applied based on the idea of encoding the k nearest neighbors into a single vector and re-ranking them using the Jaccard correlation metric. The modeling of the data to be learned is also a topic of great relevance, which is explored in [366], where Graph Neural Network (GNN) representations are employed to learn features, especially in scenarios with occlusions.

Other approaches that are gaining a lot of attention and are of fundamental importance, especially in person recognition in videos, are pose estimators and gait recognition. Among the main works, DeepPose [319] can be cited, which uses convolutional neural networks to classify different people by pose through the extraction of a skeleton of points from the human body. A method that uses LSTM and residual networks for this purpose was also proposed [58], which provided even more effective results. Another pose estimator is the PifPaf [149] that estimates poses by applying two layers: one to locate the most discriminative parts of the body, the Part Intensity Field (PIF); and another to associate body parts with each other (matching), the Part Association Field (PAF).

This work focuses on unsupervised methods for Re-ID, which, although often more challenging, present a promising solution in many scenarios where labeled data is scarce. Some unsupervised approaches apply hierarchical clustering and modifications to the loss function used by the neural network to generate more effective pseudo-labels for self-supervised training, as in the case of the Hierarchical Clustering with Hard-batch Triplet Loss (HCT) method [402]. Also, for the generation of pseudo-labels, some authors have used Generative Adversarial Networks (GANs) to perform domain-adaptive training [144]. A network trained to generate augmentations for a pedestrian image, enabling the extension of training collections and transfer learning capability, was also proposed [407]. In [404], images are divided into clusters, and augmentations are performed to reduce the distance between elements of the same group and vice versa.

Among the most recent approaches in unsupervised Re-ID, we can mention the Framework for Transferable Representations of Pedestrians (VAL-PAT) [23] that enhances pedestrian analysis by learning transferable representations through self-supervised contrastive learning, image-text contrastive learning, and multi-attribute classification. Also employing contrastive learning, the Offline-Online Associated Camera-Aware Proxies (O2CAP) [354] is a clustering-based approach with camera-aware proxies that splits clusters based on camera views to better manage intra-ID variance and inter-ID similarity. This approach employs offline and online proxy-level contrastive learning losses to associate proxies and reduce noise from delayed pseudo-label updates. In contrast, the Discriminative Identity-Feature Exploring and Differential Aware Learning (DIDAL) [201] addresses intra-instance redundancy using synthetic complementary attention and GNNs. This method extracts and models discriminative identity features and is evaluated on Re-ID and vehicle re-identification datasets.

Table 3.1 summarizes the evolution of unsupervised state-of-the-art Re-ID methods in the literature since 2017, with results on three datasets. More detailed information about Re-ID datasets and effectiveness measures (R1 and MAP) is provided in Chapter 4 (Sections 4.1 and 4.2). The methods that had some type of training with labeled data were trained on another dataset, typifying transfer learning. The abbreviations in parentheses indicate the datasets used for training [1]. For example, the use of (D, M) indicates that training was performed on the DukeMTMC dataset (source) and testing was performed on the Market1501 dataset (target) or vice versa. In total, there are 33 methods divided into 4 distinct categories, which were presented in the terminology discussion introduced

[1] C03 = CUHK03, M = Market1501, D = DukeMTMC, MT = MSMT17.

in Section 2.4.1. The MAP and R1 values are presented according to the values reported in each publication.

Comparisons with state-of-the-art Re-ID methods are discussed in different chapters, each considering the methods available at the time of their respective publications. A comprehensive discussion comparing the approaches proposed in this work with all the methods in Table 3.1 is presented in Chapter 11.

Table 3.1 – State-of-the-art methods in Re-ID with results of MAP (%) and R1 (%).

Method	**Year**	**Datasets**					
		Market1501		**DukeMTMC**		**CUHK03**	
		R1	**MAP**	**R1**	**MAP**	**R1**	**MAP**
Unsupervised Methods							
ARN [181]	2018	70.3	39.4	60.2	33.4	—	—
EANet [118]	2018	66.4	40.6	45.0	26.4	51.4	31.7
TAUDL [170]	2018	63.7	41.2	61.7	43.5	44.7	31.2
ECN [431]	2019	75.1	43.0	63.3	40.4	—	—
MAR [397]	2019	67.7	40.0	87.1	48.0	—	—
UTAL [171]	2019	69.2	46.2	62.3	44.6	56.3	42.3
SSL [189]	2020	71.7	37.8	52.5	28.6	—	—
HCT [402]	2020	80.0	56.4	69.6	50.7	—	—
CAP [353]	2021	91.4	79.2	81.1	67.3	—	—
IICS [376]	2021	89.5	72.9	80.0	64.4	—	—
RLCC [415]	2021	90.8	77.7	83.2	69.2	—	—
ICE [43]	2021	93.8	82.3	83.3	69.9	—	—
MGH [368]	2021	93.2	81.7	83.7	70.2	—	—
MGCE-HCL [297]	2022	92.1	79.6	82.5	67.5	—	—
MCRN [367]	2022	92.5	80.8	83.5	69.9	—	—
O2CAP [354]	2022	92.5	82.7	83.9	71.2	—	—
DIDAL [201]	2023	94.2	84.8	—	—	—	—
VAL-PAT [23]	2023	—	—	86.1	74.9	—	—
Domain Adaptative Methods							
HHL (D,M) [430]	2018	62.2	31.4	46.9	27.2	—	—
HHL (C03) [430]	2018	56.8	29.8	42.7	23.4	—	—
ATNet (D,M) [197]	2019	55.7	25.6	45.1	24.9	—	—
CSGLP (D,M) [273]	2019	63.7	33.9	56.1	36.0	—	—
ISSDA (D,M) [306]	2019	81.3	63.1	72.8	54.1	—	—
ECN++ (D,M) [432]	2020	84.1	63.8	74.0	54.4	—	—
MMCL (D,M) [348]	2020	84.4	60.4	72.4	51.4	—	—
JVCT+ (D,M) [44]	2021	90.5	75.4	81.9	67.6	—	—
MCRN (D,M) [367]	2022	93.8	83.8	84.5	71.5	—	—
Cross-Domain Methods (single-source)							
EANet (C03) [118]	2018	59.4	33.3	39.3	22.0	—	—
EANet (D,M) [118]	2018	61.7	32.9	51.4	31.7	—	—
SPGAN (D,M) [71]	2018	43.1	17.0	33.1	16.7	—	—
DAAM (D,M) [121]	2019	42.3	17.5	29.3	14.5	—	—
AF3 (D,M) [195]	2019	67.2	36.3	56.8	37.4	—	—
AF3 (MT) [195]	2019	68.0	37.7	66.3	46.2	—	—
PAUL (MT) [380]	2019	68.5	40.1	72.0	53.2	—	—
Cross-Domain Methods (multi-source)							
CAMEL [396]	2017	54.5	26.3	—	—	31.9	—
EMTL [370]	2018	52.8	25.1	39.7	22.3	—	—
Baseline by [153]	2019	80.5	56.8	67.4	46.9	29.4	27.4

3.3 Query Performance Prediction

Query performance prediction (QPP) [233] is a challenging task that consists of predicting the quality of results generated by an IR system. The key challenges [262] include: *(i)* most methods lack generalization for different evaluation scenarios since the quality of queries can vary significantly; *(ii)* in the absence of labeled information, predicting the quality of a query is a complex task requiring methods to exploit different patterns and relationships within the data, which enforces the need for modeling the information appropriately; *(iii)* there is no widely used protocol for evaluating these approaches, but QPP methods in image search often use AP effectiveness measures and the Pearson coefficient.

There are two main categories of QPP approaches: pre-retrieval and post-retrieval [262]. Pre-retrieval QPP methods [56, 374] estimate the effectiveness of a search query prior to the retrieval of any documents. These methods can rely on analyzing the query itself, considering factors such as query length, term specificity, and term frequency within a corpus. This work focuses on post-retrieval QPP methods [243, 248]. In contrast to pre-retrieval methods, post-retrieval methods predict the quality of the query after the retrieval process has taken place. They utilize information from the retrieved documents, such as relevance scores, document features, and feedback mechanisms, to assess the quality and effectiveness of the query in producing relevant results.

Initially proposed in ad-hoc text retrieval [56], such approaches also attracted the attention of the image retrieval research community [179, 374, 233, 266, 248], assuming a diverse taxonomy as query difficulty prediction [374], query difficulty estimation [179], and effectiveness estimation [266, 248]. Despite recent advances, query performance prediction in CBIR remains a scarce and largely unexplored task when compared to text retrieval [262].

One of the earliest contributions to QPP in the image domain [374] utilized query words to calculate a set of four text-based pre-retrieval features and trained a model for QPP in image retrieval. However, later research shifted focus towards post-retrieval predictors [262]. Some works employed textual queries to perform post-retrieval QPP image retrieval in different scenarios [313, 314], including web image search [314]. Among other research directions, some approaches introduced post-retrieval predictors that categorize the retrieved images into pseudo-positive and pseudo-negative groups using pseudo-relevance feedback [131, 130]. Then, a voting scheme is employed to determine the relevance of these images. These pseudo-relevance labels are subsequently used to estimate the Average Precision (AP).

Given the importance of appropriately modeling information in these scenarios, the Authority [243] and Reciprocal Density [248] measures were proposed based on graph-based formulations of ranking information. They predict a score that estimates the effectiveness of ranked lists of image queries in a completely unsupervised manner based on the cluster hypothesis [155]. The cluster hypothesis considers that the images belonging to a highly effective ranked list should appear in the ranked lists of each other.

With the rise of deep learning, an approach [300] proposed to transform the ranked list of images into a similarity or correlation matrix, which is then fed into a Convolutional Neural Network (CNN) regression model for retrieval quality evaluation. However, the method is supervised.

In light of this discussion, this work proposes two novel self-supervised approaches for QPP in image retrieval: Deep Rank Noise Estimator (DNRE) and Regression for Query Performance Prediction Framework (RQPPF), both presented in Chapter 5. In addition to using synthetic data for training, these methods also incorporate innovative modeling techniques to leverage ranked list data and exploit contextual information.

While the RQPPF models meta-features to train different regression models, DRNE employs a denoising CNN on images computed from ranked lists. The DRNE transforms ranked lists, which are numeric data, into visual images for processing by denoising convolutional neural networks. This is a challenging and also relatively unexplored task. This is an innovative approach in comparison to related works [179, 374, 233, 266, 248]. In the literature, there are some works that proposed strategies to transform non-image data into images. In [280], the authors proposed different approaches for creating images from feature vectors, like creating images of bar graphs and gray images where blacker pixels represent low distances and whiter pixels represent higher distance values.

There is also the DeepInsight approach [281] which is a very recent and promising technique. It consists of mapping all the features into a 2D space using a dimensionality reduction technique (e.g. t-SNE [214], kPCA). After the distribution is learned, the image is cropped according to its convex hull (the smallest rectangle where all the data points fit). The data points are represented according to the learned distribution and the differences in color are given according to differences in feature values. This approach can be used for different classification tasks where the datasets are not composed of images (e.g. text, audio, signals).

Regarding signal processing, which consists of a unidimensional data stream, a possible representation for analyzing this data is the use of recurrence matrices [357]. This can be used to create images in order to analyze recurrent patterns between systems and functions. This technique provides a wide range of applications.

The proposed DRNE relies on the idea of noise removal from images that represent similarity information encoded in ranked lists, analogous to the approach performed in [246]. However, in DRNE, we train a denoising deep learning network, pairing the ranked list image to its MAP (Mean Average Precision) to obtain a score related to its effectiveness.

In this way, we exploit the denoising network in a query performance prediction task.

Among the most relevant state-of-the-art deep denoisers, we can cite the DnCNN [409] (Denosing Convolutional Neural Network) which can learn noise patterns from pairs of clean and noisy images. The deep denoisers generally have the advantage of being capable of learning different noise patterns without requiring high execution times for parameter adjusting or image processing, like in most of the statistical approaches (e.g. BM3D [57]). There are also more recent approaches, like RDNN [417] (Residual Dense Neural Network) which was originally proposed for image super-resolution but can also be employed for denoising tasks. More recently, there is the DRUnet [408], a variant of the UNet network employed for denoising. The cited residual networks, besides being more effective, generally tend to be less efficient regards time and more memory-consuming when compared to DnCNN.

For training denoisers, the lack of clean image data to be used as groundtruth may be a challenge for certain applications such as medical imaging and remote sensing [158, 269, 223, 290]. In this scenario, different training strategies were proposed. The Noise2Noise [164] and Noisier2Noise [223] approaches involve training with pairs of noisy images, where the clean image can be predicted by learning common patterns in both images which are supposed to be present in the clean image. There is also the Noise2Void [152] where the learning process is done with only corrupted or noisy images, and the noisy pattern is learned considering the given dataset. There are also other strategies like the one based on Stein's unbiased risk estimator (SURE) [290] which proposes an MSE (Mean Squared Error) unsupervised estimation which can be used during training. Among the self-supervised strategies, there are some that implement a CNN with a “blind spot” in the receptive field of the network [158] and others that generate the groundtruth data using the most promising statistical methods (e.g. BM3D [57]) in order to train a network like DnCNN for example [290].

3.4 Semi-Supervised Classification and Graph Convolutional Networks

This section presents an overview of the methods proposed for semi-supervised classification, including recent approaches and their main ideas, with a particular focus on image data and deep learning techniques.

Semi-supervised approaches perform training considering both labeled and unlabeled data, which is advantageous in multiple scenarios where there is little labeled data [85]. Some of them rely on the generation of pseudo-labels [338]. Among the traditional methods for generating pseudo-labels, we can cite: Label Spreading [433] and Pseudo-label [162]. There are also several supervised approaches that later presented semi-supervised variants that do not require the generation of pseudo-labels. For example: Support Vector Machines [54] (SVM) and Optimum Path Forest [8] (OPF).

The taxonomy and categories of semi-supervised approaches vary in the literature [85, 338]. Generally, there is some overlap among categories. In the following subsections, we present them according to 4 research directions [338]: category regularization; stronger augmentation; convergence with self-supervised learning; and graph-based approaches.

- **Consistency regularization**

  These methods rely on a concept known as category regularization. The central idea is to force the approach to produce similar results for augmented versions of the same unlabeled image. This is generally done by considering an additional term in the loss function. The first method as far as it is known, to use this concept is called II-Model [157]. In II-Model, they use translation and random horizontal flips as augmentations for unlabeled data, which is often called weak augmentation.

  However, the main issue with II-Model is the unstable target, which compromises the algorithm learning procedure. The Mean Teacher [309] approach was proposed with the intent to address this issue. For this, they use two separate models: the Student network and the Teacher network. While the Student is trained as usual, the Teacher does not use back-propagation, and the weights are updated at each iteration using the weights from the Student network.

- **Stronger Augmentation**

  Data augmentation is of crucial importance for various semi-supervised approaches [338]. Some strategies focus on improving the performance of classification by employing different kinds of data augmentation techniques, in such a way that the inputs given to the two branches of the neural model (or, to the two separate networks) are sufficiently distinct. There are many methods that fit in this category, among them: Virtual Adversarial Training and Entropy Minimisation [220] (VAT), Unsupervised Data Augmentation [22] (UDA), MixMatch [28], FixMatch [289], ReMixMatch [27], AlphaMatch [101]. Some of them also mix other ideas, such as the concept of consistency regularization.

- **Convergence with Self-Supervised Learning**

  Recently, self-supervision has been used by several semi-supervised methods. Self-supervised approaches are a category of representation learning algorithms capable of generating supervision signals without any human annotations. Most approaches in this category use self-supervision to generate a set of pseudo-labels for training. Among the main approaches in this category, we can cite: SimCLR [47], CoMatch [169], Self-Match [145].

- **Graph-based Approaches**

A promising research direction is methods based on graphs. There are different traditional graph-based approaches, both transductive and inductive ones [85]. The idea is that the elements of the dataset can be represented as nodes and the edges can be used to propagate or represent some kind of information between these nodes. Graph-based methods are usually based on the manifold assumption [85]: the graphs, constructed based on the local similarity between data points, provide a lower-dimensional representation of the potentially high-dimensional input data. This makes these approaches advantageous for scenarios with data of high dimensionality.

Recently, Graph Convolutional Networks (GCN), have been proposed for semi-supervision. While CNNs are specially built to operate on regular (Euclidean) structured data, the GNNs work on graphs with different numbers of vertexes and unordered nodes (irregular on non-Euclidean structured data). There are many variants of GCNs proposed: GCN-Net [146], GCN-SGC [363], GCN-GAT [344], GCN-APPNP [147], GCN-ARMA [30]. Also, variants of GNNs: GNN-LDS [90], GNN-KNN-LDS [90].

The GCNs exploit feature vectors and graph-based neighborhood structures to learn more effective representations [141, 135]. Due to these aspects, the GCNs have been recently applied to graph-based data on semi-supervised learning tasks, achieving state-of-the-art results. Several GCN variations have been proposed with relevant results [344, 363, 147, 30]. The use of GCN has many different applications. There are some recent works that exploit graph learning for question and answer systems [228], including conversational image search [227].

However, there are still not many approaches that use GCNs in image classification [274, 343, 307]. Among the multiple research topics, there is finding the best approach to model the graph and the features, which are provided as the input for these networks and directly impact their performance and results.

3.5 Contrastive Learning

Traditional methods, such as the cross-entropy loss, focus primarily on achieving correct classifications but may not always encourage the learning of robust, discriminative features that generalize well to new, unseen data, among other issues (e.g., lack of robustness to noisy labels [418, 296], possibility of poor margins [84, 199]). In light of this, the contrastive losses, that aim to differentiate between similar and dissimilar data points, are a promising solution [47, 143]. Many recent works have been using contrastive loss for diverse applications: self-supervised facial expression recognition [286], blind video restoration [218], self-supervised Vision Transformers [221], and many others [107].

The Simultaneous Contrastive Learning of Representations [47] (SimCLR), a pioneer in the field of self-supervised learning, was proposed for learning visual representations by maximizing the agreement between differently augmented views of the same image through a contrastive loss in the latent space. This method significantly contributed to the field by facilitating the training of more robust and generalizable features without relying on labeled data. Although SimCLR [47] offers promising results, it is not capable of exploiting labeled data because the method is entirely unsupervised. Considering this issue, the Supervised Contrastive Learning [143] (SupCon) was proposed to extend the principles of SimCLR by incorporating labels for more discriminative learning in supervised tasks.

Both SimCLR [47] and SupCon [143] leverage pairwise comparisons for effective representation learning. However, this strategy may be limited since it does not consider contextual information [250]. Based on data augmentation, a recent work [10] proposed to enhance document ranking on small datasets of different text document types (news, finance, and science) through supervised contrastive learning. The approach involves augmenting training data by utilizing portions of relevant documents from query-document pairs. This augmented dataset is then used with a supervised contrastive learning objective, differing from traditional pairwise training objectives which did not show improvement with data augmentation.

There are different means of exploiting contextual similarity information in metric learning applications, among them: employing graph approaches [392, 362, 208, 278, 136], data augmentation [10, 93], and using kNN information in parts of the model framework [224, 136, 311, 93]. However, very few incorporate some type of contextual similarity information directly into the contrastive loss formulation. Some examples are the Nearest-Neighbor Contrastive Learning of Visual Representations [82] (NNCLR), the Contextual Loss [183], and the kNN Contrastive Loss [441]. The NNCLR [82] is unsupervised and based on SimCLR. It introduces a loss function that compares an augmentation not with the original element, but with the closest neighbor of that element. Besides its contributions, it strictly uses only a single closest neighbor in the comparison, ignoring other elements present in the neighborhood. By contrast, the Contextual Loss [183] improves similarity prediction by counting the number of neighbors two samples have in common. Conversely, the kNN Contrastive Loss [441] computes the average contrastive loss for an element and its k neighbors and was proposed and designed for classification in dialogue systems, specifically considering out-of-domain (OOD) samples, as opposed to image classification.

4 Experimental Protocol

This chapter outlines the experimental protocol considered for assessing the effectiveness of the seven approaches proposed in this work. Section 4.1 defines the effectiveness measures used to assess the results for each task. Section 4.2 presents the datasets and descriptors used for general-purpose image retrieval, classification, and person Re-ID tasks.

4.1 Effectiveness Measures

This section presents the effectiveness measures employed for evaluating the three tasks considered in this work: image retrieval (Subsection 4.1.1), query performance prediction (Subsection 4.1.2), and image classification (Subsection 4.1.3). Each task is assessed using different measures.

In general-purpose scenarios, the measures are computed for all queries (in the context of retrieval and query performance prediction) or every instance in the testing set (in the context of classification), and then the average is calculated. For person Re-ID, each dataset has a particular query set. The measures widely used in the literature were selected for most evaluations, but some experiments followed the specific protocol defined according to the benchmark.

4.1.1 Retrieval

Effectiveness measures are essential for evaluating the quality of retrieval results [15]. In this subsection, we present the measures employed for evaluating retrieval tasks in this study. All of them consider ranked lists as input. The results of most measures are defined in the $[0, 1]$ range, where higher values represent better results.

- **Precision**

Precision can be understood as the fraction of relevant instances among the retrieved instances, which is calculated as:

$$P_k = \frac{c}{k}, \tag{4.1}$$

where k is the number of retrieved items and c is the number of correct items among the ones retrieved. In the text, P@k denotes the precision at position k.

- **Recall**

Unlike precision, recall is the fraction of relevant instances retrieved over the total number of relevant instances. Please note that the notation R used here does not refer to a ranker, but rather to the recall measure which is defined as:

$$R_k = \frac{c}{n_r}, \tag{4.2}$$

where k is the number of retrieved items, c is the number of correct items among the retrieved items, and n_r is the number of relevant items. The number of relevant items is the maximum number of items that can be correctly identified in a given circumstance. Note that if k is equal to the size of the dataset, the result is always 1. In the text, R@k denotes the recall at position k.

- **Mean Average Precision (MAP)**

The MAP is the most common measure used to assess the effectiveness of ranked lists in retrieval tasks. For each of the ranked lists in the set $\mathcal{T}$, the average precision AP (Average Precision) can be computed. Precision and recall are calculated at each position in the ranked list, generating a curve that describes the function $P(R)$, in which precision is given as a function of recall. For illustrative purposes, Figure 4.1 shows an example of a *precision* × *recall* curve where the area formed by the curve corresponds to the AP.

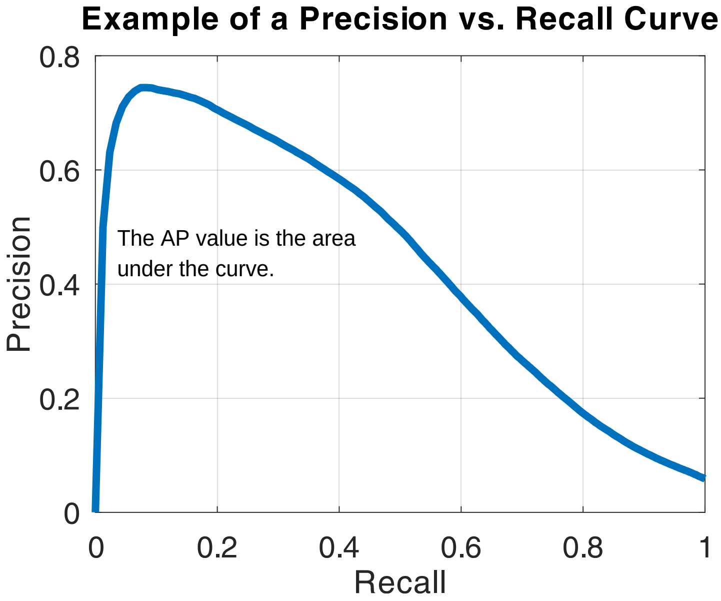


Figure 4.1 – Example of precision × recall curve.

More formally, let q be a query and n_r be the number of relevant items in the dataset for the given query q. Let $\langle r_i \mid i = 1, 2, .., d \rangle$ be a ranked relevance vector of depth d, where r_i indicates if the ith item is either 0 (not relevant, different from query class) or 1 (relevant, same class as the query), the AP is defined as follows:

$$AP_q = \frac{1}{n_r} \sum_{i=1}^{d} \left(\frac{r_i}{i} \sum_{j=1}^{i} r_j \right). \tag{4.3}$$

MAP is obtained by taking the average of the average precision (AP) of each of the ranked lists in the set $\mathcal{T}$. Let Q be the number of queries, the MAP is defined as follows:

$$MAP = \frac{\sum_{l=1}^{Q} AP\left(q_l\right)}{Q}. \tag{4.4}$$

For most cases, all images of the datasets are considered as queries when computing the MAP, with some exceptions: Holidays [127] dataset and Re-ID [429, 422, 428] datasets have a specific set of queries.

- **Cumulative Matching Characteristics (CMC)**

The CMC curves are commonly used to assess the effectiveness of person re-identification methods. Given the ranked list of a query q, a score $I_q(k)$ for position k is calculated using the following criteria:

$$I_q(k) = \begin{cases} 1 & \text{if at least one image from the same class is in the top-}k\text{ positions,} \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

Thus, $I_q(k)$ indicates the presence (1) or absence (0) of any correct match within the top k positions. Each position k on the CMC curve is given by the average of the $I_q(k)$ scores calculated for each query.

The value k on a Cumulative Match Characteristic (CMC) curve is denoted as Rk. It represents the likelihood that at least one element from the same class appears within the first k positions of ranked lists. The R1 is one of the most commonly used for Re-ID and corresponds to the first value of the CMC curve. It indicates the number of ranked lists where the image of the same individual appears in the top position following the query image, being equivalent to Precision@1 in this case. Despite the similarity in notation, we should not confuse R1, from the CMC curve, with recall, which is denoted as R@1.

It is important to note that protocols may vary; for instance, certain studies suggest that images of the same class taken with the same camera should not be considered as belonging to the same class [176, 422]. Additionally, variations exist in multi-query scenarios [422, 428]. This dissertation specifically evaluates Re-ID datasets in single-query, multi-shot scenarios.

- **NS-Score**

Other specific measures may be used depending on the dataset. In this work, the NS-Score, or simply NS, is considered for the UKBench [230] dataset. The authors of the dataset proposed this measure, and it is extensively employed in the literature for this dataset, which counts the number of relevant images in the top 4 positions. Since 4 is the number of relevant images per class, a score equal to 4 is a perfect score. Of all the retrieval effectiveness measures, this is the only one that has a different range, which is $[0, 4]$.

4.1.2 Query Performance Prediction

Query performance prediction (QPP), which is the task of estimating the effectiveness of a ranked list can be understood as a subtopic of image retrieval. To evaluate the quality of a QPP approach, we measure the Pearson correlation between the predicted scores and one of the measures that use labels (i.e., Precision, Recall, or MAP). The Pearson correlation coefficient measures the linear correlation between two variables X and Y. It is mathematically defined as:

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}. \tag{4.6}$$

Where:

- r is the Pearson correlation coefficient.
- X_i and Y_i are the individual sample points indexed with i.
- $\bar{X}$ and $\bar{Y}$ are the means of the X and Y datasets respectively.
- n is the number of data points.

This formula quantifies the degree to which a relationship between the two variables can be described by a line. The value of r ranges from -1 to +1, where +1 indicates perfect positive correlation, -1 indicates perfect negative correlation, and 0 indicates no linear correlation. In this dissertation, X_i is the value predicted by a QPP approach and Y_i is the real value of an effectiveness measure (e.g., MAP or Precision). For example, the better the prediction correlates with the MAP score, the more effective the QPP task is.

4.1.3 Classification

This subsection presents all the classification measures used in this work. They are described using the following concepts:

- True positives (TP): number of correct predictions where instances are accurately identified as belonging to a specific class.
- True negatives (TN): number of correct predictions where instances are accurately identified as not belonging to a specific class.
- False positives (FP): number of incorrect predictions where instances are wrongly identified as belonging to a specific class.
- False negatives (FN): number of incorrect predictions where instances are wrongly identified as not belonging to a specific class.

Since the considered datasets are multi-class, each of these is calculated separately for every class and aggregated to evaluate the overall effectiveness of the classifier.

- **Accuracy (Acc)**

Accuracy is defined as the ratio of correctly predicted observations to the total observations. It can be expressed as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{4.7}$$

- **F-Measure (F1 Score)**

F-Measure is the harmonic mean of precision and recall:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.8}$$

For retrieval, we defined precision and recall considering the ranked list definition. To facilitate the understanding of the reader, these measures can also be equivalently defined as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4.9}$$

In this work, we considered the macro-average F-Measure. It consists in computing a separate F-Measure for each class and then averaging these scores to get an overall measure. This does not take class imbalance into account.

4.2 Datasets and Descriptors

A wide variety of datasets was considered, totaling 17: 13 of general-purpose and 4 of Re-ID. Since general-purpose cases are broader scenarios, a wider number of datasets was employed. For Re-ID, there is often a limited number of datasets available. In this work, the most widespread ones were considered. All datasets are multi-class and single-label, featuring various classes with each image uniquely belonging to only one class.

4.2.1 General-Purpose

Table 4.1 shows all the 13 general-purpose datasets used in the experiments and their information. For each dataset, the evaluation measures for query performance prediction, retrieval, and classification are included. Notice that some datasets were used exclusively for retrieval, some exclusively for classification, and some for both.

The descriptors used for the general-purpose scenarios and their respective MAP values for each dataset are detailed in Table 4.2. Datasets that were exclusively used for classification were not included. A wide range of descriptors were used (global, local, and deep learning). The Euclidean distance was employed in all the cases to obtain the ranked lists from the extracted features. All the CNNs were trained in the ImageNet dataset with a PyTorch implementation [2] and used in the target dataset to extract features. The same is valid for the Vision Transformers, but considering other public implementations [3]. The MAP computation considered all the images as queries, except for Holidays where the protocol specifies a particular set of queries [127]. Notice that the MAP values can also be found in the respective chapters where the experimental evaluations of the methods were conducted.

Table 4.1 – General-purpose datasets used in the experimental evaluation.

Dataset	**Num. of Classes**	**Dataset Size**	**Evaluation Measures**		
			QPP	**Image Retrieval**	**Image Classification**
ORL Faces [**116**]	40	400	—	Recall@15	—
Flowers [**229**]	17	1,360	Pearson(MAP)	MAP	Acc., F-Measure
MPEG-7 [**161**]	70	1,400	Pearson(MAP)	MAP, Recall@40	—
Holidays [**127**]	500	1,491	—	MAP	—
Brodatz [**35**]	16	1,776	Pearson(MAP)	MAP	—
Corel5k [**194**]	50	5,000	—	MAP	Acc., F-Measure
UKBench [**230**]	2,550	10,200	—	NS, MAP	—
CUB200 [**346**]	200	11,788	—	—	Acc., F-Measure
Dogs [**142**]	120	20,580	—	MAP	—
CIFAR-100 [**150**]	100	60,000	—	—	Acc.
MiniImageNet [**345**]	100	60,000	—	—	Acc.
ALOI [**98**]	1,000	72,000	—	MAP	—
Food101 [**33**]	101	101,000	—	—	Acc.

[2] CNNs: `https://github.com/Cadene/pretrained-models.pytorch`

[3] VIT-B16: `https://github.com/faustomorales/vit-keras`
T2T-VIT: `https://github.com/yitu-opensource/T2T-ViT`
SWIN-TF: `https://github.com/rishigami/Swin-Transformer-TF`

Table 4.2 – Descriptors used for general-purpose datasets.

Category	**Type**	**Descriptor**	**MAP (%)**							
			MPEG-7	**Brodatz**	**Flowers**	**Corel5k**	**UKBench**	**Holidays**	**Dogs**	**ALOI**
Global	**Color**	ACC [119]	—	—	18.99	23.44	87.72	64.29	—	—
		SPACC [119, 209]	—	—	19.20	23.86	85.30	62.37	—	—
		CLD [53]	—	—	18.54	17.86	59.58	37.59	—	—
		SCD [53]	—	—	10.25	14.56	83.04	54.26	—	—
		SCH [53]	—	—	13.43	17.56	48.98	24.19	—	—
		FOH [337, 209]	—	—	11.42	15.87	57.05	25.77	—	—
		BIC [295]	—	—	25.56	—	80.46	—	—	—
	Shape	PHOG [59, 209]	—	—	14.74	15.80	41.60	31.15	—	—
		AIR [103]	89.39	—	—	—	—	—	—	—
		ASC [191]	85.28	—	—	—	—	—	—	—
		IDSC [190]	81.70	—	—	—	—	—	—	—
		CFD [244]	80.71	—	—	—	—	—	—	—
		BAS [13]	71.42	—	—	—	—	—	—	—
		SS [317]	37.82	—	—	—	—	—	—	—
	Texture	LBP [231]	—	48.40	10.34	14.83	47.19	28.82	—	—
		SPLBP [231, 209]	—	—	10.92	15.41	52.14	33.09	—	—
		EHD [215]	—	—	12.46	16.80	44.10	25.83	—	—
		CCOM [148]	—	57.57	—	—	—	—	—	—
		LAS [308]	—	75.15	—	—	—	—	—	—
	Color and Texture	CEDD [40]	—	—	20.48	23.00	70.45	51.59	—	—
		SPCEDD [40, 209]	—	—	21.94	28.70	74.98	56.09	—	—
		FCTH [41]	—	—	20.56	23.93	73.70	48.44	—	—
		SPFCTH [41, 209]	—	—	21.73	26.43	77.78	55.43	—	—
		JCD [401]	—	—	20.89	24.73	74.85	52.84	—	—
		SPJCD [401, 209]	—	—	22.56	28.02	76.67	56.58	—	—
		COMO [342]	—	—	21.83	21.05	79.77	49.66	—	—
	Holistic	GIST [232]	—	—	9.82	15.98	45.44	21.59	—	—
Local	**Bag of Words**	SIFT [205]	—	—	28.47	12.60	74.52	54.63	—	—
		VOC [356]	—	—	—	—	91.14	—	—	—
Deep Learning	**CNN**	CNN-SENet [117]	—	—	43.16	56.92	92.15	71.60	—	78.41
		CNN-ResNet [110]	—	—	51.83	64.81	94.54	74.88	63.73	81.97
		CNN-FBResNet [110]	—	—	52.56	64.21	93.88	72.65	—	—
		CNN-ResNeXt [372]	—	—	51.91	62.39	93.67	74.16	—	—
		CNN-DPNet [51]	—	—	50.93	65.15	90.47	70.59	—	79.09
		CNN-VGGNet [198]	—	—	39.05	47.85	87.99	67.96	—	—
		CNN-BnVGGNet [198]	—	—	41.87	52.72	89.24	67.60	—	—
		CNN-InceptionV4 [302]	—	—	42.35	58.66	86.82	63.84	—	—
		CNN-InceptionResNet [302]	—	—	42.20	61.17	87.23	62.87	—	—
		CNN-BnInception [122]	—	—	46.58	46.60	91.84	70.06	—	—
		CNN-NASnet-Large [445]	—	—	40.74	53.55	86.90	64.48	—	—
		CNN-AlexNet [151]	—	—	46.04	37.67	85.57	65.25	—	—
		CNN-Xception [52]	—	—	47.31	54.44	90.83	64.94	—	76.07
	Hybrid Network	CNN-OLDFP [222]	—	—	—	—	97.74	88.46	—	—
	Transformers	VIT-B16 [77]	—	—	87.12	74.19	—	—	79.83	79.40
		T2T-VT24T [399]	—	—	38.03	58.97	—	—	—	76.90
		SWIN-TF [202]	—	—	—	73.92	97.93	85.52	45.54	—

4.2.2 Person Re-Identification

In this work, we consider a total of 4 Re-ID datasets, which are detailed in Table 4.3. For the CUHK03 [176] dataset, the *detected* version was considered, which uses the bounding boxes extracted with the DPM [173] detector and follows the experimental protocol proposed by [429]. For the Market1501 [422] and DukeMTMC [428] datasets, the protocol adopted is the one proposed by the authors of the datasets. To keep it brief, the DukeMTMC and Market1501 datasets are often referred to simply as Duke and Market. For the dataset Airport [137], a different protocol was adopted, where all the images (training, test, and gallery) were treated as query images and gallery simultaneously. In all cases, the MAP was reported with R1 for comparison. The measure R1 corresponds to the first value of the CMC curve, which indicates the number of ranked lists that have at least one image corresponding to the same class in the first position after the query image. All evaluations follow a single-query protocol, where each query considers only one image at a time. Additionally, all datasets are multi-shot, containing multiple images of each individual.

Table 4.3 – Re-ID datasets used the experimental evaluation.

Dataset	**N. Classes**	**Size**	**Train**	**Galery**	**Test**	**Cam.**	**Detector**
CUHK03 [176, 429]	1,467	14,097	7,365	5,332	1,400	2	DPM
Market1501 [422]	1,501	32,217	12,936	15,913	3,368	6	DPM
DukeMTMC [428]	1,812	36,411	16,522	17,661	2,228	8	Manual
Airport [137]	9,651	39,902	—	—	—	6	ACF

Table 4.4 presents the descriptors employed for the Re-ID datasets in the experimental evaluation. The number of descriptors per dataset varies from 21 to 28, consisting of different types (i.e., traditional, bag of words, deep learning). For most of the extractions, the Euclidean distance was considered. The only exception is the descriptor OSNET-AIN, where the cosine distance was employed, as done by the authors of the model [436]. The process of obtaining the descriptors is entirely unsupervised. We do not consider the labels of the dataset being evaluated. For the non-deep descriptors (GBICOV, LOMO, GOG, WHOS, ELF, HLBP, SDC e BOVW), the Principal Component Analysis (PCA) was employed to reduce the features vectors to 100 positions before computing the ranked lists, as done in the literature [137].

The convolutional neural networks (CNN) were trained on different datasets, which are indicated by the abbreviations in parentheses. Most were pre-trained on the ImageNet [70] dataset, except HACNN which was trained from scratch on the Re-ID datasets. The MSMT17 [359] was utilized to train most of the models, since it's a large dataset (126,441 images of 4,101 people in 15 cameras), facilitating generalization. The majority of the networks were trained considering only the specified training subset, however, some used all the images (train, gallery, and queries) of the MSMT17 dataset: RESNET50, OSNET, OSNET-IBN e OSNET-AIN. To keep the protocol completely unsupervised, we ensured that none of the networks were trained on the target dataset. The features were extracted with the pre-trained weights [4] available on Torchreid [435].

Throughout the text when a Re-ID descriptor is mentioned without specifying the database on which it was trained, it refers to training on MSMT17 (MT). This standard was adopted to facilitate reading.

[4] Torchreid: `https://kaiyangzhou.github.io/deep-person-reid/MODEL_ZOO.html`

Table 4.4 – Values of MAP (%) and R-01 (%) for each Re-ID descriptor on each dataset. The dataset used to train the model is described between parentheses (M = Market, D = DukeMTMC, MT = MSMT17).

	Datasets						
	CUHK03		**Market1501**		**DukeMTMC**		**Airport**
Descriptors	**R1**	**MAP**	**R1**	**MAP**	**R1**	**MAP**	**MAP**
GBICOV [213]	0.63	0.82	10.21	3.27	—	—	—
LOMO [184]	0.79	0.89	19.15	6.46	6.60	2.82	35.35
GOG [217]	0.49	0.77	21.56	7.55	10.82	4.40	34.11
WHOS [192]	0.39	0.56	20.01	6.23	7.50	2.65	34.75
ELF [106]	0.34	0.52	12.02	3.85	2.42	0.83	31.17
HLBP [375]	0.32	0.43	7.07	2.18	0.76	0.54	32.68
SDC [419]	0.18	0.34	11.02	3.78	2.96	1.18	31.57
BOVW-350 [422]	1.69	1.80	33.11	13.34	14.41	6.71	32.73
BOVW-500 [422]	1.56	1.81	32.33	12.94	14.14	6.68	33.09
MobileNetV2 (M) [275]	4.39	4.34	—	—	24.01	12.34	35.62
MobileNetV2 (D) [275]	4.30	4.30	37.80	15.63	—	—	37.23
MobileNetV2 (MT) [275]	8.87	8.51	37.86	16.56	42.59	23.79	38.84
RESNET50 (M) [110]	3.84	3.90	—	—	25.67	13.62	38.01
RESNET50 (D) [110]	5.84	5.85	42.64	18.39	—	—	40.25
RESNET50 (MT) [110]	13.68	13.08	46.59	22.82	52.29	32.00	41.95
HACNN (M) [177]	5.51	5.69	—	—	23.79	13.13	36.40
HACNN (D) [177]	3.11	3.28	43.74	18.87	—	—	38.85
HACNN (MT) [177]	9.71	9.68	49.23	23.30	42.19	25.57	42.94
MLFN (M) [39]	4.91	5.19	—	—	30.39	16.96	38.67
MLFN (D) [39]	4.72	4.74	45.55	20.26	—	—	40.15
MLFN (MT) [39]	10.58	10.19	46.59	21.98	48.70	28.98	41.17
OSNET (MT) [437]	20.83	19.84	65.94	37.36	65.98	45.20	45.47
OSNET-IBN (M) [436]	10.48	10.22	—	—	48.52	26.59	40.96
OSNET-IBN (D) [436]	8.01	7.85	57.48	26.01	—	—	40.65
OSNET-IBN (MT) [436]	21.70	20.78	66.45	37.13	67.41	45.52	45.37
OSNET-AIN (M) [436]	12.14	11.67	—	—	52.42	30.35	42.05
OSNET-AIN (D) [436]	9.54	9.24	61.10	30.64	—	—	42.79
OSNET-AIN (MT) [436]	28.49	27.00	69.95	43.30	71.14	52.69	52.26
TransReID (MT) [111]	—	—	—	43.52	—	55.42	—

4.2.3 Summary and Discussion

Due to the large amount of information associated with the experimental evaluation, which includes a diverse range of datasets, methods, and tasks, a table has been created to help easily understand which dataset is used with each method. Table 4.5 contains bullets to indicate which methods were applied to each dataset. The methods were divided into 3 categories: query performance prediction (QPP), retrieval, and classification. The colors are used to distinguish between the supervision types. Notice that the methods RFE and Manifold-GCN appear twice since they were used for both classification and retrieval. This is possible because these methods produce embeddings that can be used for different tasks, which makes them flexible.

Besides the high variety of datasets used in each case, please notice that, except for CCL, all the methods evaluate their results on Market1501 and DukeMTMC, which are the most commonly used benchmarks for person Re-ID. Since Re-ID is a retrieval task, these datasets are not employed when classification is performed.

For the query performance prediction approaches, a very similar set of datasets was considered to make the comparisons feasible. The retrieval scenario considers a great variety of general-purpose datasets, with Holidays and UKBench being the ones used for comparison with the state-of-the-art. HRSF is an exception because it has the objective and is exclusively evaluated on Re-ID. For classification, the CCL, due to the nature of supervised metric learning, which requires big datasets and more examples for training, larger datasets with hundreds of images per class were considered.

Table 4.5 – Datasets used in the evaluation of each of the proposed methods, categorized by task and type of supervision. The following colors are considered: blue for unsupervised, orange for semi-supervised, and red for supervised.

Supervision Type				Unsupervised					Semi-supervised			Supervised
Task Type				QPP		Retrieval				Classification		
Category	Dataset	Size	Classes	DRNE	RQPPF	HRSF	JaccardMax	RFE	Manifold-GCN	RFE	Manifold-GCN	CCL
General-Purpose	ORL Faces [116]	400	40					•				
	Flowers [229]	1,360	17	•				•		•	•	
	MPEG-7 [161]	1,400	70	•	•			•				
	Holidays [127]	1,491	500				•	•				
	Brodatz [35]	1,776	16	•	•			•				
	Corel5k [194]	5,000	50				•	•		•	•	
	UKBench [230]	10,200	2,550				•	•				
	CUB200 [346]	11,788	200								•	
	Dogs [142]	20,580	120				•					
	CIFAR-100 [150]	60,000	100									•
	MiniImageNet [345]	60,000	100									•
	ALOI [98]	72,000	1,000					•				
	Food101 [33]	101,000	101									•
Person Re-ID	CUHK03 [176, 429]	14,097	1,467			•		•	•			
	Market1501 [422]	32,217	1,501	•	•	•	•	•	•			
	DukeMTMC [428]	36,411	1,812	•	•	•	•	•	•			
	Airport [137]	39,902	9,651			•						

5 Self-Supervised Contextual Effectiveness Estimation Measures

The possibility of estimating the effectiveness of a set of ranked lists computed by a given visual feature without any labeled data is a very challenging but relevant approach [266, 248]. For image retrieval, some measures addressed the problem in unsupervised scenarios, such as the Authority Score [243] and Reciprocal Score [248]. Despite the significant results, such measures are mainly grounded on graph-based formulations of ranking information and do not exploit deep learning models or regression approaches. In contrast, most query performance prediction methods based on machine learning are supervised [61, 233].

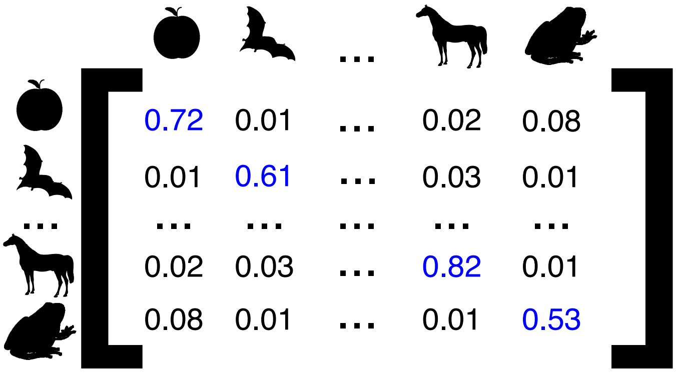
In this chapter, two self-supervised effectiveness measures are presented: the Deep Rank Noise Estimator (DRNE) [330] and the Regression for Query Performance Prediction Framework (RQPPF) [336]. They innovate by proposing completely self-supervised training based on synthetic data using different contextual representations to estimate the effectiveness of the ranked lists. Specifically, DRNE utilizes a denoising convolutional neural network (CNN) on contextual images derived from ranked lists, whereas RQPPF employs regression analysis on contextual meta-features.

The chapter is organized as follows: Section 5.1 outlines the methodology for generating synthetic data, a technique employed by both measures. Section 5.2 details the DRNE approach, while Section 5.3 introduces the RQPPF method. Additionally, Section 5.4 covers the experimental evaluation of these approaches, including visual results.

5.1 Synthetic Data Generation

In most scenarios, neural networks, and regression models rely on labeled data, which is not always easily available. To propose an entirely unsupervised training setting, we have developed an algorithm that generates synthetic ranked lists. The objective of synthetically generated ranked lists is to emulate the retrieval process, which can include relevant and non-relevant elements. The proposed approach considers a set of virtual classes to define the notion of relevance. In this way, it becomes possible to assess the effectiveness of synthetic ranked lists, which can be used for training deep learning and regression models.

Our synthetic scenarios rely on the generation of a confusion matrix of probabilities M. It is a squared $C \times C$ matrix where C is the number of virtual classes, present in the synthetic scenario. Being k_v the size of each virtual class, $C = N/k_v$, where N is the dataset size. The matrix is required to be symmetrical with all the values in the range $[0, 1]$, and all the rows and columns are also required to sum to 1, to keep the consistency with the idea of probabilities. The position (i, j) in this matrix corresponds to the probability of the elements of class i being mistaken by elements of class j. Following this reasoning, an element in the diagonal (position (i, j), where $i = j$) corresponds to the probability of an element being correctly attributed to its class. From this perspective, imposing restrictions on the values in the diagonal can increase or decrease the effectiveness of the ranked lists being generated. Figure 5.1 illustrates the similarities among classes, where the diagonal elements are highlighted in blue.



	🍎	🦇	...	🐴	🐸
🍎	0.72	0.01	...	0.02	0.08
🦇	0.01	0.61	...	0.03	0.01
...	...	...	...	...	...
🐴	0.02	0.03	...	0.82	0.01
🐸	0.08	0.01	...	0.01	0.53

Figure 5.1 – Illustration of a confusion matrix of probabilities between classes.

For a more detailed discussion of this approach, Algorithm 1 presents the method for generating a synthetic confusion matrix of probabilities. The algorithm receives the dataset size N, number of virtual classes C, and diagonal restriction values $minDiag$ and $maxDiag$ and produces a matrix M where each element represents the probability of one class being confused with another. The algorithm is divided into the following steps:

- Initialization: The algorithm begins by initializing a square matrix M of size $C \times C$ with zeros (line 2). This matrix is populated with probabilities in subsequent steps.

- Diagonal assignment: Next, the algorithm assigns random values to the diagonal elements of the matrix within the specified interval $[minDiag, maxDiag]$ (lines 5-7). This ensures that the diagonal values, representing the probability of correctly classifying each class, are within the defined range.

- Off-diagonal elements calculation: The algorithm then calculates the off-diagonal elements (lines 10-18). For each row i, it calculates the remaining probability mass after accounting for the diagonal value $1 - M[i, i]$ (line 11). This remaining probability mass is distributed uniformly among the off-diagonal elements in the row using a uniform random distribution (line 12). The function $randomUniformDistribution$ generates $N - 1$ values whose sum is $lineSum$. Consequently, the range of each off-diagonal element is from 0 to $lineSum$, ensuring that the resulting sum of each row is equal to 1.

- Matrix symmetrization: To make the confusion matrix symmetric (lines 21-27), the algorithm averages each pair of off-diagonal elements $M[i, j]$ and $M[j, i]$ and assigns this average value to both elements. This step ensures that the confusion between any two classes is bidirectional and equal.

- Normalization: Finally, the algorithm normalizes the matrix M (line 30) to ensure that the sum of the elements in each row is 1, preserving the probabilistic interpretation of the matrix.

Algorithm 1: Generate synthetic confusion matrix of probabilities

```
    Require: Dataset size $N$, number of virtual classes $C$, and diagonal restriction values
             $minDiag$ and $maxDiag$.
    Ensure: Confusion matrix of probabilities $M$.
 1: ▷ Initialize a square matrix with zeros
 2: $M \leftarrow initMatrix(C, C)$
 3:
 4: ▷ Assign random values to the diagonal within the given interval
 5: for $i \leftarrow 1$ to $N$ do
 6:     $M[i, i] \leftarrow random(minDiag, maxDiag)$
 7: end for
 8:
 9: ▷ Compute the off-diagonal elements
10: for $i \leftarrow 1$ to $N$ do
11:     $lineSum \leftarrow 1 - M[i, i]$
12:     $values \leftarrow randomUniformDistribution(N - 1,\ lineSum)$
13:     for $j \leftarrow 1$ to $N$ do
14:         if $i \neq j$ do
15:             $M[i, j] \leftarrow values[j]$
16:         end if
17:     end for
18: end for
19:
20: ▷ Make the matrix symmetric
21: for $i \leftarrow 1$ to $N$ do
22:     for $j \leftarrow i + 1$ to $N$
23:         $avgValue \leftarrow (M[i, j] + M[j, i])/2$
24:         $M[i, j] \leftarrow avgValue$
25:         $M[j, i] \leftarrow avgValue$
26:     end for
27: end for
28:
29: ▷ Normalize the matrix
30: $M \leftarrow normalize(M)$
31:
32: return M
```

Using the M matrix, a set of synthetic ranked lists can be generated utilizing this matrix to determine the probability of confusion between classes. However, since incorrect elements tend to be more random than the correct ones, we also generate a symmetrical confusion matrix M_c to attribute probabilities for randomly selecting the elements that belong to the same class. For each virtual class in the synthetic dataset, a matrix M_c is computed considering the Algorithm 1. The only distinction is that, instead of being a $C \times C$ matrix, it is a $k_v \times k_v$ matrix.

In this scenario, Algorithm 2 presents a method for computing a synthetic ranked list τ_q for an element of index q. The algorithm receives as input the index q, size L of the ranked list, number of virtual classes C, size k_v of classes, a confusion matrix of probabilities between classes M, and a confusion matrix of probabilities between the elements in the same class M_c. The output is a synthetic ranked list τ_q. The algorithm begins by initializing a synthetic ranked list of size L full of zeros (line 2). The list $addedElementsForClass$ is employed to register the number of elements added in the ranked list for each class C during the iterations, which is initialized with zeros in line 5. The role of $addedElementsForClass$ is to prevent adding elements for classes with no remaining elements, i.e., in case all of them were already added in τ_q. Line 8 gets the class of the element q, while Line 9 makes a copy of the corresponding row of matrix M that contains the probability between its class and other classes. The next part iterates for adding elements to τ_q (lines 12-35), involving the following steps:

- Random class selection: A random class is selected based on the probabilities obtained from the matrix M (line 14). A random value is generated in the range $[0, 1]$, and the function *getRandomClass* compares this value against the cumulative sum of the probabilities in the corresponding row $M[classQ]$. The class is selected where the random value lies within the cumulative probability range. This ensures that classes with higher probabilities have a higher chance of being selected, reflecting the distribution specified by the matrix M.

- Element selection: Since incorrect elements tend to be more random than correct ones, the process of selecting elements of the same and different classes is performed separately (lines 17-23).
    - Different class: The function $getRandomElementDiffClass$ (line 19) returns the index of a random element of the *randomClass* ensuring that this element is not already present in τ_q. Each element within the class has an equal probability of being chosen.
    - Same class: For this step, the matrix M_c is used, which is similar to matrix M but instead of containing the probabilities of selection between classes, it contains the probability of selection between elements of the same class, which in

this case is the class of element q. The function *getRandomElementSameClass* (line 22) returns an element belonging to *classQ*, ensuring that this element is not already present in τ_q.

- Updating the ranked list: The selected element is added to the synthetic ranked list (line 26).
- Tracking added elements: The algorithm updates the count of elements added for the chosen class (line 28). If the number of elements added for a class reaches the size of the virtual class k_v (line 30), this means that all of the elements of that class have already been added. The algorithm then updates the distribution of probabilities between classes (line 31) to prevent selecting that class again and resets the counter (line 33) for that class to avoid entering this if statement in the next iteration.

A set of synthetic ranked lists $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_N\}$, can be generated by executing this algorithm from $q = 1$ to $q = N$. This method of computing synthetic ranked lists is employed to generate data to train the proposed approaches presented in this chapter.

Algorithm 2: Generate synthetic ranked list

```
Require: Image index $q$,
         Confusion matrix of probabilities $M$,
         Confusion matrix of probabilities for elements in the same class $M_c$,
         Size of ranked list $L$,
         Number of virtual classes $C$,
         Size of virtual classes $k_v$.
Ensure: A synthetic ranked list $\tau_q$.
▷ Initialize synthetic ranked list
$\tau_q \leftarrow emptyList(L)$

▷ Initialize list to record the number of elements added for each class
$addedElementsForClass \leftarrow emptyList(C)$

▷ Get class of element $q$ and the corresponding row of matrix $M$
$classQ \leftarrow getClass(q)$
$classesProbDist \leftarrow copyRow(M[classQ])$

▷ Iterate to fill the synthetic ranked list up to position $L$
for $i \leftarrow 1$ to $L$ do
    ▷ Select a random class based on the probabilities obtained from matrix $M$
    $randomClass \leftarrow getRandomClass(classesProbDist)$

    ▷ Randomly select an element from the randomClass
    if $classQ \neq randomClass$ do
        ▷ Randomly select the index of an element from a different class
        $element \leftarrow getRandomElementDiffClass(randomClass, \tau_q)$
    else
        ▷ Randomly select the index of an element from the same class
        $element \leftarrow getRandomElementSameClass(M_c, \tau_q)$
    end if

    ▷ Add element to the synthetic ranked list
    $\tau_q[i] \leftarrow element$
    ▷ Keep a record of the number of elements added for each class
    $incrementByOne(addedElementsForClass[randomClass])$
    ▷ Update matrix to prevent selecting class with no remaining elements
    if $addedElementsForClass[randomClass] = k_v$ do
        $updateProbDist(classesProbDist, randomClass)$
        ▷ Prevent entering this if statement again for the same class in the next iteration
        $addedElementsForClass[randomClass] \leftarrow 0$
    end if
end for

return $\tau_q$
```

5.2 Deep Rank Noise Estimator (DRNE)

In this section, we propose a new method to estimate the effectiveness of ranked lists in a self-supervised fashion, the Deep Rank Noise Estimator (DRNE). We innovate by proposing a new model architecture based on a well-known denoiser, which has results comparable to the state-of-the-art, the DnCNN [409] (Denoiser CNN). To keep the entire workflow unsupervised, we trained the model with synthetic data. In order to create such data, we emulate the behavior of real visual features with different degrees of effectiveness. Based on the generated data, the ranked lists are converted to images according to a strategy inspired by [246] and used to train the network, which interprets the incorrectness of a ranked list as noise. When the same representation is generated for real visual features, the network is able to estimate the noise and therefore the rank effectiveness.

To the best of our knowledge, this work is the first method which deals with the challenging task of unsupervised effectiveness estimation by using a denoising deep learning model. In this way, many contributions and innovations are proposed as part of the DRNE and can be highlighted, among them: *(i)* ranked lists are represented as contextual images; *(ii)* the noise of the contextual images is interpreted as the incorrectness of the ranked lists; *(iii)* it generates synthetic data in order to train denoising networks in a totally unsupervised manner.

Our proposed strategy aims at computing effectiveness estimation measures for ranked lists without requiring any labeled data, in a self-supervised fashion. We name our method as Deep Rank Noise Estimator (DRNE) [330]. The workflow is presented in Figure 5.2. This approach can be summarized into three main steps:

1. **Computing Synthetic Data**: They are used to simulate real scenarios for training the CNN, but without using any real label or groundtruth.

2. **Ranked Lists as Images**: A strategy to represent ranked lists as images, since they are numerical data, they need to be converted to images to be provided as input to the CNN. Our approach for this step is based on [246].

3. **Effectiveness Estimation CNN**: The DnCNN [409] architecture was modified to be used as an effectiveness estimator for ranked lists, based on their "noise" level present in the images.

In addition to Step 1, which involves generating synthetic data and is discussed in Section 5.1, the subsequent steps are outlined in the following subsections.

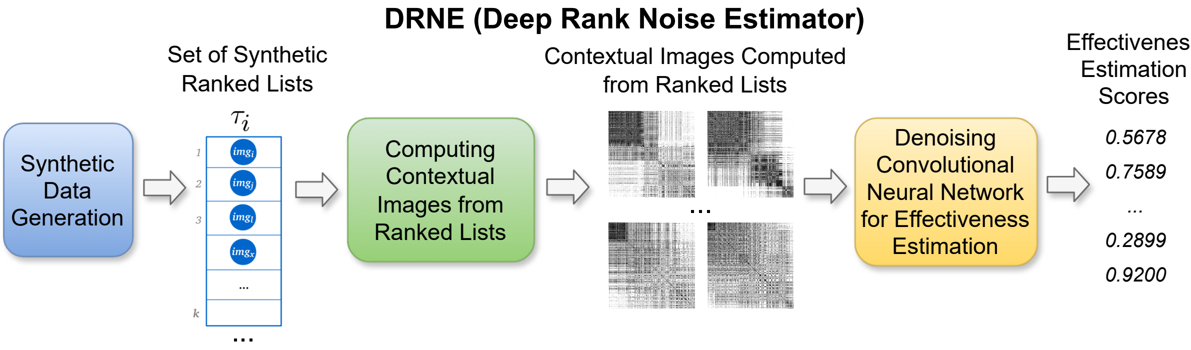

DRNE (Deep Rank Noise Estimator)
Synthetic Data Generation
Set of Synthetic Ranked Lists
τ_i
1
2
3
k
img_i
img_j
img_l
img_x
...
Computing Contextual Images from Ranked Lists
Contextual Images Computed from Ranked Lists
Denoising Convolutional Neural Network for Effectiveness Estimation
Effectiveness Estimation Scores
0.5678
0.7589
...
0.2899
0.9200

Figure 5.2 – Diagram illustrating the main stages of the DRNE.

5.2.1 Computing Contextual Images from Ranked Lists

The ranked lists consist of numerical data, where each value corresponds to the index of the image being ranked. In this work, we propose a model for transforming a ranked list into image data, based on what was proposed in [246].

Given a pair of ranked lists τ_i and τ_j, a grayscale image can be modeled such that the pixel (p_x, p_y) is defined as the mean of the positions that the elements occur in both lists:

$$pixel(p_x, p_y) = (\tau_x(y) + \tau_y(x))/2, \tag{5.1}$$

where $p_x = \tau_i(x)$ and $p_y = \tau_j(y)$.

For DRNE, we always consider images of the same ranked list to the same ranked list (such that $\tau_i = \tau_j$), which produces symmetrical images. The positions with higher similarity are represented by darker pixels and the ones with lower similarity by brighter ones. Regarding the image size, it is directly related to the value of L, which is the size of the ranked lists. We considered $L = 200$ to obtain 200x200 images in all the cases. The use of the same size of image for all the datasets shows the scalability potential of our method.

To illustrate this process, Figure 5.3 depicts an image generated from a hypothetical set of ranked lists. Specifically, this example focuses on the image calculated for the element at index 0, denoted as obj_0. It is important to note that the image is symmetric along the diagonal, which is black. The pixels outlined in red are those computed in this example. Due to the symmetry, two pixels are colored red; computing one automatically computes the other. The pixel at position (0, 1) in the generated image corresponds to elements 0 and 4 (obj_0 and obj_4), which occupy positions 0 and 1 in the ranked list of obj_0 (referred to as τ_{obj_0}). The pixel value is calculated as the average of the positions that the elements occupy in each other's ranked lists, which in this case is 25, as depicted in the figure. The computation of the pixel value is defined by Equation 5.1.

Figure 5.4 presents examples of images generated for synthetic ranked lists with

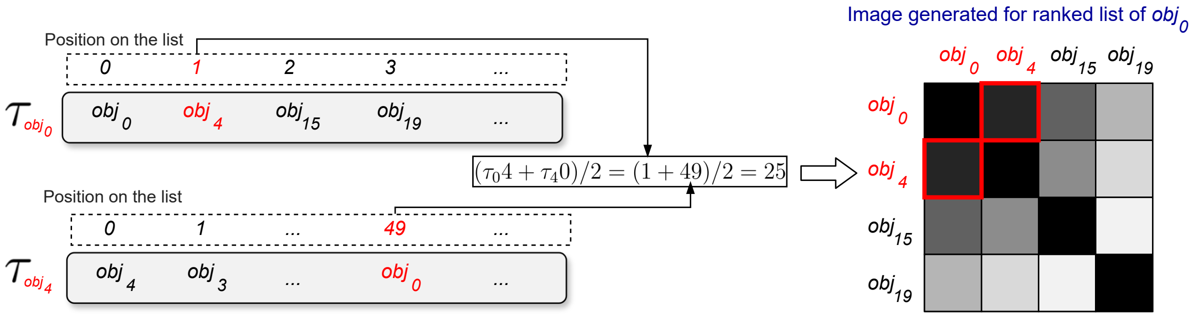


Figure 5.3 – Illustration that exemplifies the calculation of a contextual image constructed from a hypothetical ranked list.

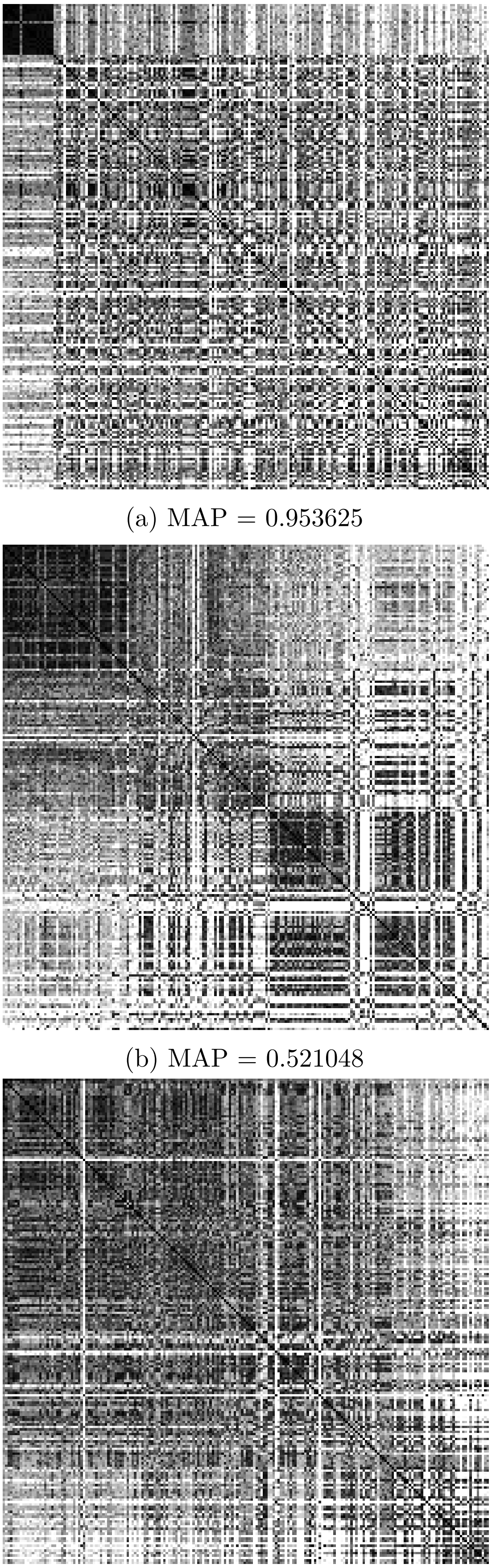
different effectiveness levels. The synthetic data was generated considering $N = 1400$, $L = 200$, $C = 70$, and $k_v = 20$. It can be seen that as the MAP (Mean Average Precision) increases, more blacker pixels tend to appear in the upper left corner of the image, for example. However, there are still many other aspects that can be analyzed in this type of image, since there are multiple possible images for ranked lists with the same MAP.

5.2.2 Denoising Convolutional Neural Network for Effectiveness Estimation

As a part of the DRNE approach, this work proposes a Convolutional Neural Network for estimating the effectiveness of ranked lists based on their image representations. The idea is that each ranked list image contains a certain level of noise, which is related to its effectiveness. Following this reasoning, the more effective a ranked list is, the less noise is associated with it, and vice versa.

Our method involves applying the model to extract noise and assigning a score, which we expect to be related to the effectiveness of the ranked list. Figure 5.5 presents the model proposed and considered for all the experiments in this work. We modified the DnCNN [409] model to consider 10 blocks of convolution, batch normalization, and activation layer. The learned noise is flattened and submitted to a sequence of dense and dropout layers which should learn a single float score that represents the effectiveness of the ranked list provided as input. The MAP of the synthetic data is considered as the groundtruth during training.

In all the experiments, the Nesterov-accelerated Adaptive Moment Estimation (NAdam) optimizer was used with a learning rate of 10^{-4} and Mean Squared Error (MSE) loss. The network was trained considering batches of size 2, where both images correspond to the same image but with different augmentations. The method is set to have a 50% probability of thresholding the pixels of the image. If the image is chosen for thresholding, a random value between 100 and 255 is selected. All pixels with values above this threshold are then set to 255. This is done to improve the network generalization during the learning process.



(a) MAP = 0.953625

(b) MAP = 0.521048

(c) MAP = 0.198827

Figure 5.4 – Examples of images generated for synthetic ranked lists with different degrees of effectiveness.

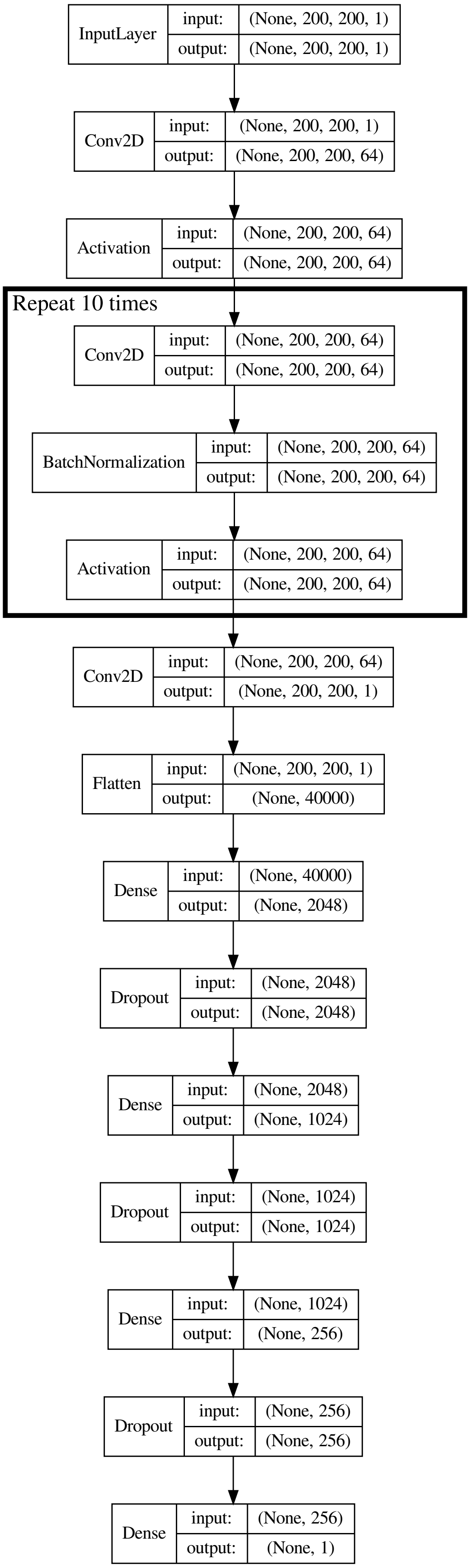


Figure 5.5 – Proposed CNN model for effectiveness prediction.

5.3 Regression for Query Performance Prediction Framework (RQPPF)

This section presents the proposed Regression for the Query Performance Prediction Framework - RQPPF. The proposed approach exploits synthetically generated data to support a self-supervised learning strategy. Contextual rank-based measures are extracted from the data to train the regression models. Among the primary contributions of RQPPF, we can highlight: *(i)* a framework for applying regression models to the task of query performance prediction, the RQPPF; *(ii)* a strategy for modeling ranked lists as feature vectors, including reciprocal neighborhood analysis; *(iii)* an approach consistently trained on synthetic data, eliminating the need for real labels and enabling cost-effective self-supervised training; *(iv)* significant results were achieved on 4 datasets, including person re-identification (Re-ID) scenarios.

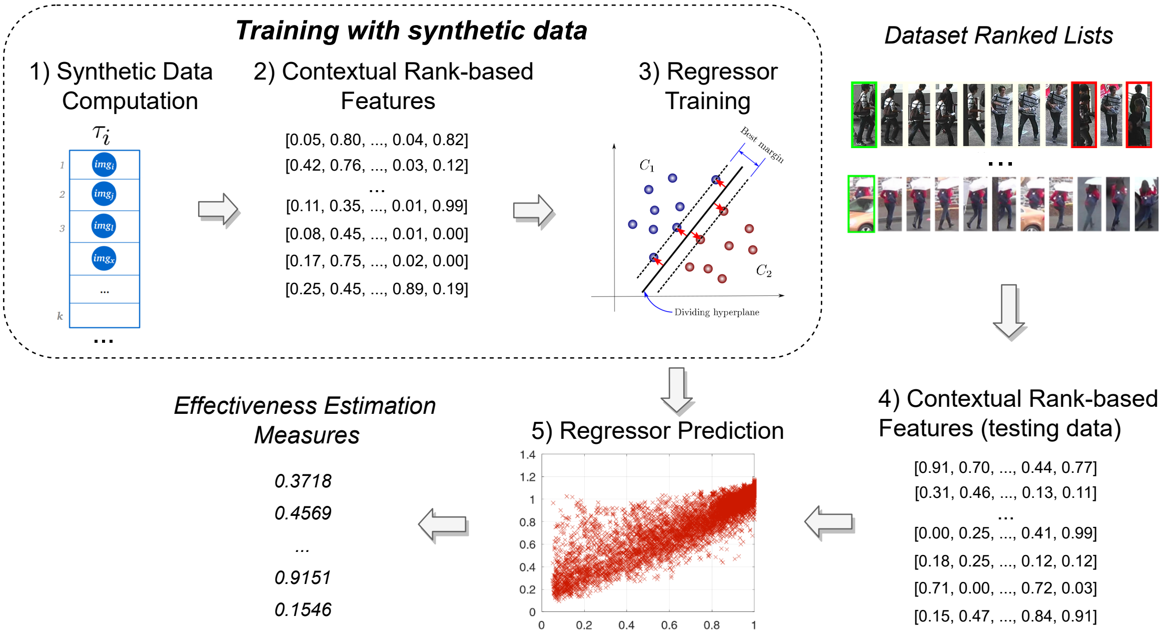


Figure 5.6 – Diagram of the proposed approach (RQPPF) for self-supervised query performance prediction.

Figure 5.6 illustrates the main steps of the proposed method. The main ideas of each step are outlined as follows:

1. **Synthetic Data Computation:** How to train a regression model if no label data is available? This challenge was addressed in our approach by generating synthetic ranked lists. Each ranked list is assigned to a virtual class and a synthetic confusion matrix is used to define the probability of including non-relevant results (from other virtual classes) in the retrieval results.

2. **Contextual Rank-Based Features (for Training Data):** Contextual rank-based features are extracted from the synthetic ranked lists. The features include information from the reciprocal ranking references and unsupervised effectiveness estimation measures.

3. **Regression Model Training:** The contextual rank-based features and effectiveness measures based on virtual classes are available for each synthetic ranked list. Based on this data, the regression model is trained;

4. **Contextual Rank-Based Features (for Testing Data):** The same contextual rank-based measures extracted from synthetic training data are also extracted from real-world datasets for testing.

5. **Regression Model Prediction:** The self-supervised regression model is employed on contextual rank-based measures from testing data for query performance prediction.

Section 5.3.1 presents functions used as part of the formulation of the proposed approach, including the Authority Score [243] and the Reciprocal Density [248] effectiveness estimation measures. While Section 5.3.2 describes how the meta-features are computed along with equations and formal definitions, Section 5.3.3 discusses how the framework employs the regression models for self-supervised training.

5.3.1 Background Formulation

A function is defined to explain some steps of our approach as $f_p : \mathcal{C} \times \mathbb{N} \rightarrow \mathcal{C}$, such that given an image o_q and the position p, it returns the image in the position p of the ranked list τ_q:

$$f_p(o_q, p) = \{o_i \mid o_i \in \mathcal{C} \wedge \tau_q(o_i) = p\}. \tag{5.2}$$

We also define the function $f_{in}(i, q) \rightarrow \{0, 1\}$, which indicates if an image belongs to a neighborhood set, returning 1 if $o_i \in \mathcal{N}(q, k)$.

Our approach uses rank-based measures, represented by $\gamma(\cdot, \cdot)$, which can be any unsupervised effectiveness estimation measure. In this work, we considered both Authority [243] and Reciprocal [248] as measures to compute the meta-features.

The Authority Measure [243] is based on a graph of image relationships from ranked lists for effectiveness estimation. Each image in top-k positions of the ranked list τ_q defines a node. For each image o_i in the top-k of τ_q, the ranked list τ_i is also analyzed. If there are images in common in ranked lists τ_q and τ_i, an edge is created. The Authority Score is computed based on the graph density:

$$\gamma_A(\tau_q, k) = \frac{\sum_{i \in \mathcal{N}(q,k)} \sum_{j \in \mathcal{N}(i,k)} f_{in}(j, q)}{k^2}, \tag{5.3}$$

where γ_A ranges from 0 to 1, with a higher score indicating a fully connected graph within the top-k positions.

Similar to the Authority [243], Reciprocal Density [248] is defined based on graph density. However, it also incorporates neighbor weights:

$$\gamma_R(\tau_q, k) = \frac{1}{k^4} \sum_{i \in \mathcal{N}(q,k)} \sum_{j \in \mathcal{N}(i,k)} f_{in}(j, q)\ w_r(q, i)\ w_r(i, j). \tag{5.4}$$

Weights are determined by the function $w_r(q, i) = k + 1 - \tau_q(i)$, with higher weights indicating a frequent occurrence of reciprocal neighbors in top-ranked positions.

5.3.2 Contextual Rank-based Features

Our approach uses the method presented in Section 5.1 for generating synthetic data. One of the main contributions of this work is the strategy to compute contextual rank-based features, also known as meta-features, which are computed for both training and testing data. The process of computing meta-features is composed of the following steps.

(i) **Reciprocal Neighborhood**: A binary vector $\mathbf{b}$ is modeled to encode the reciprocal (mutual) neighborhood, where 1 indicates that the corresponding element is a mutual neighbor and 0 otherwise. The idea is that elements that are reciprocal neighbors have a higher relevance. Let $\mathbf{b_i}$ represent a binary vector corresponding to image o_i, and let d denote the depth to which the ranked list is analyzed. The vector $\mathbf{b_i}$ is defined based on the reciprocal references among images for image o_i, as:

$$\mathbf{b_i} = [b_{i_1}, b_{i_2}, \dots, b_{i_d}], \tag{5.5}$$

where b_{i_j} is computed for o_j from $j = 1$ to $j = d$, such that $b_{i_j} = 1$ if $o_j \in \mathcal{N}(i, k)\ \wedge\ o_i \in \mathcal{N}(j, k)$, and $b_{q_i} = 0$, otherwise. The o_j is the j-th element in the ranked list of element i, such that $o_j = f_p(o_i, j)$.

(ii) **Effectiveness Estimation Measures**: Query performance prediction approaches can be used to encode effectiveness estimation into the meta-feature, such that higher values indicate higher effectiveness. This approach considers either Authority [243] or Reciprocal [248] for this step. Let $\mathbf{q_i}$ be a vector for the image o_i that contains the query performance prediction values of one of these approaches. The vector $\mathbf{q_i}$ is defined considering a depth d:

$$\mathbf{q_i} = [q_{i_1}, q_{i_2}, \dots, q_{i_d}], \tag{5.6}$$

where q_{i_j} is the value provided by an effectiveness estimation function; such that $q_{i_j} = 1 + \gamma(\tau_j)$, where $o_j = f_p(o_i, j)$. Both Authority (γ_A) [243] and Reciprocal (γ_R) [248] scores can be used interchangeably as the γ function.

***(iii)* Reciprocal Rank Position**: The reciprocal position of two images in their ranked lists, i.e., the position of image i in the ranked list of image j and vice versa, can provide valuable insights into their similarity. If both images rank each other highly, it indicates a strong mutual similarity. Conversely, if the ranks are low, it suggests that the images are less similar to each other. This reciprocal ranking approach can be particularly useful in refining image similarity. Because of this, a vector $\mathbf{p}$ is computed to encode the mean position between two images in their ranked lists. Similar to the other vectors, $\mathbf{p_i}$ is defined as follows:

$$\mathbf{p_i} = [p_{i_1}, p_{i_2}, \ldots, p_{i_d}]. \tag{5.7}$$

Let $o_j = f_p(o_i, j)$ be the j-th element in the ranked list of element i, the value of p_{i_j} is based on the mean of $\tau_j(i)$ and $\tau_i(j)$, defined as:

$$p_{i_j} = \frac{1}{((\tau_i(j) + \tau_j(i))/2 + 1)}. \tag{5.8}$$

The inverse ensures that the top positions in the ranked lists correspond to higher values, effectively functioning as a similarity measure.

***(iv)* Contextual Rank-based Feature (Meta-Feature)**: The vectors $\mathbf{b_i}$, $\mathbf{q_i}$, and $\mathbf{p_i}$, are used to compute the meta-feature $\mathbf{f}$ that is used to train the regression model.

$$\mathbf{f} = \mathbf{b} \times (\mathbf{b} + \mathbf{q} \times \mathbf{p}). \tag{5.9}$$

In summary, $\mathbf{b}$ is a binary vector considering reciprocal neighbors, $\mathbf{q}$ represents effectiveness measures by γ, and $\mathbf{p}$ encodes position data. These aspects are all combined in the vector $\mathbf{f}$. Using the contextual rank-based features $\mathbf{f}$, we can train a regression model on synthetic data and subsequently evaluate its performance using a real dataset for testing.

5.3.3 Regression Models

Our framework is flexible and can be trained with different regression models. Let a regression model be defined as a pair of functions (ϕ_{tr}, ϕ_{ts}), such that ϕ_{tr} is responsible for training and ϕ_{ts} for testing. Both train and test features are modeled as described in Equation 5.9. The training procedure is performed considering only synthetic data, which is represented by a set of synthetic features $\mathbf{f}_s$. The set $\mathbf{f}_s$ is taken as input by ϕ_{tr}, such that $\phi_{tr}(\mathbf{f}_s)$ returns the effectiveness estimation scores of training data that are evaluated along the epochs by the loss function. In this work, the models were trained considering the default parameters[5]. We highlight that, for training, only synthetic data was considered, which makes the approach self-supervised.

[5] Python 3.8 sklearn implementation was considered for all regression models.

The idea is to apply transfer learning to real scenarios. The real dataset is used as a test set to obtain the effectiveness estimation results. We denote the set of test features by $\mathbf{f}_t$, which is provided as input for the prediction regression model (ϕ_{ts}), such that $\phi_{ts}(\mathbf{f}_t)$ returns the effectiveness estimation scores of the real dataset used for testing. The higher the score, the higher the predicted effectiveness.

5.4 Experimental Evaluation

This section presents the experimental evaluation for both DRNE and RQPPF, conducted under the same protocol. Additionally, both methods are compared with each other. A discussion is provided at the end of this section, which includes a joint analysis and combinations of the approaches.

5.4.1 Experimental Protocol

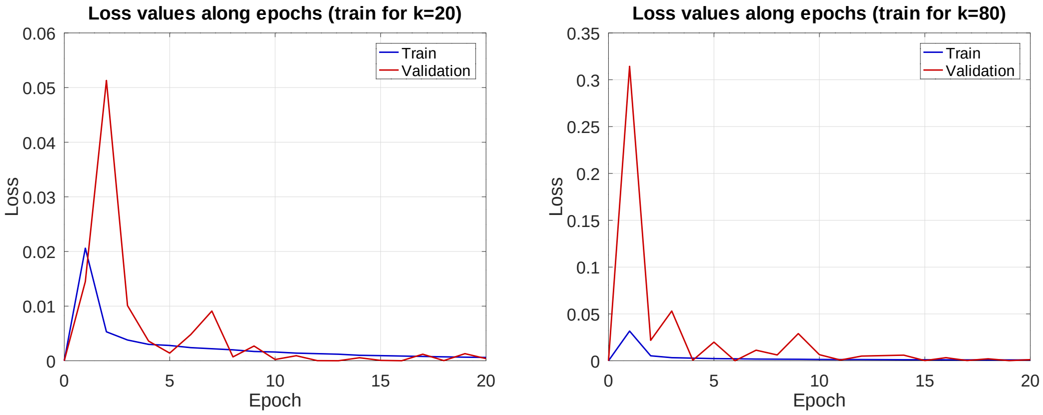
The experiments considered 5 different datasets with sizes ranging from 1,336 to 36,411 images: Flowers [229], MPEG-7 [161], Brodatz [35], Market1501 [422], and DukeMTMC [428]. More detailed information about the datasets and effectiveness measures is provided in Chapter 4. For all the datasets, the Pearson correlation of the QPP approach and the Mean Average Precision (MAP) was considered for evaluating the effectiveness. In all the cases, all images were considered as query images, except for Re-ID datasets, where only query images specified by the dataset protocol were considered, as done by most of the authors in the literature. The descriptors vary in each case, according to the properties of each dataset. In total, more than 30 different descriptors were considered for these experiments.

Two different trainings were done, both of them considered artificially generated data for keeping the strategy and analysis unsupervised. One of the main parameters is the size of the virtual classes (k_v), which impacts the images generated for training. While the first training considered data with $k_v = 20$, the second used $k_v = 80$. The models trained with $k_v = 80$ were evaluated on the Flowers [229] dataset, while the models trained with $k_v = 20$ were evaluated on all the other datasets. The artificial dataset contains 1,400 and 1,360 images for training with $k_v = 20$ and $k_v = 80$, respectively. To compute meta-features, the RQPPF considered the value of the neighborhood and depth d equal to the size of virtual classes, such that $k = k_v$. Of the total synthetic images, 200 were randomly selected for validation, and the remaining images were used for training. In both cases, 7 synthetic ranked list sets were generated with different levels of effectiveness. This adjustment is done by restricting the intervals in the diagonal of the confusion matrix: the first descriptor uses [0, 0.25], second uses [0, 0.5], third uses [0, 0.75], fourth uses [0, 1], fifth uses [0.25, 1], sixth uses [0.5, 1], and seventh uses [0.75, 1].

5.4.2 DRNE Parameter Analysis

Figure 5.7 shows the loss values for training and validation data in both artificial training sets (i.e., with $k_v = 20$ and $k_v = 80$) along 20 epochs. As can be seen, the losses decrease as the epochs increase. After 15 epochs, we can see that the model starts to decrease the training loss much slower than before, but the validation still varies. For this reason, we trained the model for 15 epochs in all experiments to avoid overfitting on the artificial data.

Our method is compared to both Authority and Reciprocal. To keep the comparison fair, we used $k = 20$ for DRNE and the baselines in all cases. The only exception is the Flowers dataset, where $k = 80$ was used since it has larger classes than the others.



(a) Training for $k_v = 20$.

(b) Training for $k_v = 80$.

Figure 5.7 – Losses along training epochs for train and validation sets.

5.4.3 DRNE Results

Table 5.1 presents the Pearson correlation between MAP and the effectiveness estimation measures (Authority Score, Reciprocal Density, and DRNE) for around 30 different descriptors on the Flowers dataset. To keep the comparison fair in this case, $k = 80$ was used for our approach and the baselines. The best results are highlighted in bold for each line. For the negative correlations (FOH and SCH), none of the methods were highlighted in bold. The last line of the table contains the correlation when considering all the ranked lists of all descriptors together. The original MAP of each descriptor is also presented with the objective of facilitating the analysis of the results. Notice that the best results (higher correlations) tend to be more frequent on descriptors of high effectiveness (which is the case of the CNNs). Consequently, scenarios with descriptors of low effectiveness tend to be more challenging (e.g. FOH, SCH, GIST). Besides that, the proposed approach (DRNE) achieved the best results in most of the cases, even in difficult scenarios (e.g. ACC, EHD, SPLBP). These cases of negative correlation occur due to the low effectiveness of such descriptors and still require more investigation.

Table 5.1 – Pearson correlation between MAP and effectiveness estimation measures on Flowers dataset.

Descriptors	**Original MAP**	**Auth.**	**Recipro.**	**DRNE (ours)**
CNN-FBResNet [110]	52.56%	0.73744	0.67153	**0.79920**
CNN-ResNeXt [372]	51.91%	0.76568	0.66525	**0.79265**
CNN-ResNet [110]	51.83%	0.72981	0.63672	**0.79903**
CNN-DPNet [51]	50.93%	0.77143	0.72479	**0.79896**
CNN-Xception [52]	47.31%	0.74365	0.64060	**0.76958**
CNN-BnInception [122]	46.58%	0.57857	0.48638	**0.72061**
CNN-AlexNet [151]	46.04%	0.46586	0.35353	**0.63521**
CNN-SENet [117]	43.16%	0.58722	0.57195	**0.63076**
CNN-InceptionV4 [302]	42.35%	**0.67885**	0.58592	0.61974
CNN-InceptRN [302]	42.20%	**0.62725**	0.53364	0.55041
CNN-BnVGGNet [198]	41.87%	0.48524	0.36175	**0.63133**
CNN-NASNetLg [445]	40.74%	**0.63091**	0.55103	0.54974
CNN-VGGNet [198]	39.05%	0.50498	0.32844	**0.63850**
SIFT [205]	28.47%	0.34815	0.31624	**0.48026**
BIC [295]	25.56%	0.21481	0.16794	**0.36447**
SPJCD [401, 209]	22.56%	0.27962	0.24767	**0.33553**
SPCEDD [40, 209]	21.94%	0.31110	0.26055	**0.34731**
COMO [342]	21.83%	0.10506	0.08213	**0.25892**
SPFCTH [41, 209]	21.73%	0.19618	0.18878	**0.26632**
JCD [401]	20.89%	0.15319	0.11306	**0.24018**
FCTH [41]	20.56%	0.18428	0.13488	**0.23862**
CEDD [40]	20.48%	0.13077	0.10192	**0.20104**
SPACC [119, 209]	19.20%	0.07436	0.03312	**0.20229**
ACC [119]	18.99%	0.03264	0.02153	**0.28373**
CLD [53]	18.54%	0.32734	0.25345	**0.34693**
PHOG [59, 209]	14.74%	0.33586	0.33548	**0.37418**
SCH [53]	13.43%	-0.21997	-0.20886	-0.13598
EHD [215]	12.46%	0.03510	0.06457	**0.20214**
FOH [337, 209]	11.42%	-0.06418	-0.06645	-0.03603
SPLBP [231, 209]	10.92%	0.06942	0.07869	**0.14425**
LBP [231]	10.34%	0.01482	0.02083	**0.07323**
SCD [53]	10.25%	**0.25619**	0.10035	0.05702
GIST [232]	9.82%	-0.01581	0.02297	**0.02691**
All Descriptors	—	0.39789	0.31277	**0.42907**

An experiment was conducted to evaluate the complementary among the results provided by each effectiveness estimation measure. Table 5.2 shows the Pearson correlation between each pair of measures for the MPEG-7 dataset. Notice that Authority and Reciprocal are highly correlated, while DRNE is the least correlated with the other two, which indicates that our approach has great potential to be combined with the others.

Table 5.2 – Pearson correlation between estimation measures for all descriptors of MPEG-7 dataset.

	Authority	**Reciprocal**	**DRNE (ours)**
Authority	1.0000	0.96928	0.86480
Reciprocal	0.96928	1.0000	0.87641
DRNE (ours)	0.86480	0.87641	1.0000

All the remaining datasets, where $k = 20$ was used, are presented in Table 5.3. Besides the individual results for each measure, combinations of measures are also presented. The combinations were done by summing the measures, also the abbreviations A, R, and

Table 5.3 – Pearson correlation between MAP and effectiveness estimation measures on datasets considering train with $k = 20$.

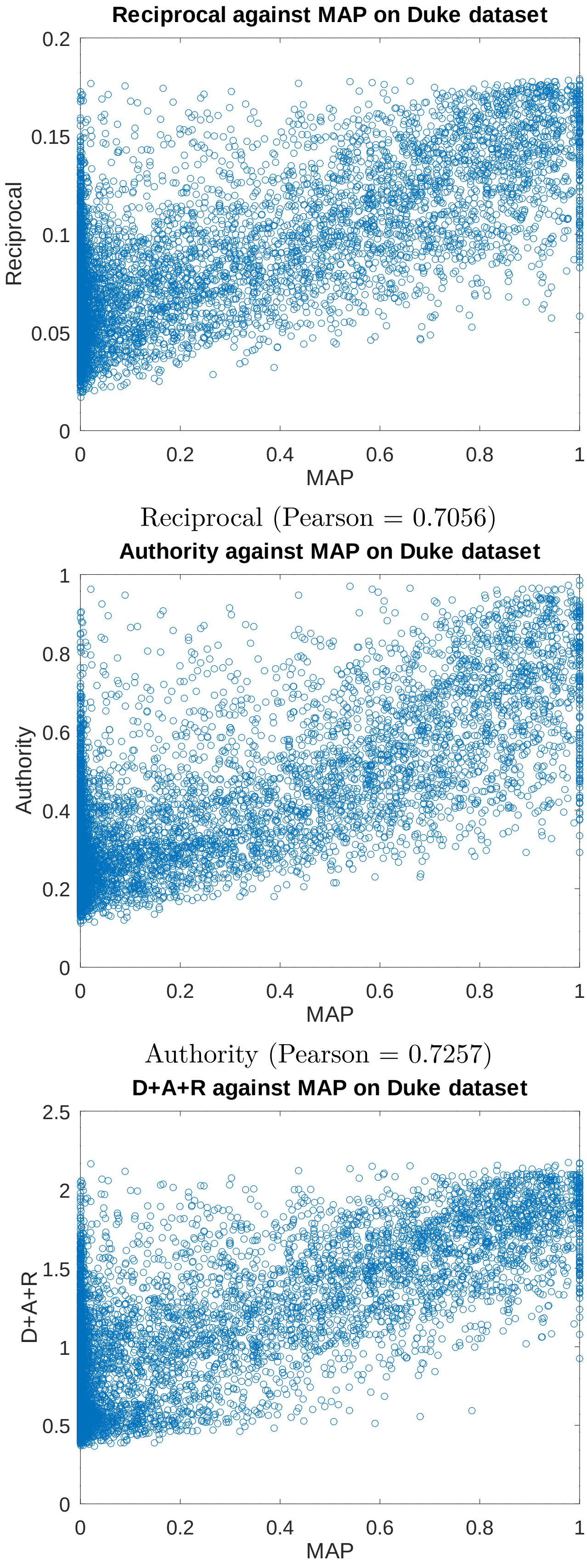
Dataset	Descriptors	Original MAP	Auth.	Recipro.	DRNE	A+R	D+A	D+R	D+A+R
MPEG-7	**AIR** [103]	89.39%	0.76392	0.75069	0.87705	0.76419	0.86921	**0.88494**	0.86386
	ASC [191]	85.28%	0.76594	0.81430	0.74678	0.77823	0.79525	0.78578	**0.80045**
	IDSC [190]	81.70%	0.77826	**0.80911**	0.74767	0.78716	0.79826	0.78162	0.80235
	CFD [244]	80.71%	0.79817	0.83621	0.82587	0.80758	0.84616	0.84731	**0.84769**
	BAS [13]	71.52%	0.79029	**0.84011**	0.79698	0.80281	0.81826	0.82081	0.82334
	SS [317]	37.67%	0.78474	0.81322	0.84026	0.79460	0.83954	**0.84605**	0.84076
	All Descriptors	—	0.85355	0.88229	0.84607	0.86131	0.88019	0.86754	**0.88336**
Brodatz	**LAS** [308]	75.15%	0.64725	0.63484	0.69576	0.65333	0.70116	**0.70457**	0.70007
	CCOM [148]	57.57%	0.63535	0.60433	0.65631	0.63799	**0.67730**	0.66598	0.67608
	LBP [231]	48.40%	0.49609	0.42214	0.49984	0.49278	**0.52400**	0.50540	0.5199
	All Descriptors	—	0.57502	0.54266	0.59152	0.57759	**0.61023**	0.60107	0.60917
Market	**CNN-OSNET-AIN** [436]	43.30%	0.65170	0.60202	0.63451	0.64876	**0.66854**	0.64148	0.66665
	CNN-HACNN [177]	23.30%	0.52763	0.48562	0.52421	0.52611	**0.54461**	0.52853	0.54371
	CNN-ResNet [110]	22.82%	0.60783	0.55807	0.60246	0.60517	**0.62471**	0.60710	0.62385
	CNN-MLFN [39]	21.98%	0.57916	0.53273	0.55649	0.57662	**0.58287**	0.56158	0.58243
	BOVW [422]	13.34%	0.39235	0.31576	0.38518	0.3832	**0.40171**	0.38429	0.3983
	WHOS [192]	6.23%	0.13383	0.14891	0.22140	0.13952	0.20209	**0.21919**	0.2005
	All Descriptors	—	0.61279	0.53534	0.57867	0.60599	**0.61197**	0.58337	0.61038
Duke	**CNN-OSNET-AIN** [436]	52.69%	0.64525	0.64349	0.64988	0.64861	0.67623	0.66199	**0.67631**
	CNN-ResNet [110]	32.00%	0.69101	0.67709	0.67628	0.69335	0.70446	0.68566	**0.70552**
	WHOS [192]	2.65%	0.00572	0.03644	**0.11433**	0.01163	0.07868	0.10800	0.07615
	All Descriptors	—	0.72574	0.70560	0.71234	0.72660	0.74163	0.72138	**0.74189**

D were used for Authority, Reciprocal, and DRNE, respectively. Notice that in most of the cases, the best results correspond to our approach or a combination that involves our approach. While most of the datasets consider classes of the same size, this does not occur for the Re-ID datasets (Market and Duke). Even with this challenge, the results are very promising. The combination of the three measures achieved up to 0.74 Pearson correlation in the Duke dataset, which is very significant considering that no labels were used.

Figure 5.8 presents a graph where each dot corresponds to a ranked list of the DukeMTMC dataset. The dots are plotted according to the value presented by the effectiveness estimation (which uses no labels) and the MAP (which uses labels). As can be seen, the results provided by the combination of the three measures present a more linear shape, and consequently a higher Pearson correlation as well.

Two visual query examples are presented in Figure 5.9 with the DRNE score obtained for each of them. The query image is presented in green borders and the incorrect results are in red borders. Notice that DRNE attributed a lower score for the ranked list which presented wrong images and a higher score (very close to 1) for the one without errors.

Regarding execution time, the prediction time is 9.2909 ± 9.38663 milliseconds considering the mean and standard deviation for 44,880 different ranked lists. A training of 9,600 images takes about 25 minutes to run for each epoch on NVIDIA RTX 2080 GPU. For a training of 20 epochs, it is required around 8 hours in total.

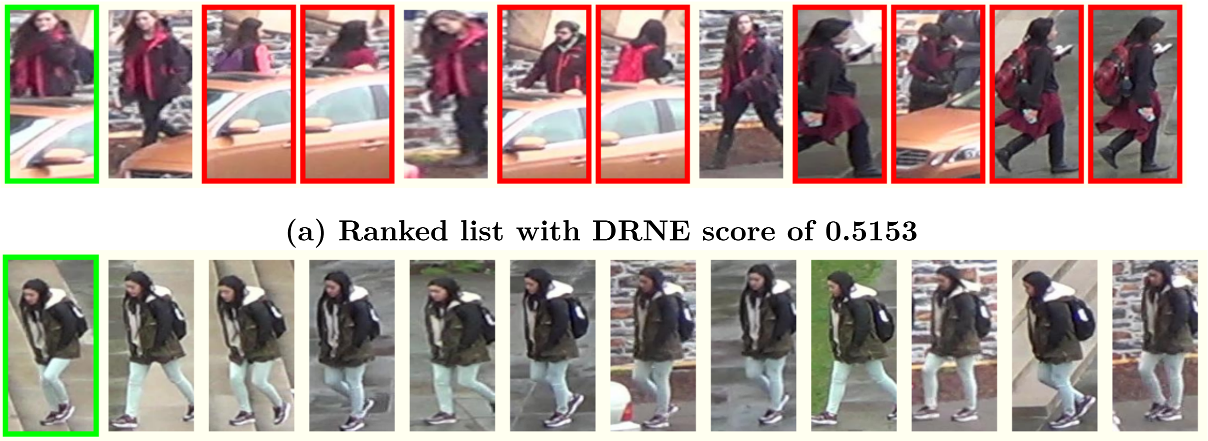


Reciprocal (Pearson = 0.7056)

Authority (Pearson = 0.7257)

DRNE+Authority+Reciprocal (Pearson = 0.7418)

Figure 5.8 – Correlation of MAP and effectiveness estimation measures on DukeMTMC.



(a) **Ranked list with DRNE score of 0.5153**

(b) **Ranked list with DRNE score of 0.9650**

Figure 5.9 – Two examples of ranked lists (good and bad queries) for Duke dataset and OSNET-AIN descriptor.

5.4.4 RQPPF Parameter Analysis

Our method requires two parameters: k, which defines the neighborhood size used for computing the measures in all the stages of the algorithm; and d, which denotes the size of the features used for training. An evaluation was conducted in order to assess the set of parameters that provided the highest Pearson correlation in relation to the MAP. In this case, the Support Vector Regression (SVR) + Reciprocal was considered for training on the synthetic data. Figure 5.10 presents the results considering the ranked lists of all descriptors on the MPEG-7 dataset. Notice that the feature size shows an asymptotic behavior that reaches stabilization closer to the value of 20. In contrast, neighborhood size presents a parabolic pattern where the highest value is also close to 20. Therefore, we adopted the value of 20 for both parameters in all the following experiments.

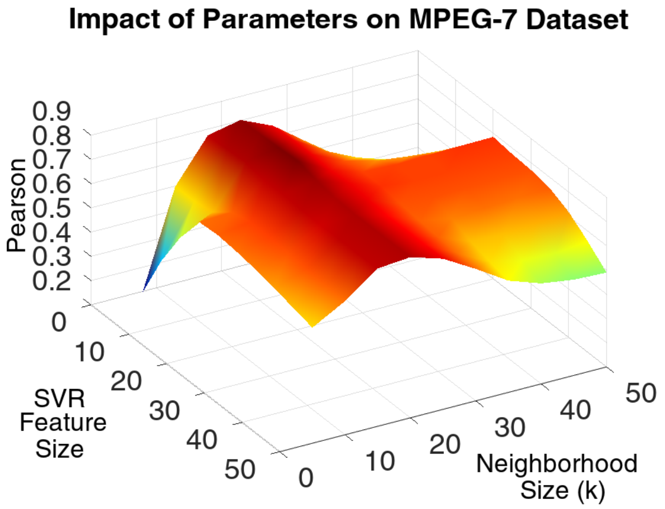


Figure 5.10 – Impact of parameters on Pearson correlation between MAP and our approach on MPEG-7 dataset (all descriptors). Through this analysis, we were able to consider the best k value (neighborhood size).

The proposed method trains a regression model with a set of meta-features computed based on different rank-based functions. Table 5.4 presents the Pearson correlation results for our approach considering different regression models and two measures: Authority [243] and Reciprocal [248]. Both MPEG-7 and Brodatz were used since they are among the smallest datasets considered. The best result for each column is highlighted in red and the second best is highlighted in blue. As can be seen, the SVR with the linear kernel provided the best results in most cases, followed by the Bayesian Ridge. Accordingly, all the remaining experiments were conducted using the SVR with a linear kernel for the proposed RQPPF.

Table 5.4 – Pearson correlation between our proposed approach and MAP considering different regression models and effectiveness estimation measures. For each column, the red values represent the highest values and the blue values represent the second-highest values found.

Regression Model	**MPEG-7**		**Brodatz**	
	+Auth.	**+Rec.**	**+Auth.**	**+Rec.**
LinearRegression	0.87833	0.89399	0.61228	0.60527
KernelRidge	0.81046	0.79683	0.46840	0.43406
BayesianRidge	**0.87854**	**0.89408**	**0.61244**	0.49098
SVR Linear [79]	**0.87975**	**0.89773**	**0.61335**	**0.60627**
SVR RBF [79]	0.84466	0.82981	0.52859	0.49610
SVR Poly [79]	0.82602	0.88899	0.55406	**0.61989**
SGD [414]	0.83519	0.79746	0.50003	0.43640
XGBRegressor [46]	0.83668	0.85111	0.51784	0.51241
LGBMRegressor [139]	0.84391	0.85384	0.52260	0.50898
CatBoostRegressor [75]	0.83835	0.85141	0.51689	0.50473
GradientBoosting [92]	0.84641	0.86428	0.52423	0.50524

5.4.5 RQPPF Results

The experimental evaluation was conducted based on the conjecture that the scores predicted by the regression model present a high Pearson correlation with ground-truth effectiveness measures (e.g., MAP, Precision). The proposed RQPPF framework can use different unsupervised effectiveness estimation measures (as described in Section 5.3.2) to compute meta-features. With the objective of analyzing the impact of each measure, we compared the results obtained by RQPPF using both Authority [243] and Reciprocal [248] scores in relation to the original measures. The results are presented in Tables 5.5 and 5.6, respectively. The relative gains are computed as $(value_{after} - value_{before})/value_{before}$. Notice that positive relative gains were obtained in all cases. The results clearly show that our approach is capable of improving the unsupervised measure results that it uses as the basis of its training.

An experiment was performed with the intent of visualizing the correlation of our RQPPF using SVR+Reciprocal with MAP. Figure 5.11 presents a plot where each dot corresponds to a different ranked list and its position depends on the MAP and the

Table 5.5 – Relative gains obtained by RQPPF using the Authority estimation measure for modeling the features.

Dataset	**Descriptor**	**Original**	**RQPPF**	
		Auth.	**+ Auth.**	**Gain (%)**
MPEG-7	**AIR** [103]	0.76392	0.79068	+3.51
	ASC [191]	0.76594	0.79472	+3.76
	IDSC [190]	0.77826	0.80696	+3.69
	CFD [244]	0.79817	0.83005	+3.99
	BAS [13]	0.79029	0.81953	+3.70
	SS [317]	0.78474	0.83188	+6.01
Brodatz	**LAS** [308]	0.64725	0.70560	+9.02
	CCOM [148]	0.63535	0.68194	+7.33
	LBP [231]	0.49609	0.52141	+5.10
Market	**CNN-OSNET** [436]	0.65170	0.65943	+1.19
	CNN-HACNN [177]	0.52763	0.54317	+2.95
	CNN-ResNet [110]	0.60783	0.60861	+0.13
	CNN-MLFN [39]	0.57916	0.58535	+1.07
	WHOS [192]	0.13383	0.16308	+21.86
Duke	**CNN-OSNET** [436]	0.64525	0.66044	+2.35
	CNN-ResNet [110]	0.69101	0.69458	+0.52
	WHOS [192]	0.00572	0.04542	+694.06

Table 5.6 – Relative gains obtained by RQPPF using the Reciprocal estimation measure for modeling the features.

Dataset	**Descriptor**	**Original**	**RQPPF**	
		Rec.	**+ Rec.**	**Gain (%)**
MPEG-7	**AIR** [103]	0.75069	0.84824	+12.99
	ASC [191]	0.81430	0.82931	+1.84
	IDSC [190]	0.80911	0.82437	+1.89
	CFD [244]	0.83621	0.85888	+2.71
	BAS [13]	0.84011	0.84934	+1.10
	SS [317]	0.81322	0.84186	+3.52
Brodatz	**LAS** [308]	0.63484	0.70560	+11.15
	CCOM [148]	0.60433	0.67188	+11.18
	LBP [231]	0.42214	0.48704	+15.37
Market	**CNN-OSNET** [436]	0.60202	0.63481	+5.45
	CNN-HACNN [177]	0.48562	0.52337	+7.77
	CNN-ResNet [110]	0.55807	0.58761	+5.29
	CNN-MLFN [39]	0.53273	0.56069	+5.25
	WHOS [192]	0.14891	0.17515	+17.62
Duke	**CNN-OSNET** [436]	0.64349	0.66518	+3.37
	CNN-ResNet [110]	0.67709	0.69373	+2.46
	WHOS [192]	0.03644	0.05878	+61.31

proposed effectiveness estimation values. The evaluation was conducted on all the ranked lists and descriptors of MPEG-7. Notice that the distribution is very linear, with a Pearson correlation equal to 0.8977. This clearly evinces the capacity of our approach to predict the effectiveness of ranked lists.

Finally, for visualization purposes, Figure 5.12 presents two examples of ranked lists where the query images are indicated by green borders and the incorrect images are indicated by red borders. The MAP is presented alongside the effectiveness obtained by our method for two ranked lists: one with higher effectiveness and one with lower effectiveness. It is possible to see that the estimations are coherent with the visual correctness of the ranked lists presented.

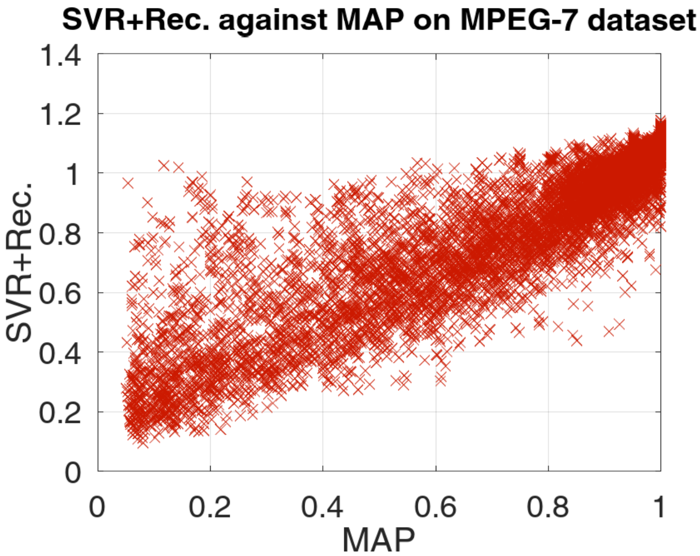
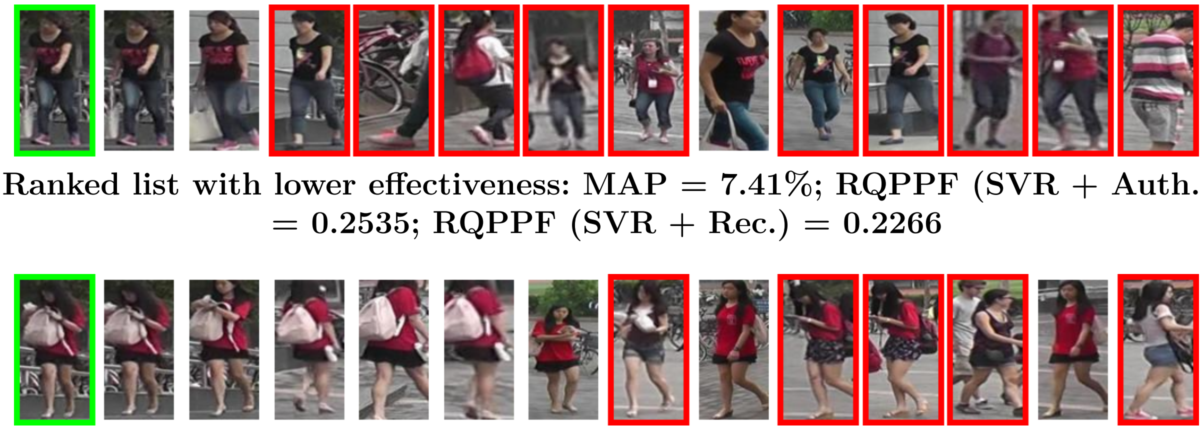


Figure 5.11 – Proposed approach against MAP on MPEG-7 dataset (all descriptors). Pearson Correlation = 0.8977.



Ranked list with lower effectiveness: MAP = 7.41%; RQPPF (SVR + Auth.) = 0.2535; RQPPF (SVR + Rec.) = 0.2266

Ranked list with higher effectiveness: MAP = 27.30%; RQPPF (SVR + Auth.) = 0.4873; RQPPF (SVR + Rec.) = 0.5330

Figure 5.12 – Two examples of RQPPF results on ranked lists of Market dataset (CNN-HACNN descriptor).

5.4.6 Joint Comparison and Discussion

In this section, the proposed methods DRNE and RQPPF are jointly evaluated alongside Authority [243] and Reciprocal [248]. The proposed methods used the same synthetic data for training. To make the comparison fair, $k = 20$ was used for all the methods. Table 5.7 presents the comparison with Authority and Reciprocal in 4 datasets, including both general-purpose and Re-ID. The values of the proposed approaches are highlighted with a gray background. The results of RQPPF are presented for two different measures (both RQPPF+Auth. and RQPPF+Rec.). The best result for each line is highlighted in red and the second best is highlighted in blue. Notice that RQPPF provided the best results in most cases, with very few exceptions. It is also possible to see that in cases where the original MAP of the descriptor is too low or too high (e.g., WHOS and AIR), it is more difficult to predict the effectiveness correctly. In these cases, DRNE obtained the best outcomes.

Table 5.7 – Comparing RQPPF and DRNE to baselines. Pearson correlation between MAP and effectiveness estimations is reported. The results of the proposed methods are highlighted with a gray background.

Descriptor	**Original MAP**	**Auth.**	**Rec.**	**DRNE**	**RQPPF +Auth.**	**RQPPF +Rec.**
			MPEG-7			
AIR [103]	89.39%	0.76392	0.75069	**0.87705**	0.79068	**0.84824**
ASC [191]	85.28%	0.76594	**0.81430**	0.74678	0.79472	**0.82931**
IDSC [190]	81.70%	0.77826	**0.80911**	0.74767	0.80696	**0.82437**
CFD [244]	80.71%	0.79817	**0.83621**	0.82587	0.83005	**0.85888**
BAS [13]	71.52%	0.79029	**0.84011**	0.79698	0.81953	**0.84934**
SS [317]	37.67%	0.78474	0.81322	**0.84026**	0.83188	**0.84186**
			Brodatz			
LAS [308]	75.15%	0.64725	0.63484	0.69576	**0.70560**	**0.70560**
CCOM [148]	57.57%	0.63535	0.60433	0.65631	**0.68194**	**0.67188**
LBP [231]	48.40%	0.49609	0.42214	**0.49984**	**0.52141**	0.48704
			Market			
OSNET [436]	43.30%	**0.65170**	0.60202	0.63451	**0.65943**	0.63481
HACNN [177]	23.30%	**0.52763**	0.48562	0.52421	**0.54317**	0.52337
ResNet [110]	22.82%	**0.60783**	0.55807	0.60246	**0.60861**	0.58761
MLFN [39]	21.98%	**0.57916**	0.53273	0.55649	**0.58535**	0.56069
WHOS [192]	6.23%	0.13383	0.14891	**0.22140**	0.16308	**0.17515**
			Duke			
OSNET [436]	52.69%	0.64525	0.64349	0.64988	**0.66044**	**0.66518**
ResNet [110]	32.00%	0.69101	0.67709	0.67628	**0.69458**	**0.69373**
WHOS [192]	2.65%	0.00572	0.03644	**0.11433**	0.04542	**0.05878**
Colors for each row:		**Second highest value**			**Highest value**	

We also performed combinations of the proposed approaches with the baselines, where each pair of effectiveness estimations (E_1, E_2) is formulated as $(E_1 + 1) \times (E_2 + 1)$ for each ranked list. Table 5.8 presents the results. The best isolated corresponds to the best result among the methods reported in Table 5.7. Abbreviations are also included: Regression for Query Performance Prediction Framework (RQPPF); Authority (A); Reciprocal (R); Deep Rank Noise Estimator (DRNE). The values reveal that the combinations further improved the Pearson correlation. However, the WHOS is still a challenge since the original MAP is very low (2.65%).

Table 5.8 – Pearson correlation between MAP and combinations of methods on Re-ID datasets.

Dataset	Descriptor	Original MAP	Best Isolated	(RQPPF+A, RQPPF+R)	(RQPPF+A, DRNE)	(RQPPF+R, DRNE)	(RQPPF+A, RQPPF+R, DRNE)
Market	**OSNET** [436]	43.30%	0.65943	0.65357	**0.66372**	0.64534	0.66164
	HACNN [177]	23.30%	0.54317	0.54410	0.55456	0.54263	**0.55666**
	ResNet [110]	22.82%	0.60861	0.61045	0.62695	0.61393	**0.62722**
	MLFN [39]	21.98%	0.58535	0.58599	0.59100	0.57574	**0.59560**
	WHOS [192]	6.23%	**0.22140**	0.16950	0.18861	0.19395	0.18374
Duke	**OSNET** [436]	52.69%	0.66518	0.66047	**0.67933**	0.67907	0.67127
	ResNet [110]	32.00%	0.69458	0.69938	**0.71094**	0.70905	0.70844
	WHOS [192]	2.65%	**0.11433**	0.05063	0.07766	0.08451	0.06834

6 Rank Correlation Measures for Manifold Learning on Image Retrieval

Recently, rank-based approaches [323, 327] have achieved highly effective retrieval results. Rank structures provide a rich source of contextual similarity information, once the most relevant information is organized at the top of ranked lists. The Ranked-List Similarities (RL-Sim) algorithm [247] exploits rank correlation measures based on the conjecture that, if two images are similar, their respective ranked lists are expected to be similar as well. In this research direction, rank correlation measures and the overlap between the neighborhood sets have been successfully exploited [323, 327] to compute more effective similarity measures in retrieval tasks.

In this scenario, the relevance of effectively quantifying the similarities between ranked lists is latent, once many manifold learning methods are based on such correlation measures. The Rank-Biased Overlap (RBO) [358] measure, based on a probabilistic user model, uses a key parameter that determines the weight for the top positions in the ranking and has been widely used. However, most of the measures are dependent on the depth of ranked lists considered or the size of the k-neighborhood set.

In this chapter, a novel rank correlation measure is proposed and validated on an unsupervised manifold learning algorithm for image retrieval. We propose a measure based on the Jaccard index, which is capable of identifying maximum similarity indications at different depths of ranked lists. Therefore, the proposed measure is more robust to the definition of the size of the neighborhood set, which is essential in unsupervised scenarios and allows the achievement of more effective results. The measure is used on a manifold learning algorithm based on a Correlation Graph (CG) and Strongly Connected Components (SCC) [249].

A wide experimental evaluation was conducted to assess the effectiveness of the proposed approach. General image retrieval and person Re-ID datasets were considered. CNN and ViT models were considered through transfer learning on unsupervised scenarios. The results demonstrated that the proposed JacMax measure achieved superior results than the RBO measure in all evaluated scenarios. The proposed approach was also evaluated on the fusion of features, achieving results comparable or superior to the state-of-the-art in most datasets.

This chapter is organized as follows: Section 6.1 discusses the main ideas and formally defines the proposed measure. Section 6.2 presents the experimental evaluation on unsupervised manifold ranking.

6.1 Proposed Method

There is a myriad of rank correlation measures proposed in the literature [323]. The most effective results on unsupervised manifold learning for retrieval tasks have been achieved by measures that consider the size of intersection/overlap at k-neighborhood sets (i.e., Jaccard). However, defining an appropriate value for k is a challenging task. Small values can lead to cuts that are unrepresentative in certain scenarios. Larger sizes, in turn, may bring in information that is not relevant.

An alternative is given by weighted measures which assign higher weights to overlaps at top positions (i.e., Intersection, RBO) [323, 247], or multi-level analysis [67]. In fact, assigning weights to top positions is a relevant strategy and improves the robustness of neighborhood size definition, but faces other difficulties in how to define the weights.

In this work, we propose to solve this challenge by identifying the depth that presents the maximum Jaccard index until a depth k. The main conjecture behind this approach is that a high overlap between ranked lists, at any depth, should be considered a strong indication of similarity. If it occurs at top positions, these are the most confident positions. If it occurs to depths closer to k, it requires a greater overlap.

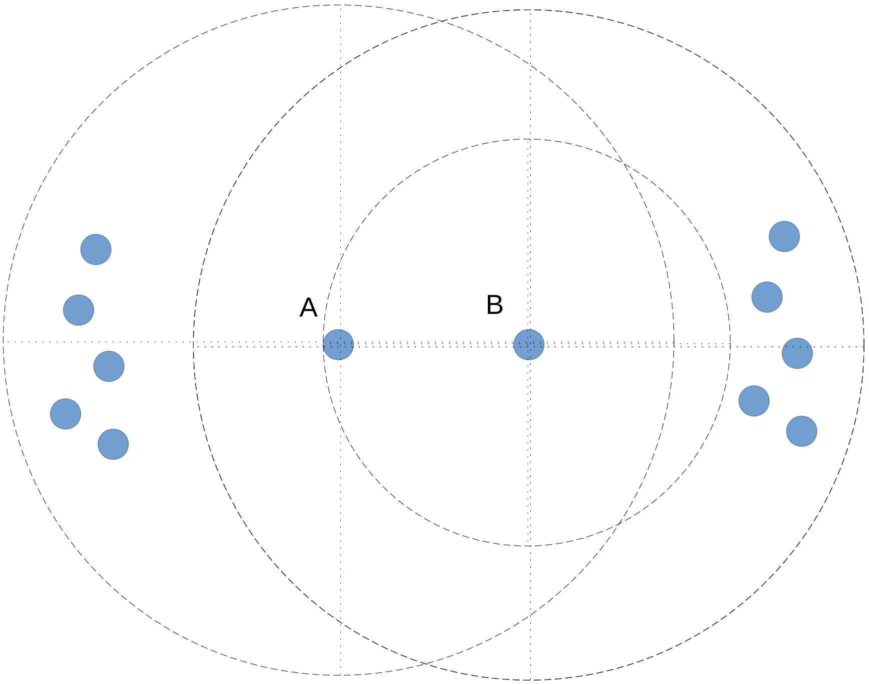
6.1.1 Jaccard Max Definition

A rank correlation measure defines a quantitative measure for assessing the similarity of two ranked lists. Given the broad use of top-k ranking analysis in retrieval applications, how to effectively compare such information assumes a fundamental relevance in many scenarios. Based on the model discussed in the previous section, a rank correlation measure can be defined as a function $\lambda : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}$. Once most measures consider the top positions of ranked lists, a set $\mathcal{N}(o_i, k)$ is used to denote the k-neighborhood set which contains the top-k elements of the ranked list τ_i.

The original Jaccard index is a traditional statistic measure that computes the correlation between two ranked lists based on the size of the intersection and the union of neighborhood sets. The index is formally defined as:

$$Jaccard(\tau_i, \tau_j, k) = \frac{|\mathcal{N}(o_i, k) \cap \mathcal{N}(o_j, k)|}{|\mathcal{N}(o_i, k) \cup \mathcal{N}(o_j, k)|}. \tag{6.1}$$

Aiming at discussing possible limitations of the original Jaccard, Figure 6.1 presents an illustrative example of data samples distributed in a 2D space according to their similarities. Points A and B are close to each other, while their respective neighborhoods are considerably distant. In this case, the Jaccard index would provide a high correlation for k=2, which encloses both A and B. In contrast, for higher k values, the Jaccard index is lower, since the neighborhoods of A and B are far apart. This is an interesting example of how the Jaccard index is susceptible to the value of k and can ignore strong indications of similarity at distinct depths for different pairs. In this case, besides A and B providing a clear indication of similarity, the Jaccard index indicates the opposite due to the distribution of neighborhood elements at higher depths.



- A and B are top-2 neighbors with Jaccard=1.0 for k=2
- However, Jaccard score is lower for higher k values (e.g. k=6)

Figure 6.1 – Illustrative example of original Jaccard index limitation.

Given this issue and inspired by the original Jaccard index, a more robust correlation measure is proposed to detect strong similarity indications at different depths of ranked lists. The Jaccard Max measure can be defined as:

$$JacMax(\tau_i, \tau_j, k) = \max_{1 \leq k_d \leq k} \frac{|\mathcal{N}(o_i, k_d) \cap \mathcal{N}(o_j, k_d)|}{|\mathcal{N}(o_i, k_d) \cup \mathcal{N}(o_j, k_d)|}. \tag{6.2}$$

The max operator is useful to ensure that the highest Jaccard similarity is returned for a given depth d. This can be used to compute a more reliable and effective similarity between elements.

6.1.2 Application on Manifold Learning

The proposed rank correlation measure is validated on an unsupervised manifold learning algorithm for image retrieval. The manifold learning algorithm [249] is based on a Correlation Graph (CG) and Strongly Connected Components (SCCs). The correlation measures are exploited to encode contextual similarity information in the graph, by assigning weight to edges. The main idea of the algorithm consists of distinguishing highly effective edges and expanding relationships through these edges.

This approach computes an unweighted directed graph where each node represents an image and the edges are defined based on a correlation measure. It applies a threshold th, such that, all nodes that provide a correlation higher than th are built. The algorithm starts with th_{start}, performing increments of th_{inc} for every iteration, until reaching th_{end}.

While the analysis of the graph edges aims to identify reliable similarity relationships, the SCCs are used to expand and identify novel relationships across the graph. Similar images are expected to be assigned to the same SCCs. In this way, the algorithm can take into account intrinsic inter-class geometry and can be more effective at measuring distances between images.

6.2 Experimental Evaluation

This section presents the protocol and evaluation of the Jaccard Max correlation measure on the Correlation Graph approach for image retrieval datasets.

6.2.1 Experimental Protocol

A wide experimental evaluation was conducted, considering 6 different public image datasets with sizes ranging from 1,491 to 36,411 images: Corel5k [194], Dogs [142], Holidays [127], UKBench [230], Market [422], and Duke [428]. Four of them are used for general image retrieval and two of them for person Re-ID. More detailed information about the datasets and effectiveness measures is provided in Chapter 4.

Concerning the parameters, we used $L = 1000$ for all general image retrieval datasets and $L = 2000$ for Re-ID datasets. The neighborhood size is $k = 50$ for Corel5k and Dogs, $k = 4$ for datasets with very few images per class (UKBench and Holidays), and $k = 20$ for Re-ID (Market and Duke). For single feature executions, we used the default parameters of the Correlation Graph: $th_{start} = 0.35$, $th_{inc} = 0.01$, and $th_{end} = 1$. For rank-aggregation, we considered: $th_{start} = 0.05$, $th_{inc} = 0.001$, and $th_{end} = 1$.

6.2.2 Results

We conducted an experiment with the objective of comparing the Correlation Graph with our proposed measure in contrast to the RBO. Table 6.1 presents the results for all the datasets and descriptors. Notice that our proposed measure achieved the highest MAP value in all cases.

An experiment was conducted performing the rank-aggregation of the best descriptors using our proposed JaccardMax. The results are shown for different effectiveness

Table 6.1 – Re-ranking results considering MAP (%).

Datasets	**Descriptors**	**Original MAP**	**Correlation Graph**	
			RBO	**JacMax**
Corel5k	**ResNet** [110]	64.50	85.93	**86.15**
	VIT-B16 [77]	75.02	88.39	**89.92**
	SWIN-TF [202]	73.92	94.11	**95.15**
Dogs	**ResNet** [110]	63.73	80.93	**82.81**
	VIT-B16 [77]	79.83	86.67	**87.48**
	SWIN-TF [202]	45.54	68.24	**69.26**
Holidays	**ResNet** [110]	74.88	71.98	**75.66**
	VIT-B16 [77]	82.40	79.71	**83.44**
	SWIN-TF [202]	85.52	82.42	**85.21**
	CNN-OLDFP [222]	88.46	86.24	**90.25**
UKBench	**ResNet** [110]	94.54	95.31	**97.17**
	VIT-B16 [77]	93.28	94.25	**96.29**
	SWIN-TF [202]	97.93	98.25	**99.01**
	CNN-OLDFP [222]	97.74	97.81	**98.92**
Market	**OSNet-AIN** [437]	43.27	42.89	**57.39**
	TransReID [111]	43.52	55.13	**55.64**
Duke	**OSNet-AIN** [437]	52.66	45.82	**68.39**
	TransReID [111]	55.42	29.39	**70.77**

measures (NS Score, R1, and MAP) in Table 6.2. Notice that combining features provided even higher results than the previous single descriptor experiment (Table 6.1).

Table 6.2 – Rank-aggregation results for different measures.

Dataset	**Features**	**NS Score**	**R1 (%)**	**MAP (%)**
Corel5k	Best Isolated Feature	—	—	75.02
	RESNET + VIT	—	—	94.96
	RESNET + SWIN-TF	—	—	95.86
	VIT + SWIN-TF	—	—	**96.32**
Dogs	Best Isolated Feature	—	—	79.83
	RESNET + SWIN-TF	—	—	81.18
	VIT + SWIN-TF	—	—	85.44
	RESNET + VIT	—	—	**88.24**
Holidays	Best Isolated Feature	—	—	88.46
	VIT + SWIN-TF	—	—	86.02
	CNN-OLDFP + SWIN-TF	—	—	90.31
	CNN-OLDFP + SWIN-TF + VIT	—	—	**91.12**
UKBench	Best Isolated Feature	3.85	—	97.93
	RESNET + SWIN-TF	3.94	—	99.05
	CNN-OLDFP + VIT	3.95	—	99.13
	CNN-OLDFP + SWIN-TF	**3.97**	—	**99.55**
Market	Best Isolated Feature	—	69.57	43.52
	OSNET-AIN + OSNET-IBN	—	73.25	59.84
	OSNET-AIN + OSNET-IBN + TReID	—	73.40	60.82
	OSNET-AIN + TReID	—	**75.42**	**63.53**
Duke	Best Isolated Feature	—	71.81	55.42
	OSNET-AIN + OSNET-IBN	—	76.21	69.27
	OSNET-AIN + OSNET-IBN + TReID	—	**78.77**	73.39
	OSNET-AIN + TReID	—	78.59	**73.96**

We used the best results obtained by our approach for comparing with the state-of-the-art. Table 6.3 presents a comparison for the Holidays dataset (MAP), where the value of 91.12% is among the best results. Table 6.4 shows the comparison for the UKBench dataset (N-S Score). The proposed method surpasses all the results presented with 3.97 which is very close to 4 (maximum value).

Table 6.3 – State-of-the-art on Holidays dataset (MAP).

MAP for the state-of-the-art methods				
Sun *et al.* [299]	Zheng *et al.* [423]	Pedronette *et al.* [241]	Li *et al.* [178]	Liu *et al.* [203]
85.50%	85.80%	86.19%	89.20%	90.89%
Yu *et al.* [398]	Gordo *et al.* [104]	Valem *et al.* [329]	Berman *et al.* [26]	**Our** **Result**
91.40%	90.30%	90.51%	**91.80%**	**91.12%**

Table 6.4 – State-of-the-art on UKBench dataset (N-S Score).

***N-S scores* for the state-of-the-art methods**				
Lv *et al.* [210]	Liu *et al.* [203]	Pedronette *et al.* [241]	Bai *et al.* [20]	Liu *et al.* [159]
3.91	3.92	3.93	3.93	3.93
Bai *et al.* [17]	Valem *et al.* [329]	Valem *et al.* [327]	Chen *et al.* [50]	**Our** **Result**
3.94	3.94	3.95	3.96	**3.97**

Table 6.5 – Comparison with person Re-ID baselines.

Method	**Year**	**Datasets**			
		Market1501		**DukeMTMC**	
		R1	**MAP**	**R1**	**MAP**
Unsupervised Methods					
EANet [118]	2018	66.4	40.6	45.0	26.4
ECN [431]	2019	75.1	43.0	63.3	40.4
UTAL [171]	2019	69.2	46.2	62.3	44.6
CAP [353]	2021	**91.4**	**79.2**	**81.1**	67.3
Domain Adaptive Methods					
HHL [430]	2018	62.2	31.4	46.9	27.2
CSGLP [273]	2019	63.7	33.9	56.1	36.0
ECN++ [432]	2020	**84.1**	**63.8**	74.0	54.4
MMCL [348]	2020	**84.4**	60.4	72.4	51.4
Cross-Domain Methods (single-source* and multi-source)**					
*EANet [118]	2018	61.7	32.9	51.4	31.7
**EMTL [370]	2018	52.8	25.1	39.7	22.3
*AF3 [195]	2019	67.2	36.3	56.8	37.4
*AF3 [195]	2019	68.0	37.7	66.3	46.2
*PAUL [380]	2019	68.5	40.1	72.0	53.2
**Baseline by [153]	2019	**80.5**	56.8	67.4	46.9
Our Proposed Approach					
Our Result		**75.42**	**63.53**	**78.59**	**73.96**

Table 6.5 shows a comparison with baselines for unsupervised person Re-ID considering MAP (%) and R-01 (%). Each result corresponds to the highest reported by the authors of the methods. For each column, the values higher than the ones obtained by our approach are highlighted in bold. The results show that our approach obtained very significant results, superior to most of the baselines for the Market dataset. For Duke, the method achieved the best MAP and the second-highest R1.

6.2.3 Visual Analysis

With the objective of visualizing the effectiveness of our approach, some ranked lists are presented where the query image is shown in green borders and the wrong results in red borders. For comparing RBO with our proposed measure, Figure 6.2 presents an example of a query. Different from our approach, notice that RBO included many wrong results among the top positions.

Similarly, we present a visualization for person Re-ID. Figure 6.3 presents three ranked lists of the same query obtained for fusion on the Market dataset considering OSNET-AIN + TransReID. Notice that our approach removed the wrong images present in the isolated descriptors.

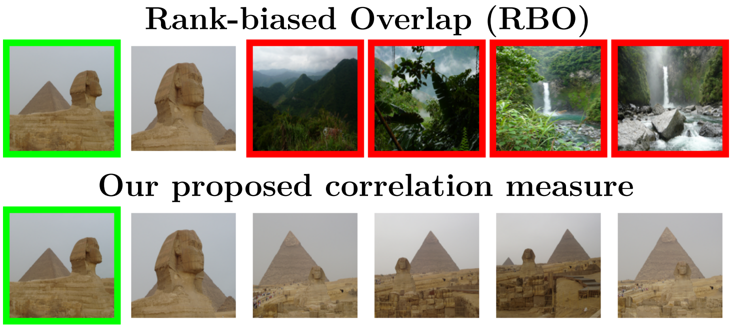


Figure 6.2 – Query on Holidays with results for RBO and JacMax.

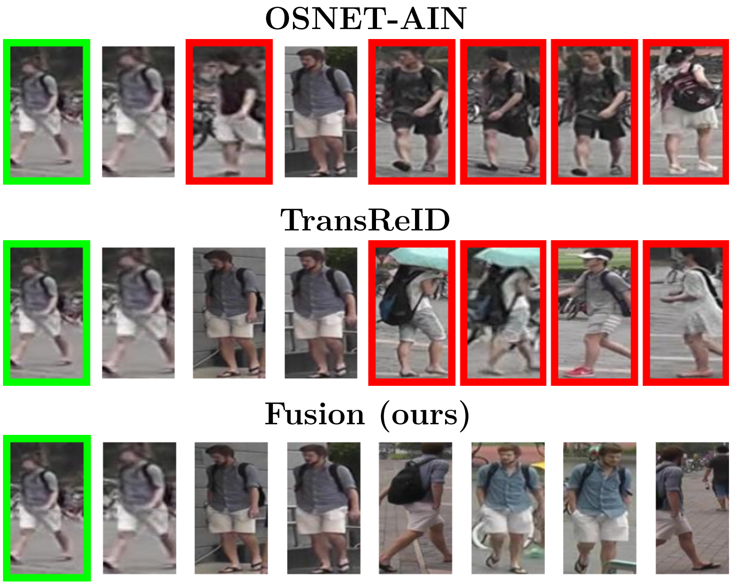


Figure 6.3 – Visual example of fusion result on Market dataset.

7 Hypergraph Rank Selection and Fusion (HRSF)

Significant progress has been made in Content-Based Image Retrieval (CBIR) systems over recent decades, particularly in feature extraction methods [294, 80, 438]. However, effective image retrieval remains challenging due to the complexity of human visual perception, which cannot be captured by a single visual feature [329, 260]. This complexity arises because images encompass multiple attributes, including color, texture, shape, and spatial relationships, and their interpretations and meanings can vary significantly depending on the context [24]. Given the wide variety of available visual descriptors, approaches for selection and fusion are essential to leverage the complementarity of different features [260].

Several fusion methods have been proposed [364, 260, 20, 413], aiming to achieve more effective retrieval results, although only some are applied for person Re-ID. Fusion approaches are typically categorized as early and late fusion. Early fusion combines raw data or extracted features, in contrast to late fusion, which merges the outputs of later processing stages, such as ranked lists or distance and similarity matrices. For early fusion in Re-ID, some approaches proposed the use of hypergraphs [9, 403]. The general idea is to compute a hypergraph for each feature and fuse them according to weights that the method has learned from training. For late fusion in Re-ID, there are strategies based on rank aggregation [389, 424], where the aggregation is performed by attributing weights for each query of each ranker [6].

However, for most of these fusion methods, no pre-selection of features [424] is performed, and all the features are used as input for the fusion step, which is generally not efficient. Selecting the optimal combination of visual features for a specific retrieval scenario is a highly challenging task due to the vast variety available [329]. It is well-known that finding the best combination of ranked lists generated by different features is an NP-hard problem [318, 83]. As the number of features increases linearly, the number of possible combinations increases exponentially. Even with supervised methods, selecting the most effective combinations of visual features is a challenging task [5]. This complexity arises from the need to consider multiple factors, such as the diversity and complementarity of features. The task becomes even more difficult in an unsupervised setting, where the absence of labeled data makes it necessary to focus exclusively on unlabeled data.

This chapter addresses the challenging task of unsupervised selection and fusion of

[6] In this work, a ranker is a set of ranked lists computed from features, as described in Section 2.2.3.

different features for more effective person re-identification. We propose a novel Hypergraph Rank Selection and Fusion (HRSF) [331] framework, which combines an unsupervised rank-based formulation for feature selection [329] with a robust hypergraph model [251] for query performance prediction and rank aggregation based on manifold learning. Among our main contributions, we can highlight:

- The proposed HRSF framework uses a rank-based late fusion model, suitable for selection and fusion of a broad diversity of features. The selection is performed by exploiting an unsupervised measure for query performance prediction;
- A hypergraph rank-based formulation is used to encode the high-order relationship among images. This strategy is exploited for both selection and fusion tasks. Hypergraph models were little exploited in Re-ID literature [9, 403];
- The proposed technique is able to learn representations through the hypergraph structure that encodes multiple features from different rankers. The manifold learning based on a hypergraph model [251] allows effective fusion and final ranking. In addition, our approach innovates by fusing different feature extractors trained on different datasets;
- Different from most fusion approaches for Re-ID which often consider ad hoc selections or combine all the features of the input set [389, 424, 403], our approach is capable of dealing with various features in a completely unsupervised scenario, selecting combinations in a very large search space. To the best of our knowledge, this is the first work that performs an explicit selection and subsequently fusion of features on person Re-ID tasks in a completely unsupervised way.

The proposed Hypergraph Rank Selection and Fusion (HRSF) approach is based on a general framework for rank selection and fusion [329] and a hypergraph-based manifold learning approach [251]. However, while some aspects are in common, there are also crucial differences. Among them, we can mention:

- The problem of selecting and fusing rankers in unsupervised scenarios is very challenging. Hence, the use of an effective algorithm for selection is fundamental. While Unsupervised Selective Rank Fusion (USRF) [329] employs traditional query performance prediction approaches (Authority and Reciprocal scores), we propose a new measure named *Hypergraph Query Performance Prediction* (HQPP);
- Originally, USRF [329] uses an approach based on Cartesian product of ranking references [332] for fusion tasks. Differently, the proposed HRSF uses the Log-based Hypergraph of Ranking References (LHRR) [251] method, which in combination with the HQPP, makes both the selection and fusion based on hypergraph structures. The

LHRR [251] method is also more robust than the CPRR [332], achieving superior retrieval results in most datasets;

- Both [329] and [251] were originally proposed and evaluated only on general-purpose image retrieval scenarios. The proposed HRSF is employed and validated for person Re-ID tasks;
- In unsupervised scenarios, the optimal neighborhood size (k parameters) can be challenging to define. The experiments revealed that HRSF is more robust than HQPP to different neighborhood sizes, leading to the most effective results in the majority of scenarios.

A wide experimental evaluation was conducted on 4 different datasets with sizes ranging from 14,097 to 39,902 images. Up to 28 different rankers were considered in each case, resulting in millions of possible combinations. Experiments indicated that our approach was capable of selecting and fusing the rankers, achieving highly effective results superior to all rankers in isolation and competitive to state-of-the-art when more than 20 Re-ID approaches are considered.

The chapter is organized as follows: Section 7.1 presents the proposed approach for rank selection and fusion in Re-ID. Section 7.2 discusses the conducted experimental evaluation.

7.1 Proposed Method

This work proposes a framework named Hypergraph Rank Selection and Fusion (HRSF) for unsupervised person Re-ID tasks. Our model is inspired by a recent approach [329] proposed for rank selection and fusion on general image retrieval tasks. The method is based on effectiveness estimations and correlation among features computed by a rank-based analysis. In [329], reciprocal references are exploited for effectiveness estimation and feature selection, while rank correlation measures are used for analyzing complementarity and diversity aspects. The selected features are fused through a rank-based similarity learning method [332].

The proposed approach differs from previous work [329] on four main aspects: (*i*) the unsupervised measure used to estimate the quality of individual features, which is based on hypergraph structures; (*ii*) a more robust method to fuse the selected features; (*iii*) the proposed approach is evaluated and validated for person Re-ID and; (*iv*) it is more robust to different neighborhood sizes, leading to the most effective results in the majority of scenarios. We innovate by employing a robust hypergraph model for both tasks: query performance prediction and rank fusion. A recent manifold learning approach based on a rank-based hypergraph formulation [251] is exploited.

The hyperedges weights, used for estimating the confidence of hyperedge associations, are exploited in our approach to predict the effectiveness of the different person Re-ID rankers in an unsupervised fashion. Additionally, we use a manifold learning algorithm for fusion tasks, the LHRR [251]. This keeps our approach completely unsupervised, and more robust, mostly based on hypergraph structures.

Figure 7.1 presents an overview of the proposed approach, where the main steps are illustrated and enumerated. Given a set of different rankers provided by diverse feature extractors and distance measures, (1) a hypergraph estimation measure is employed in order to predict the performance of each ranker without using data labels. In (2), a correlation measure is applied for each pair of rankers. The computed measures are used in the equation presented in step (3), which computes the equation for each combination. The rankers selected in stage (3) are fused in stage (4), which uses LHRR [251] for rank aggregation.

The next subsections detail each of the steps of HRSF outlined in the figure. To support the reader and assist in understanding the formulas, Table 7.1 provides a summary of all the symbols used in the formulation of HRSF.

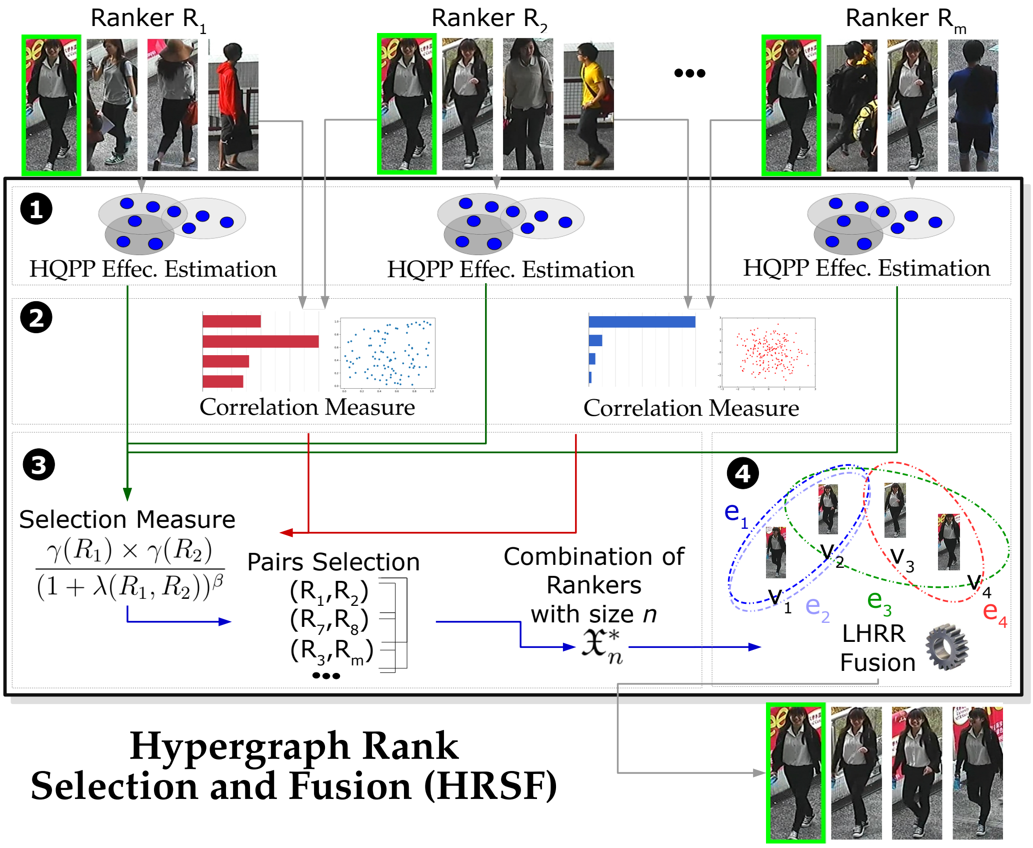


Figure 7.1 – Overview of the HRSF proposed approach.

Table 7.1 – Table of symbols used in the definition of HSRF [331].

Type	Symbol	Description
Retrieval Model	$\mathcal{C}$	Image collection.
	N	Image collection size.
	o_i	Image of index i.
	$\mathcal{N}(q, k)$	Neighborhood set for a query image o_q of size k.
	τ_q	Ranked list for the query image o_q.
	$\tau_q(j)$	Position of the image o_j in the ranked list of the image o_q.
	L	Size of the ranked lists.
	$\mathcal{T}$	Set of ranked lists for all the images in the dataset.
Selection Model	f_s	Function for ranker selection.
	R_i	Ranker of index i.
	$\tau_{i,q}$	Ranked list of the image o_q computed by the ranker o_i.
	$\mathcal{T}_i$	Set of ranked lists produced by the ranker R_i.
	$\mathcal{R}$	Set of rankers.
	m	Size of the set $\mathfrak{R}$.
	$\mathfrak{X}_n$	Candidate combination composed by n rankers.
	$\mathfrak{X}_n^*$	Selected combination composed by n rankers.
	$\mathfrak{X}^*$	Selected combination among all sizes.
	n	Size of a combination.
	w_p	Selection measure for pairs of rankers.
	β	Weight or relevance of the correlation.
	k	Neighborhood size.
	γ	Effectiveness estimation measure (HQPP).
	λ	Correlation measure (RBO).
	μ	Constant used in RBO correlation measure.
Hypergraph Model	V	Set of vertexes.
	v_i	Vertex of index i.
	E_h	Set of hyperedges.
	e_i	Hyperedge of index i.
	$h(e_i, v_j)$	Reliance of vertex v_j to belong to a hyperedge e_i.
	$r(e_i, v_j)$	Density of ranking references to v_j in ranking of o_i and its neighbors.
	$\mathbf{H_G}$	Hypergraph model.
	$\mathbf{H}$	Incidence matrix.
	$\eta_r(i, x)$	Function that assigns a weight to image x according to its position in τ_i.
	η_f	Fused affinity measure used for rank aggregation.

7.1.1 Unsupervised Ranker Selection

Given a set of available rankers for person Re-ID and no labeled data, we aim to select a combination that produces the most effective results. The selection measure proposed in [329] relies on the idea that rankers can be analyzed in pairs using an effectiveness estimator and a correlation measure. It consists of attributing weight to each pair (w_p), in such a way that the ones composed by the most effective rankers and the highest/lowest correlated ones should receive a higher score. This is presented in Equation (7.1).

$$w_p(\{R_1, R_2\}) = \frac{\gamma(R_1) \times \gamma(R_2)}{(1 + \lambda(R_1, R_2))^{\beta}}, \tag{7.1}$$

where γ and λ are used to measure the effectiveness and correlation of rankers, respectively. The γ corresponds to our proposed Hypergraph Query Performance Prediction (HQPP), which is described in Section 7.1.2. While the effectiveness can be individually estimated for each ranker, the correlation measure is applied to pairs. The exponent β can be employed to decide if the selection favors the most correlated or diverse rankers. We adopt $\beta = -1$, since in [329] it was used for scenarios with a higher number of features.

Therefore, in our approach, a ranker pair is selected based on the score obtained by w_p. All ranker pairs are ranked according to w_p (in descending order). Only the pairs with the highest scores are selected. The user can choose the number of top combinations to be selected.

After the selection of a pair of rankers, which is denoted by $\mathfrak{X}_2^*$, combinations of other sizes are selected by performing intersection and union operations, in a procedure detailed described in [329].

For computing the correlation between rankers, the Rank-Biased Overlap (RBO) [358] measure is used. This measure considers the overlap between top-k lists at increasing depths. The weight of the overlap is calculated based on probabilities defined at each depth. It can be formally defined as follows:

$$\lambda(\tau_i, \tau_j, k, \mu) = (1 - \mu) \sum_{d=1}^{k} \mu^{d-1} \times \frac{|\mathcal{N}(i,k) \cap \mathcal{N}(j,k)|}{d}, \tag{7.2}$$

where $\mathcal{N}(i,k)$ denotes the natural neighborhood of the top-k images for o_i and μ is a constant ($\mu = 0.9$ was used for all the experiments).

In this work, rather than the effectiveness estimations and the fusion method employed in [329], we used a query performance prediction measure (HQPP) and a recent manifold ranking aggregation method (LHRR) that model the ranked lists through hypergraph structures [251]. Both are discussed in the next sub-sections.

7.1.2 Hypergraph Query Performance Prediction

Query Performance Prediction (QPP) can be broadly defined as the task of estimating the effectiveness of a search/retrieval operation performed in response to a query, where no labeled data is available [285]. Initially proposed for textual retrieval systems [439, 440], the task assumed a diversified taxonomy in the literature and has been established as a promising approach in image retrieval systems [66].

In this work, we propose to use a Hypergraph Query Performance Prediction (HQPP) score for predicting the effectiveness of rankings produced by person Re-ID features. The HQPP score uses a hypergraph formulation recently proposed [251] for manifold ranking on multimedia retrieval. Hypergraphs are a robust generalization of graphs, providing a powerful tool for capturing high-order relationships in several domains [120, 277, 298]. In opposition to traditional graph-based approaches, which represent only pairwise relationships, hypergraphs allow connecting any number of nodes in order to represent similarity among sets of objects [34].

This work and the HQPP score use a hypergraph model mainly based on the following main hypotheses and ideas:

- Similar objects present similar ranked lists and, therefore, similar hyperedges. Once the hyperedges are represented by an incidence matrix, the product of the hyperedges can be exploited to compute a more effective similarity measure between nodes;
- Similar objects are expected to reference each other in the same hyperedge. Therefore, hyperedges that concentrate a high number of ranking references on a few nodes are expected to be more effective. The Hypergraph Query Performance Prediction (HQPP) is formally defined based on this conjecture.

Following the definition of [251], as also described in Section 2.6, a hypergraph can be defined as a tuple $\mathbf{H_G} = (V, E_h, h_p)$, where V represents a set of vertices and E_h denotes the hyperedge set. The set of hyperedges E_h can be defined as a family of subsets of V such that $\bigcup_{e \in E_h} = V$. To each hyperedge e_i, a positive score $h_p(e_i)$ denotes the confidence of relationships among a set of vertices established by the hyperedge e_i.

While graphs are commonly represented by adjacency matrices, hypergraphs are often represented by incidence matrices. The incidence of a hyperedge e_i on a vertice v_j is represented by an incidence matrix $\mathbf{H}$, defined as follows:

$$h(e_i, v_j) = \begin{cases} r(e_i, v_j), & \text{if } v_j \in e_i, \\ 0, & \text{otherwise,} \end{cases} \tag{7.3}$$

where $h(e_i, v_j)$ denotes the reliance of the vertex v_j to belong to a hyperedge e_i and $r(e_i, v_j)$ is a function with a codomain in the $\mathbb{R}^+$ that indicates the degree to which the vertex v_j belongs to a hyperedge e_i. A hyperedge e_i is defined for each image $o_i \in \mathcal{C}$ based on the k-neighborhood set of o_i and its respective neighbors. In this context, the function $r(e_i, v_j)$ is defined based on the density of ranking references to v_j in the ranking of o_i and its neighbors. Formally, the function is defined as:

$$r(e_i, v_j) = \sum_{y \in \mathcal{N}(i,k) \wedge j \in \mathcal{N}(y,k)} \eta_r(i, y) \times \eta_r(y, j), \tag{7.4}$$

where $\eta_r(i, x)$ is a function that assigns a weight of relevance to image x according to its position in the ranked list τ_i. The weight assigned to x according to its position in the ranked list τ_i is defined as follows:

$$\eta_r(i, x) = 1 - \log_k \tau_i(x). \tag{7.5}$$

The size of the hyperedges varies according to the number of co-occurrences of images. A high diversity of elements may indicate a high degree of uncertainty and this information will be exploited for defining the weights of hyperedges. The weight of a hyperedge $h_p(e_i)$ denotes the confidence of relationships established among vertices by the hyperedge.

In order to compute the weight $h_p(e_i)$, we use the Hypergraph Neighborhood Set $\mathcal{N}_h$, which contains the k vertices with the greatest $h(e_i, \cdot)$ scores in the hyperedge e_i. As such, the hyperedge weight $h_p(e_i)$ is defined as:

$$h_p(e_i) = \sum_{j \in \mathcal{N}_h(i,k)} h(i,j). \tag{7.6}$$

A high-effective hyperedge is expected to present an elevated value of h_p, indicating a consistent co-occurrence of the same elements with high confidence of membership. Therefore, the hyperedge weight $h_p(e_i)$ is defined as the Hypergraph Query Performance Prediction (HQPP) for a ranked list of image o_i, which is denoted by γ:

$$\gamma(\tau_i) = h_p(e_i). \tag{7.7}$$

For a given ranker R_i, the γ can be computed for all the ranked lists to obtain the value of $\gamma(R_i)$ in Equation (7.1). We highlight that HQPP was used to define γ in this work, but our approach is flexible and capable of supporting other measures.

7.1.3 Hypergraph Manifold Rank Aggregation

Once the person Re-ID features are selected, we fuse the respective produced rankings through a recently proposed manifold learning algorithm [251]. The Log-based Hypergraph of Ranking References (LHRR) [251], briefly described in this section, captures the dataset manifold structure through a hypergraph-based similarity measure, which can be used to rank aggregation tasks.

LHRR [251] exploits the hypergraph formulation discussed in the last section to represent high-order similarity relationships encoded in the dataset manifold. Subsequently, pairwise similarity scores are computed, allowing more effective ranking results. The pairwise similarity is computed based on the conjecture that similar elements present similar hyperedge representations. The similarity between hyperedges is computed based on the product of the incidence matrix $\mathbf{H}$ and its transpose to encode reciprocal relationship. A pairwise similarity matrix $\mathbf{S}$ is computed as:

$$\mathbf{S} = (\mathbf{H}\mathbf{H}^T) \circ (\mathbf{H}^T\mathbf{H}) \tag{7.8}$$

In addition to the product of hyperedges, a Cartesian product operation is conducted to extract useful pairwise relationships directly from the set of elements defined by the hyperedges. Given two hyperedges $e_q, e_i \in E_h$, the Cartesian product between them can be defined as:

$$e_q \times e_i = \{(v_x, v_y) : \; v_x \in e_q \wedge v_y \in e_i\}. \tag{7.9}$$

Let ${e_q}^2$ denote the Cartesian product between the elements of the same hyperedge e_q, for each pair of vertices $(v_i, v_j) \in {e_q}^2$ a pairwise similarity relationship cp is computed to define the membership degrees of v_i and v_j. The function is formally defined as:

$$cp(e_q, v_i, v_j) = h_p(e_q) \times h(e_q, v_i) \times h(e_q, v_j). \tag{7.10}$$

A similarity measure based on a Cartesian product is defined through a matrix $\mathbf{C}$, with each position computed as follows:

$$c(i, j) = \sum_{e_q \in E_h \wedge (v_i, v_j) \in {e_q}^2} cp(v_i, v_j). \tag{7.11}$$

The pairwise similarity defined based on hyperedges and Cartesian product operations provides complementary information. Hence, an affinity matrix $\mathbf{W}$ is computed by combining both matrices as:

$$\mathbf{W} = \mathbf{C} \circ \mathbf{S}. \tag{7.12}$$

Based on the affinity measure defined by $\mathbf{W}$, a ranking procedure can be performed for each feature giving rise to a new set of ranked lists. Next, a multiplicative rank-based formulation is used to combine the features, exploiting an adaptive weight, which is assigned to each query/feature according to the weight of the respective hyperedge. Let η_f denote the fused affinity measure; each element is computed as follows considering the top-L positions of τ_q:

$$\eta_f(q, i) = \prod_{f=1}^{m} \frac{(1 + h_p(f, e_q))}{(1 + \log_L \tau_{q,f}(i))}, \tag{7.13}$$

where $h_p(f, e_q)$ is the weight of hyperedge e_q according to the feature f and $\tau_{q,f}(i)$ denote the position of o_i in the ranked list of o_q according to the feature f. The combined affinity measure $\eta_f(\cdot, \cdot)$ gives rise to a unique set of ranked lists which is re-processed by the LHRR [251] algorithm as a single feature.

7.2 Experimental Evaluation

This section presents the experimental results conducted to evaluate our proposed approach. The experimental evaluation was conducted on 4 person Re-ID datasets (described in Section 4.2.2) with sizes ranging from 14,097 to 39,902. For each dataset, up to 28 rankers were considered of different modalities (e.g. traditional descriptors, bag of visual words, deep learning), which consists of all the Re-ID descriptors mentioned in Section 4.2.2, except TransReID. The large number of rankers is used with the objective of evaluating the capacity of our selection approach. It is desirable that, if the selection is accurate, only the most effective are selected to be fused. Also, there is a very large number of possible combinations, when all the possible sizes are considered. With 28 rankers, there are 268,435,456 possible combinations. Since it is impractical to execute all of them, selection is of fundamental importance in this context. The experimental evaluation also considers a comparison with fusion baselines and with state-of-the-art Re-ID approaches.

7.2.1 Experimental Analysis

The neighborhood size, denoted by k, is used in multiple steps of our method: for calculating the effectiveness measure, for the correlation measure, and in the fusion stage. Figure 7.2 presents an experiment that was conducted to evaluate the impact of k on the Market1501 dataset, where both R1 and MAP are shown for different values of k. Notice that the method is robust to different parameter settings. We used $k = 20$ for CUHK03, Market1501, DukeMTMC, and $k = 10$ for Airport in all of the remaining experiments. For the Airport dataset, a smaller k seems to be more adequate, since it has fewer images per individual (around 4) compared to the other collections.

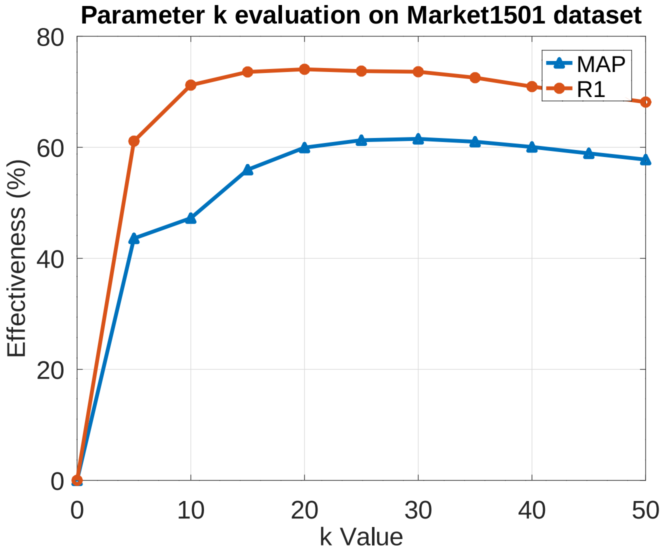


Figure 7.2 – Evaluation of the impact of parameter k on MAP and R1 for Market1501 dataset.

In this work, HQPP is proposed as a measure to estimate the quality of each ranker in the selection stage. Aiming at assessing the use of this measure in Re-ID scenarios, Figure 7.3 shows an experiment where each dot corresponds to a different ranker and the MAP (a measure that uses labeled data) is compared to the HQPP performance prediction score. As can be seen in the graph, there is a high correlation between the measures. The Pearson correlation among the dots is 0.9678, which indicates the high effectiveness of the selection strategy.

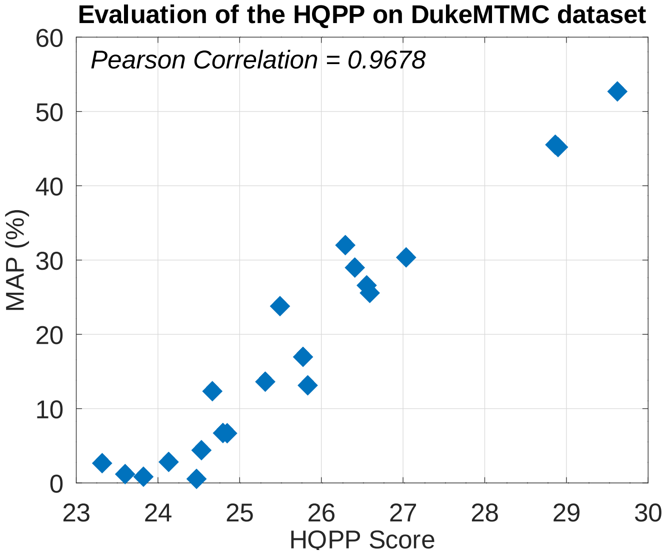


Figure 7.3 – Evaluation of the HQPP measure compared to the MAP on DukeMTMC dataset.

The HRSF ranks the best combinations for each size. The user can choose the number of top combinations to be selected. We conducted an experiment on CUHK03, Market, and DukeMTMC (Figures 7.4, 7.5, and 7.6) where the average MAP and R1 of the selected ranker pairs are presented as the number of selected pairs changes. Notice that the highest MAP and R1 values are in the first position (top-1), which evinces that the combination with the highest w_p (ranked in the first position) is also the one with the highest effectiveness.

The best combination available in the top-5 for each size is reported in Table 7.2. We report sizes from 1 to 6 ($\mathfrak{X}^*_2$, ..., $\mathfrak{X}^*_6$) and the best combination among them (which can be denoted just as $\mathfrak{X}^*$) is highlighted in bold. The best isolated ranker in each case is also listed for comparison purposes and to facilitate the visualization of the relative MAP gain. Notice that OSNET, OSNET-AIN, and OSNET-IBN are the most commonly selected rankers, which evinces the effectiveness of our selection, once these rankers are among the most effective ones. Additionally, in all cases, the proposed selection and fusion achieved better results than the best ranker in isolation. The complementarity among the methods can be exploited by our approach achieving gains up to +47% (MAP) after the selection and fusion are performed. The results also indicate that gains can be obtained when networks trained on different datasets are combined, even when the same architecture is used.

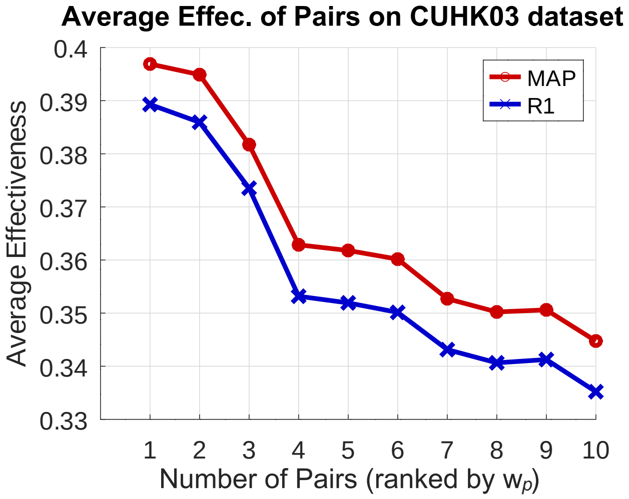


Figure 7.4 – Average MAP of top pairs on CUHK03 dataset.

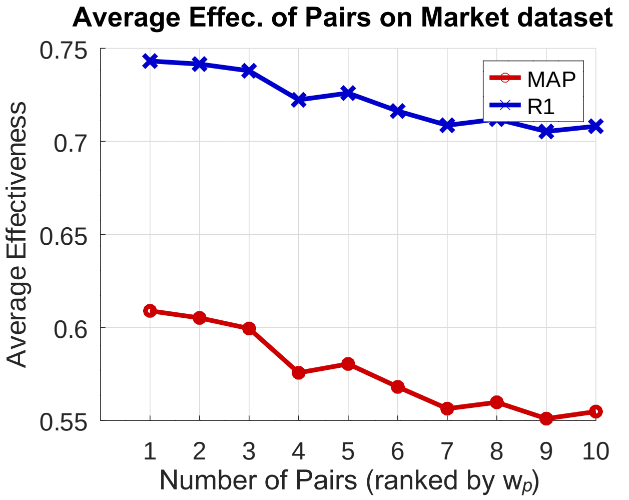


Figure 7.5 – Average MAP of top pairs on the Market dataset.

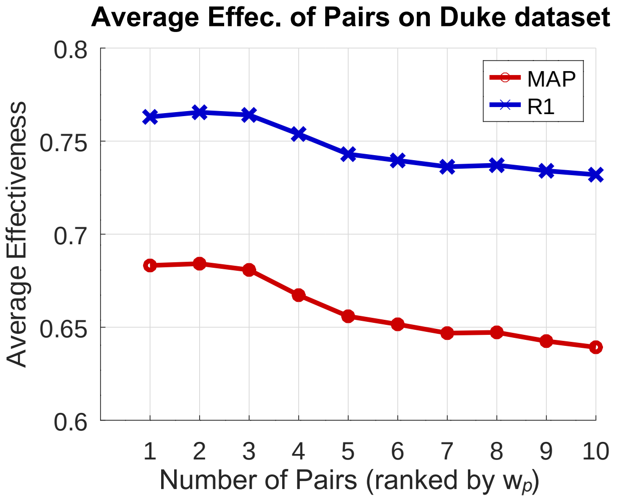


Figure 7.6 – Average MAP of top pairs on Duke dataset.

Table 7.2 – The best selected combination of each size (among the top-5) is reported on each dataset.

Dataset	**Comb. Size**	**Selected and Fused Rankers**	**R1 (%)**	**MAP (%)**	**MAP R. Gain**
CUHK03	Best R	OSNET-AIN (MT)	28.49	27.00	—
	$\mathfrak{X}_2^*$	**OSNET-AIN (MT) + OSNET-IBN (MT)**	**39.04**	**39.69**	**+47.00%**
	$\mathfrak{X}_3^*$	OSNET-AIN (MT) + OSNET-IBN (MT) + OSNET (MT)	39.13	39.58	+46.59%
	$\mathfrak{X}_4^*$	OSNET-AIN (MT) + OSNET-IBN (MT) + OSNET (MT) + OSNET-AIN (M)	38.02	38.80	+43.70%
	$\mathfrak{X}_5^*$	OSNET-AIN (MT) + OSNET-IBN (MT) + OSNET (MT) + OSNET-AIN (M) + OSNET-IBN (M)	36.15	37.11	+37.44%
	$\mathfrak{X}_6^*$	HACNN (MT) + OSNET-AIN (D) + OSNET-AIN (M) + OSNET-AIN (MT) + OSNET-IBN (MT) + OSNET (MT)	35.46	36.17	+33.96%
Market1501	Best R	OSNET-AIN (MT)	69.95	43.30	—
	$\mathfrak{X}_2^*$	OSNET-AIN (MT) + OSNET (MT)	74.32	60.89	+40.62%
	$\mathfrak{X}_3^*$	OSNET-AIN (MT) + OSNET (MT) + OSNET-AIN (D)	75.56	62.64	+44.67%
	$\mathfrak{X}_4^*$	**OSNET-AIN (MT) + OSNET (MT) + OSNET-AIN (D) + OSNET-IBN (MT)**	**75.71**	**62.94**	**+45.36%**
	$\mathfrak{X}_5^*$	OSNET-AIN (MT) + OSNET (MT) + OSNET-AIN (D) + OSNET-IBN (MT) + HACNN (D)	74.00	60.69	+40.16%
	$\mathfrak{X}_6^*$	HACNN (MT) + OSNET-AIN (D) + OSNET-AIN (MT) + OSNET-IBN (D) + OSNET-IBN (MT) + OSNET (MT)	73.57	59.85	+38.22%
DukeMTMC	Best R	OSNET-AIN (MT)	71.14	52.69	—
	$\mathfrak{X}_2^*$	OSNET-AIN (MT) + OSNET-IBN (MT)	76.80	68.51	+30.02%
	$\mathfrak{X}_3^*$	**OSNET-AIN (MT) + OSNET-IBN (MT) + OSNET (MT)**	**77.24**	**68.88**	**+30.73%**
	$\mathfrak{X}_4^*$	OSNET-AIN (MT) + OSNET-IBN (MT) + OSNET (MT) + RESNET (MT)	76.89	68.56	+30.12%
	$\mathfrak{X}_5^*$	OSNET-AIN (MT) + OSNET-IBN (MT) + OSNET (MT) + RESNET (MT) + OSNET-AIN (M)	76.39	67.72	+28.53%
	$\mathfrak{X}_6^*$	OSNET-AIN (MT) + OSNET-IBN (MT) + OSNET (MT) + RESNET (MT) + OSNET-AIN (M) + MLFN (MT)	75.90	66.96	+27.08%
Airport	Best R	OSNET-AIN (MT)	—	52.26	—
	$\mathfrak{X}_2^*$	OSNET-AIN (MT) + OSNET (MT)	—	52.43	+0.33%
	$\mathfrak{X}_3^*$	OSNET-AIN (MT) + OSNET (MT) + OSNET-IBN (MT)	—	53.38	+2.14%
	$\mathfrak{X}_4^*$	OSNET-AIN (MT) + OSNET (MT) + OSNET-IBN (MT) + OSNET-AIN (M)	—	53.91	+3.16%
	$\mathfrak{X}_5^*$	**OSNET-AIN (MT) + OSNET (MT) + OSNET-IBN (MT) + OSNET-AIN (M) + HACNN (MT)**	—	**54.09**	**+3.50%**
	$\mathfrak{X}_6^*$	OSNET-AIN (MT) + OSNET (MT) + OSNET-IBN (MT) + OSNET-AIN (M) + HACNN (MT) + MLFN (MT)	—	54.02	+3.37%

7.2.2 Comparison with Fusion Baselines

In order to evaluate the proposed method compared to other approaches that both select and fuse the input features, Table 7.3 presents the proposed approach, HRSF, compared to both early and late fusion baselines on the four datasets. In all the cases, the same set of features, which were presented in the experimental protocol, were used. For the early fusion methods, the default parameters were used and all the features were processed with PCA to reduce the feature vectors to 100 components. From all the features, the top-1000 were selected to compose the new feature vector, and the Euclidean distance was computed. As can be seen, the results of our approach are superior in most cases (CUHK03, Market1501, Airport) and comparable in others (DukeMTMC).

An experiment was conducted with the objective of clarifying the robustness of HRSF to different values of k when compared to USRF. Figures 7.7 and 7.8 present the MAP of the best combination ($\mathfrak{X}^*$) among top-5 for different values of k on Market and Duke datasets, respectively. The MAP for $k = 20$ is the same as in Table 7.3. Notice that, while both methods seem comparable for $k = 20$, our method provided a significantly higher MAP for other values of k. This is fundamental for unsupervised scenarios, where the optimal k can be challenging to define.

Table 7.3 – Proposed approach compared to early and late fusion baselines.

Dataset	Category	Method	R1 (%)	MAP (%)
CUHK03	**Early Fusion**	Laplace [112]	9.56	10.19
		SPEC [421]	9.29	9.97
	Late Fusion	USRF [329]	38.24	39.03
		HRSF (ours)	**39.04**	**39.69**
Market1501	**Early Fusion**	Laplace [112]	**82.07**	61.26
		SPEC [421]	77.14	54.90
	Late Fusion	USRF [329]	75.97	62.69
		HRSF (ours)	75.71	**62.94**
DukeMTMC	**Early Fusion**	Laplace [112]	59.29	43.56
		SPEC [421]	59.29	43.56
	Late Fusion	USRF [329]	**77.82**	**68.98**
		HRSF (ours)	77.24	68.88
Airport	**Early Fusion**	Laplace [112]	—	45.33
		SPEC [421]	—	45.32
	Late Fusion	USRF [329]	—	39.75
		HRSF (ours)	—	**54.09**

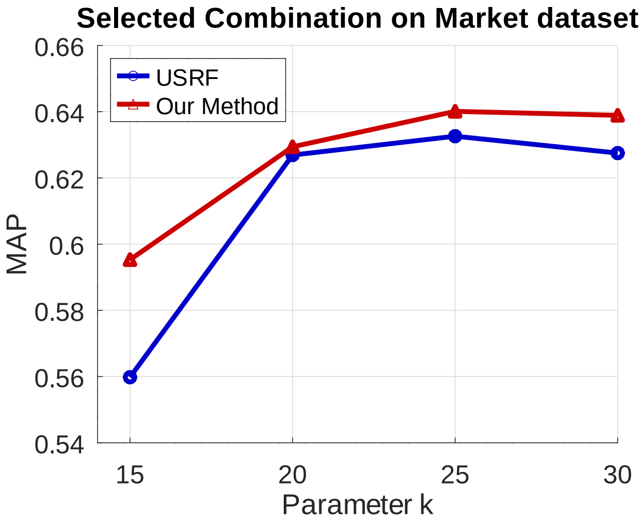


Figure 7.7 – Selected Combination (among top-5) on Market considering MAP.

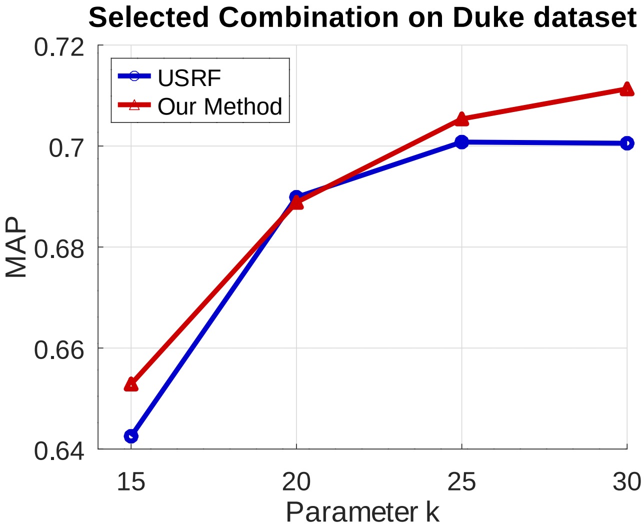


Figure 7.8 – Selected Combination (among top-5) on Duke considering MAP.

7.2.3 State-of-the-Art

This section presents comparisons with Re-ID state-of-the-art approaches. The taxonomy often varies in the literature, there are different subcategories of unsupervised Re-ID methods. However, since they are all unsupervised and often there is overlap among the categories, we insert all of them into a single group named Unsupervised Domain Adaptation Methods [22]. Table 7.4 presents our results compared to around 20 state-of-the-art Unsupervised Domain Adaptation Methods [22]. The gray cells with bold values correspond to methods that have outperformed the best HRSF result.

To perform a fair comparison, it contains only methods that did not use the labels from the target dataset for training (train on CUHK03 and test on CUHK03 is not used, for example). Therefore, supervised and semi-supervised methods are not included. The abbreviations in parentheses indicate the datasets used for training [7]. For example, the use of (D, M) indicates that the reported result corresponds to a training done either on Duke or on the Market dataset. The results reported on Market were trained on Duke and the results reported on Duke were trained on Market. None of the presented methods were trained using labels from the target dataset. The abbreviations were omitted for baselines that used more than 5 datasets as sources for training (CAMEL [396] and baseline by [153]), but they can be consulted in [153], which used similar baselines and protocol.

We provided the best results for each method (considering the original papers) to keep the evaluation as far as possible. Since the code and implementation are not available for the majority of methods, it is not possible to provide results considering the same sources in all cases.

Both MAP and R1 are reported in all the cases and our best results are presented in bold. The baselines do not perform any form of selection. We highlight that our method

[7] C02 = CUHK02, C03 = CUHK03, M = Market1501, D = DukeMTMC, MT = MSMT17

receives all the rankers as input and performs a wide selection among rankers with high and low effectiveness, which is a very challenging scenario. In contrast, none of the baselines are required to perform any selection and the features are chosen manually. The selection stage is an important aspect and contribution of HRSF that is hard to replicate in the baselines. Notice that our approach achieved competitive or superior results for all the evaluated datasets.

With the objective of facilitating the visualization of the best results in the state-of-the-art, Table 7.5 presents the rank of the methods according to MAP and R1. The gray cells with bold values correspond to methods that have achieved a higher rank than HRSF. Our method achieved the best MAP on DukeMTMC and had the second position in the other two datasets. For R1, HRSF is positioned among the top-4 in all cases. The mean of the rank on each dataset is presented in the rightmost columns. Notice that our method achieved one of the highest rank means among all of the methods, being only slightly behind ISSDA [306]. The comparisons show that, besides our results being among the best for all evaluated datasets, in some cases, other non-fusion-based methods provided higher values than our approach. There are some possible explanations for this:

- Each dataset has different aspects (e.g., image resolution, picture angles, environment, number of images per person, dataset size). For this reason, different methods may perform better or worse on distinct datasets;
- The Baseline by [153] performs a multi-source training. The results reported by [153] are based on the transfer learning of a training performed on 7 Re-ID datasets. It is considered, by far, the largest labeled source of all the baselines, which leads to high results, especially on Market where it is ranked as the 2nd/3rd best R1/MAP (shown in Table 7.5);
- An idea that is exploited by some baselines is the generation of pseudo-labels. The most promising example is ISSDA [306], which has the best results on the Market and is well-ranked on DukeMTMC. Different from the others, ISSDA employs a self-supervised iterative pseudo-label generation and training. However, besides the effectiveness, the authors [306] claim that the training stage is very time-consuming since it requires, among other aspects, the execution of a clustering algorithm. Furthermore, ISSDA has an average ranking of 1.5 against 1.67 of our method (Table 7.5, MAP measure); However, ISSDA does not report results on CUHK03. Our average ranking without considering CUHK03 is also 1.5;
- The MAR [397] performs soft label generation. The applied strategy is capable of achieving promising results for improving the DukeMTMC dataset with the R1 measure. However, apparently, the quality of the soft labels varies according to the dataset.

Table 7.4 – State-of-the-art comparison considering MAP (%) and R-01 (%).

	Datasets					
	Market1501		**DukeMTMC**		**CUHK03**	
	R1	**MAP**	**R1**	**MAP**	**R1**	**MAP**
Unsupervised Domain Adaptation Methods						
ARN [181]	70.3	39.4	60.2	33.4	—	—
EANet [118]	66.4	40.6	45.0	26.4	**51.4**	31.7
ECN [431]	75.1	43.0	63.3	40.4	—	—
MAR [397]	67.7	40.0	**87.1**	48.0	—	—
TAUDL [170]	63.7	41.2	61.7	43.5	**44.7**	31.2
UTAL [171]	69.2	46.2	62.3	44.6	**56.3**	**42.3**
HHL (D,M) [430]	62.2	31.4	46.9	27.2	—	—
HHL (C03) [430]	56.8	29.8	42.7	23.4	—	—
ATNet (D,M) [197]	55.7	25.6	45.1	24.9	—	—
CSGLP (D,M) [273]	63.7	33.9	56.1	36.0	—	—
ISSDA (D,M) [306]	**81.3**	**63.1**	72.8	54.1	—	—
EANet (C03) [118]	59.4	33.3	39.3	22.0	—	—
EANet (D,M) [118]	61.7	32.9	51.4	31.7	—	—
SPGAN (D,M) [71]	43.1	17.0	33.1	16.7	—	—
DAAM (D,M) [121]	42.3	17.5	29.3	14.5	—	—
AF3 (D,M) [195]	67.2	36.3	56.8	37.4	—	—
AF3 (MT) [195]	68.0	37.7	66.3	46.2	—	—
PAUL (MT) [380]	68.5	40.1	72.0	53.2	—	—
EMTL (C02+D+M) [370]	52.8	25.1	39.7	22.3	—	—
CAMEL [396]	54.5	26.3	—	—	31.9	—
Baseline by [153]	80.5	56.8	67.4	46.9	29.4	27.4
Unsupervised Selection and Fusion (ours)						
HRSF ($\mathfrak{X}_2^*$)	74.32	60.89	76.80	68.51	**39.04**	**39.69**
HRSF ($\mathfrak{X}_3^*$)	75.56	62.64	**77.24**	**68.88**	39.13	39.58
HRSF ($\mathfrak{X}_4^*$)	**75.71**	**62.94**	76.89	68.56	38.02	38.80
HRSF ($\mathfrak{X}_5^*$)	74.00	60.69	76.39	67.72	36.15	37.11
HRSF ($\mathfrak{X}_6^*$)	73.57	59.85	75.90	66.96	35.46	36.17
HRSF ($\mathfrak{X}^*$, best result)	**75.71**	**62.94**	**77.24**	**68.88**	**39.04**	**39.69**

Table 7.5 – State-of-the-art methods ranked by their results.

	Market1501		**DukeMTMC**		**CUHK03**		**Mean**	
	R1	**MAP**	**R1**	**MAP**	**R1**	**MAP**	**R1**	**MAP**
Unsupervised Domain Adaptation Methods								
ARN [181]	5	10	10	12	—	—	7.5	11
EANet [118]	11	7	16	15	**2**	3	9.67	8.34
ECN [431]	4	5	7	9	—	—	5.5	7
MAR [397]	9	9	**1**	4	—	—	5	6.5
TAUDL [170]	12	6	9	8	**3**	4	8	6
UTAL [171]	6	4	8	7	**1**	**1**	5	4
HHL (D,M) [430]	14	16	14	14	—	—	14	15
HHL (C03) [430]	17	17	17	17	—	—	17	17
ATNet (D,M) [197]	18	19	15	16	—	—	16.5	17.5
CSGLP (D,M) [273]	13	13	12	11	—	—	12.5	12
ISSDA (D,M) [306]	**1**	**1**	3	2	—	—	**2**	**1.5**
EANet (C03) [118]	16	14	19	19	—	—	11.67	16.5
EANet (D,M) [118]	15	15	13	13	—	—	14	14
SPGAN (D,M) [71]	21	22	20	20	—	—	20.5	21
DAAM (D,M) [121]	22	21	21	21	—	—	21.5	21
AF3 (D,M) [195]	10	12	11	10	—	—	10.5	11
AF3 (MT) [195]	8	11	6	6	—	—	7	8.5
PAUL (MT) [380]	7	8	4	3	—	—	5.5	5.5
EMTL (C02+D+M) [370]	20	20	18	18	—	—	19	19
CAMEL [396]	19	18	—	—	5	—	12	18
Baseline by [153]	**2**	3	5	5	6	5	4.34	4.34
Unsupervised Selection and Fusion (ours)								
HRSF ($\mathfrak{X}^*$, best result)	**3**	**2**	**2**	**1**	**4**	**2**	**3**	**1.67**

7.2.4 Visual Results

Some qualitative results were also elaborated to evince the quality of our obtained results. Two different queries (o_1, o_2) for the same person (ID) were chosen from the DukeMTMC dataset. Figure 7.9 presents a graph for each ranker that composes the best combination ($\mathfrak{X}^*$) on DukeMTMC dataset and the HRSF result. Each dot represents a gallery image, which is positioned in the graph according to its distance to the query images (o_1, o_2). This is a challenging example because the individual is presented from different angles (back and side) in each image, with the umbrella causing some occlusion. Additionally, certain items visible in image o_1, such as the purse and shoes, are not visible in image o_2. The idea is that, since the query images are of the same person, the distance between them should be small and the images of the same ID should be closer to the bottom left corner. Images obtained from different camera views are presented in different symbols. It shows that the HRSF method was capable of reducing the distance of all the images belonging to the same class when compared to isolated rankers (OSNET, OSNET-IBN, OSNET-AIN).

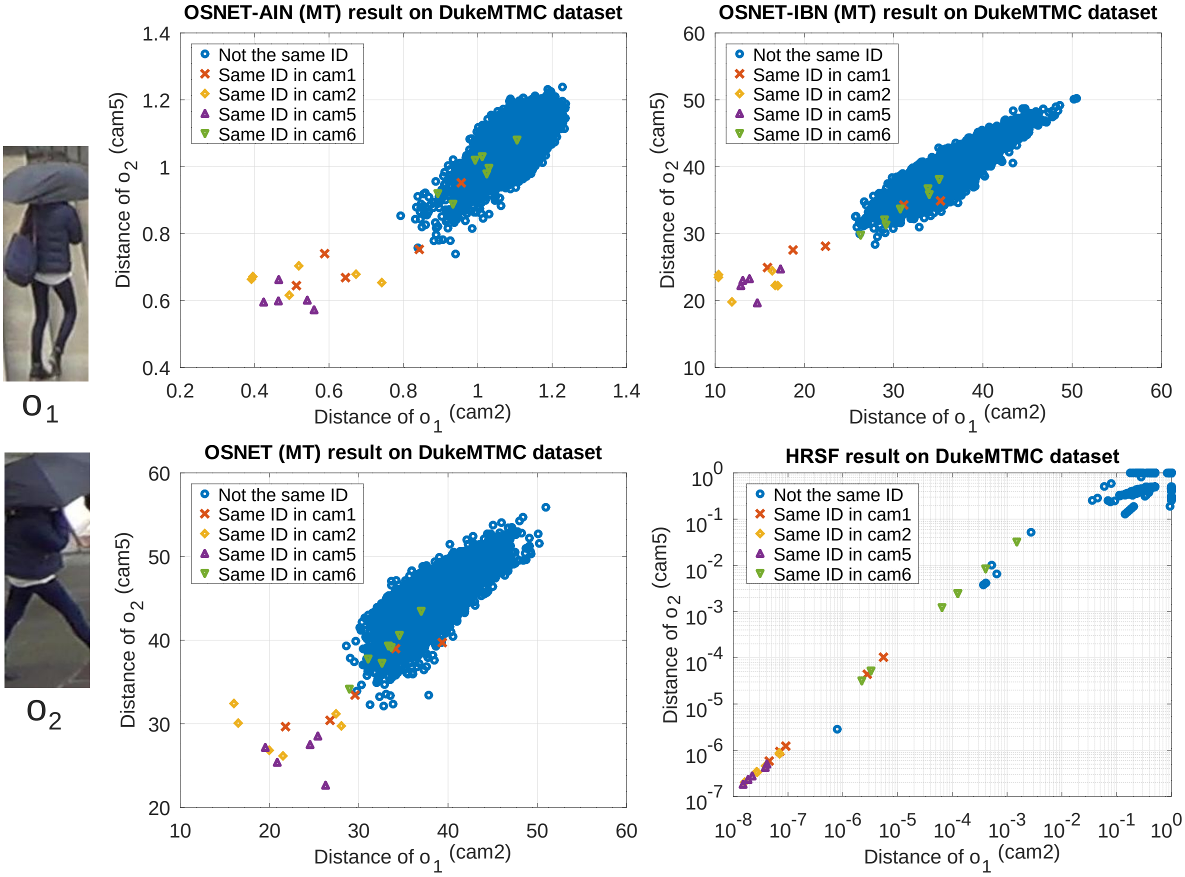
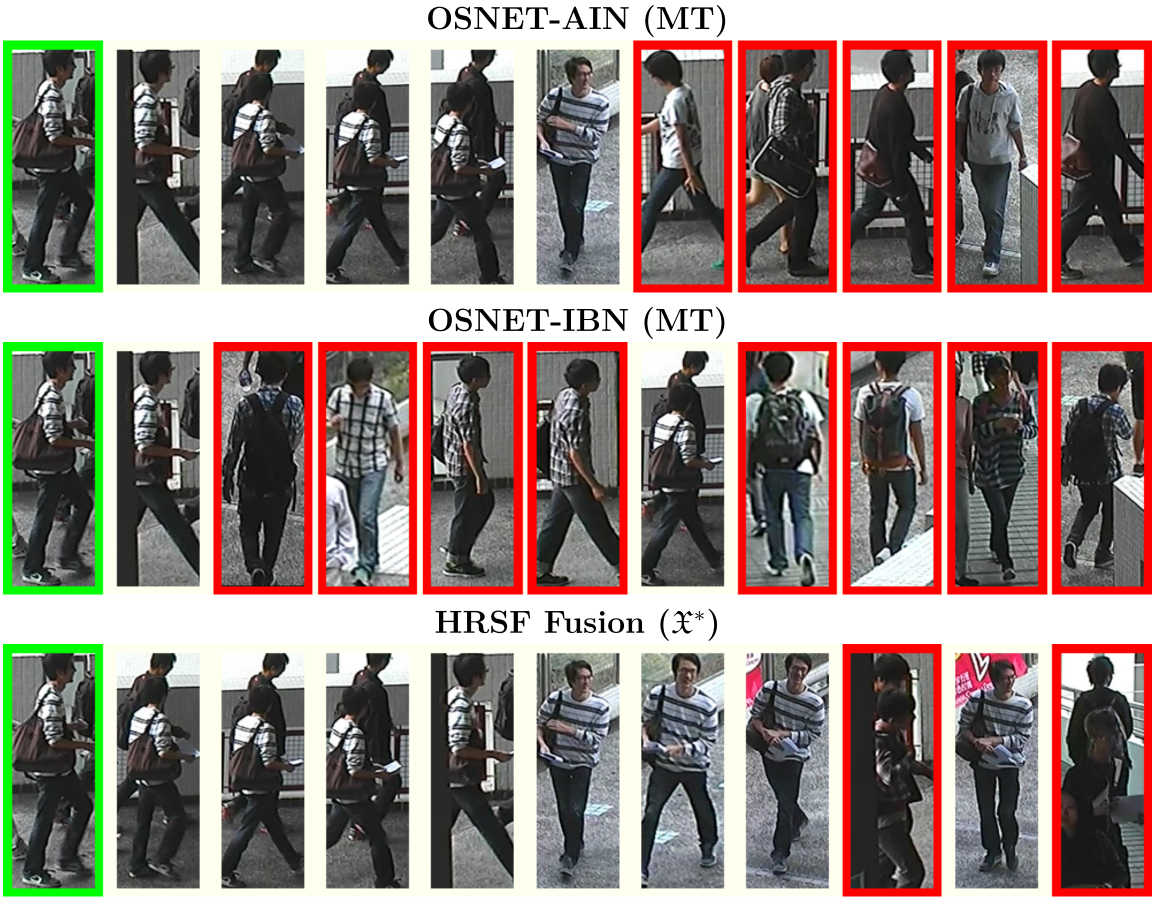


Figure 7.9 – Distance distribution for two query images on DukeMTMC dataset.

Figures 7.10 and 7.11 present examples of visual queries on the CUHK03 and DukeMTMC datasets, respectively. These are also challenging examples, featuring people at distinct angles, wearing different clothes, and with some occlusions, such as the orange car in the DukeMTMC example. The results are shown for the best combination obtained by HRSF ($\mathfrak{X}^*$) and the rankers that compose it. The query image is presented with green borders and the wrong results with red borders. Notice that, in these cases, beyond selecting the best results, our approach was also capable of removing most of the incorrectly retrieved images.



OSNET-AIN (MT)

OSNET-IBN (MT)

HRSF Fusion ($\mathfrak{X}^*$)

Figure 7.10 – Examples to illustrate the impact of HRSF selection and fusion on the CUHK03 dataset.

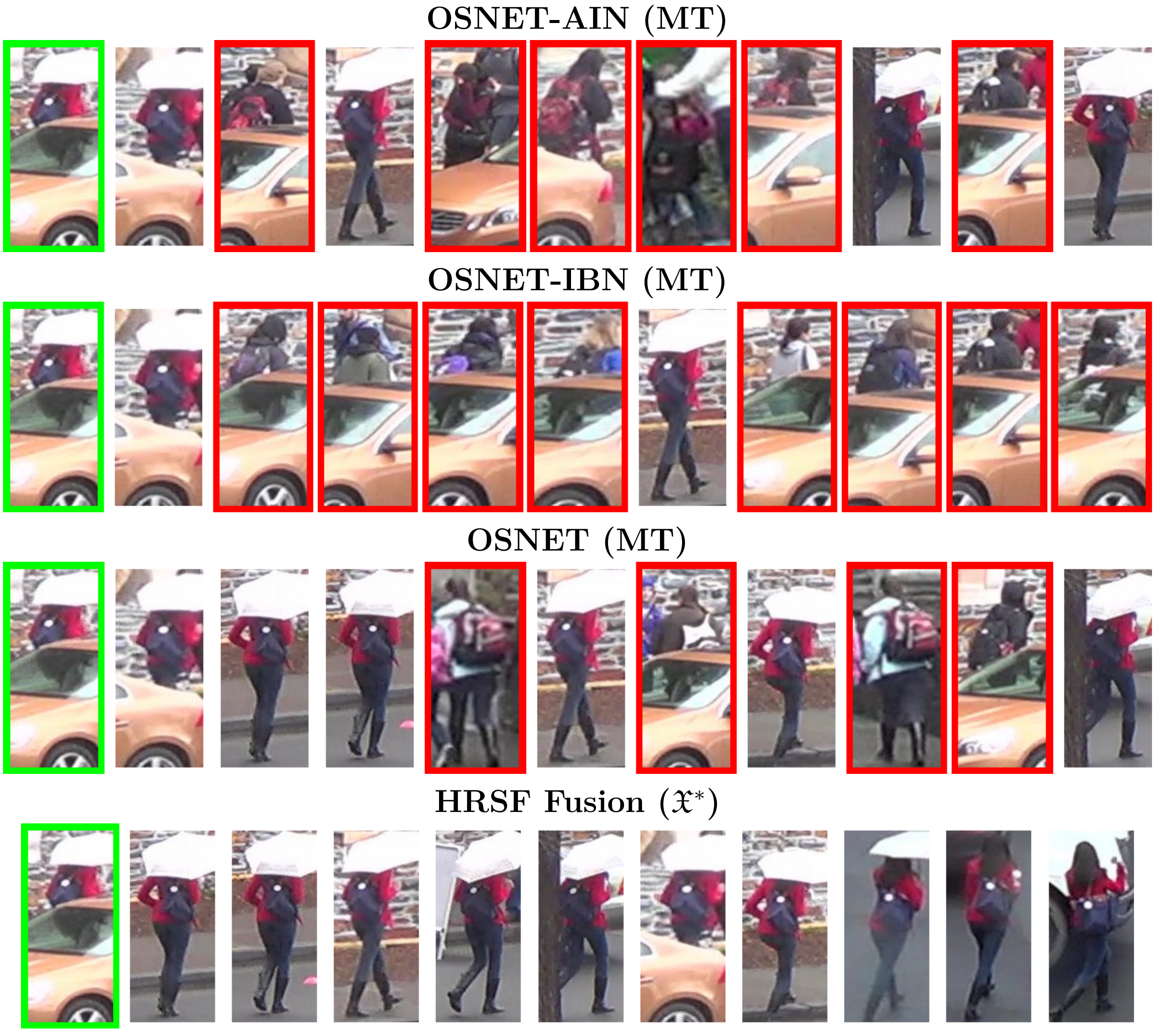


Figure 7.11 – Examples to illustrate the impact of HRSF selection and fusion on the DukeMTMC dataset.

8 Rank Flow Embedding (RFE)

Unsupervised image retrieval and semi-supervised classification are well-established and extensively researched tasks. They present significant challenges and are interconnected, with a wide range of applications in fields such as person re-identification [137], medical imaging [1], and remote sensing [355], among others. In such tasks, how images are represented and the measures used to compare them are crucial aspects [321, 446, 338].

Recently, consistent progress has been achieved in representation strategies, especially due to the evolution of deep learning with Convolutional Neural Networks (CNN) and Vision Transformers (ViT) models [321]. However, pairwise similarity measurements are still widely employed which is a major limitation, particularly for being insufficient to reveal the intrinsic relationship between images in high-dimensional spaces [123]. A promising solution is to estimate similarities more accurately by considering the underlying data manifold [18]. Strategies in this direction are based on the idea of exploiting the context of other objects, which can be performed using different structures (e.g., graphs, ranked lists, and others). This is also closer to human behavior in judging the similarity among objects. Although there have been many advancements, most strategies are tailored to solve a single specific problem, limiting their ability to generalize across different tasks.

In this chapter, our contribution is an unsupervised rank-based approach capable of refining similarity information and computing a context-sensitive representation, which can be exploited to improve the effectiveness of both unsupervised retrieval and semi-supervised classification. We propose a novel manifold learning algorithm named Rank Flow Embedding (RFE) [334]. The proposed method is based on different and complementary ideas recently exploited by manifold learning approaches in order to provide a better contextual representation of dataset objects. The algorithm computes rank-based embeddings which are refined along the processing flow for each step. This approach constitutes a key innovation in the sense that constitutes an unsupervised contextual-sensitive method capable of computing a novel representation and not only a similarity measure.

Firstly, a rank-based formulation is used to define a hypergraph model capable of representing high-order similarity information encoded in ranked lists. The hypergraph is used for iterative re-ranking, based on the similarity among embeddings defined by hyperedges (*h-embeddings*). Next, Cartesian product operations are performed on hyperedges to maximize their similarity relationships. While hyperedges effectively represent regional relationships, broader similarity relationships are also relevant. In this direction, hypergraph structures are also used to model a graph and define high-confident

Connected Components (CCs), aiming at estimating class information of datasets. The information encoded in the CCs is exploited for a new re-ranking step and used as class representatives to compute low-dimensional embeddings. Such embeddings, in turn, can be exploited for more effective semi-supervised classification tasks.

The proposed method presents various contributions and innovations regarding related work. Among them:

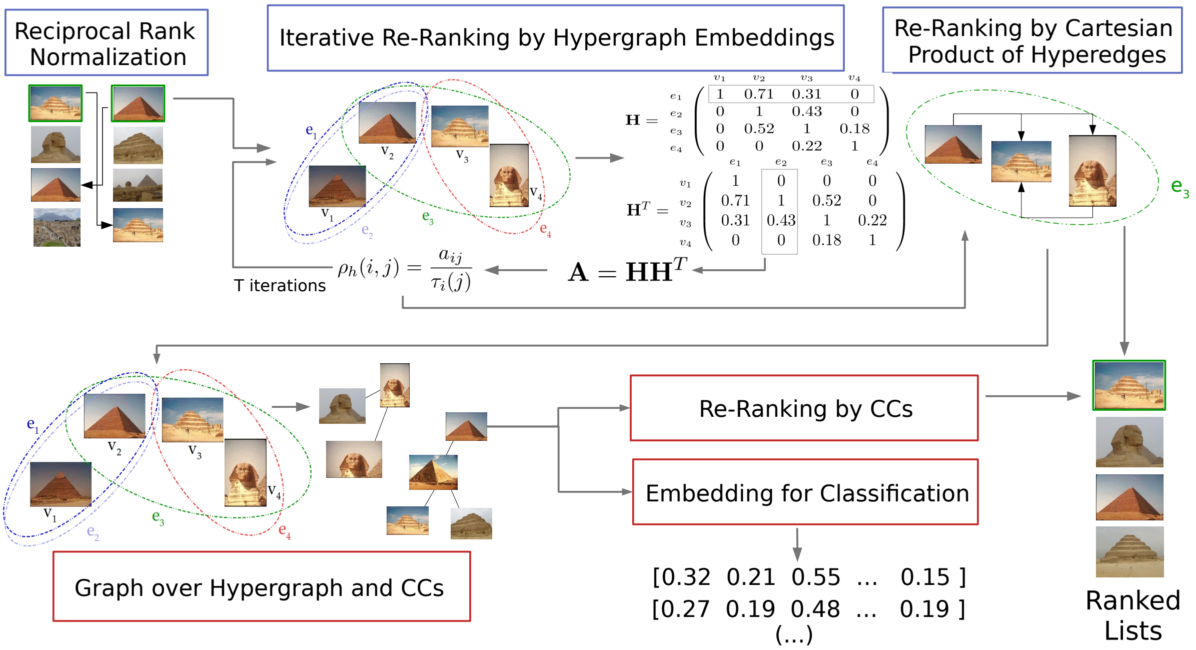
- Most unsupervised context-sensitive approaches establish a novel similarity measure [282, 253, 18], but not a novel representation. Beyond that, RFE proposes a novel rank-based approach for learning context-sensitive representations. More effective representations are fundamental for many applications, including unsupervised retrieval and semi-supervised classification, scenarios in which the method was evaluated;
- The proposed approach presents substantial innovations in the way of computing such representations. The embeddings and their encoded similarity information are refined through a flow of rank-based structures and operations. Although some strategies already have been individually exploited (graphs [351], hypergraphs [18], and connected components [249]), our work allows the sequential refinement of similarity information along these structures. In addition, the proposed approach includes relevant distinctions in how such structures are defined and used. More specifically: *(i)* The hypergraph model used is defined based on a novel rank normalization function, proposed in this work and named as reciprocal sigmoid; *(ii)* The computation of connected components is based on a ranking of candidate edges, which estimates the confidence of edges using the hypergraph embeddings. The strategy consists of a novel approach proposed in this work; *(iii)* The use of similarity to the connected components for defining the dimensions of novel representations is also an innovation proposed in this chapter.
- The method can be used in scenarios where the queries are not part of the dataset (*unseen queries*), which is fundamental for many real-world applications and has been little exploited by related work in post-processing methods.

The effectiveness of the proposed method was confirmed with a wide and diversified experimental evaluation. The experimental results were obtained on 10 public datasets, including traditional image retrieval benchmarks and person Re-ID datasets. For each dataset, different features were considered including CNN and recent Vision Transformers features. For semi-supervised classification, the evaluation considered the proposed RFE embedding classified by different Graph Convolutional Network (GCN) models. An ablation study was also conducted in order to assess the impact of each step of the proposed method. The experimental evaluation also considers comparisons with other state-of-the-art approaches on various datasets. The results demonstrate the effectiveness of the proposed method on different tasks: unsupervised image retrieval, semi-supervised classification, and person Re-ID.

This chapter is organized as follows: Section 8.1 presents the proposed RFE method. Section 8.2 describes the experimental evaluation.

8.1 Proposed Method

How to effectively design context-aware measures is a challenging question, which is closely associated with how to represent each image in terms of the collection in which is contained. Analogous to convolution and pooling operations used on CNNs, the proposed *Rank Flow Embedding* (RFE) employs subsequent rank-based operations to define more effective contextual representations. Representations are derived from similarity to other images modeled by rank information. Such representations, in turn, are used to derive more effective similarity measures. Such a mechanism is repeated through a flow of distinct and complementary operations to extract the maximum of available contextual information.


Reciprocal Rank Normalization
Iterative Re-Ranking by Hypergraph Embeddings
Re-Ranking by Cartesian Product of Hyperedges
e1
e2
e3
e4
v1
v2
v3
v4
$\mathbf{H} =$
$\mathbf{H}^T =$
$\mathbf{A} = \mathbf{H}\mathbf{H}^T$
$\rho_h(i,j) = \frac{a_{ij}}{\tau_i(j)}$
T iterations
Graph over Hypergraph and CCs
Re-Ranking by CCs
Embedding for Classification
[0.32 0.21 0.55 ... 0.15]
[0.27 0.19 0.48 ... 0.19]
(...)
Ranked Lists

$$\mathbf{H} = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline e_1 & 1 & 0.71 & 0.31 & 0 \\ e_2 & 0 & 1 & 0.43 & 0 \\ e_3 & 0 & 0.52 & 1 & 0.18 \\ e_4 & 0 & 0 & 0.22 & 1 \end{array}$$

$$\mathbf{H}^T = \begin{array}{c|cccc} & e_1 & e_2 & e_3 & e_4 \\ \hline v_1 & 1 & 0 & 0 & 0 \\ v_2 & 0.71 & 1 & 0.52 & 0 \\ v_3 & 0.31 & 0.43 & 1 & 0.22 \\ v_4 & 0 & 0 & 0.18 & 1 \end{array}$$

Figure 8.1 – Overall organization of Rank Flow Embedding: in blue boxes the initial steps and in red boxes optional steps for refining retrieval and for computing embedding for semi-supervised classification.

Figure 8.1 presents the main steps of the proposed approach and the respective workflow. The proposed manifold learning algorithm can be used for unsupervised re-ranking, producing ranked lists as output retrieval results, or for representation learning, producing contextual vector representations. The method can be summarized by the following steps:

1. **Ranked Lists Normalization**: Ranked lists are recomputed considering a sigmoid score computed based on the reciprocal positions in the ranked lists.
2. **Re-ranking by Hypergraph Embeddings**: An iterative step that employs a hypergraph structure to analyze the underlying similarity information contained in the ranked lists. This step defines the h-embeddings and hyperedge weights, which are used in the next steps.
3. **Re-Ranking by Cartesian Product**: A Cartesian product step is used to spread the similarity information among elements in the same hyperedge.
4. **Re-ranking by Connected Components**: High-confident connected components (CCs) are defined based on hypergraph structures (Step 2). The CCs are computed based on the most confidential edges identified through the hyperedge weights. The CCs encode class information and cause objects in the same CC to have their similarities increased.
5. **Embeddings by Connected Components**: More effective embeddings are computed for each dataset element considering their similarity to the identified CCs. This step is directed for semi-supervised classification since a low-dimensional embedding is obtained.

Each stage is detailed and formally defined in the next sections. In general, each step incrementally improves the effectiveness of rank-based similarity information and computes structures that are exploited in the next steps. While Steps 1-3 are suitable for general retrieval tasks, Step 4 is focused on datasets with larger similarity groups, in which information from CCs can be better exploited. Hence Step 4 is not suitable for datasets with large numbers of very small classes. Step 5 uses the constructed structures for computing embeddings used for classification. Besides the standard retrieval pipeline and the embeddings for classification, rank aggregation tasks and the use of unseen queries are also discussed.

8.1.1 Formal Definition for Rank-based Manifold and Representation Learning

The proposed RFE method aims to capture the structure of the dataset manifold by exploiting the similarity information encoded in the set of ranked lists $\mathcal{T}$. As a result, the RFE is evaluated on two objectives: (i) computing a more effective similarity measure and ranking result for unsupervised retrieval and; (ii) computing a more effective embedding to represent each image, which can be used by other tasks, as semi-supervised classification.

Regarding unsupervised manifold learning, a new and more effective set of ranked $\mathcal{T}_r$ is computed with the aim of improving the effectiveness of ranking results. More formally, we can describe the method as function f_m:

$$\mathcal{T}_r = f_m(\mathcal{T}) \tag{8.1}$$

The aggregation problem is also considered, in which different sets of ranked lists $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_d\}$ are taken as input aiming at computing a more effective set $\mathcal{T}_r$.

Regarding representation learning, the objective is to compute an embedding that provides a more effective representation for a given object o_i based on the contextual similarity information encoded in $\mathcal{T}$. Formally, it can be defined as function f_e:

$$\mathbf{e}_i = f_e(\mathcal{T}, o_i), \tag{8.2}$$

where $\mathbf{e}_i$ is a vector on a d_e-dimensional embedding space.

8.1.2 Rank Normalization by Reciprocal Sigmoid

In opposite to the majority of distance measures, the ranking information is not symmetric. The increase of symmetry generally produces a positive impact on the effectiveness of similarity information, widely exploited by reciprocal rank analysis [241, 267]. However, most of the reciprocal approaches apply linear analysis to rank positions. In this method, we use a non-linear scoring function that assigns high weights to top-rank positions, with a fast decay around the neighborhood size, given by k. With this objective, a sigmoid function is applied. Additionally, a higher relevance is assigned to the original rank position (squared) in comparison with the reciprocal rank position (linear). The new similarity between objects o_i and o_j is defined by ρ_n:

$$\rho_n(i, j) = \sigma(i, j)^2 \times \sigma(j, i). \tag{8.3}$$

The function σ which assigns weights according to rank positions is defined as:

$$\sigma(x, y) = 1 - \frac{1}{1 + e^{-\alpha(\tau_x(y) - k/2))}}, \tag{8.4}$$

where α is a constant empirically evaluated in the experimental analysis. The parameter α impacts the sigmoid-based function σ which assigns weights according to the positions of images in ranked lists. Low values of α define a slower decay, while high values indicate a fast decay.

Based on the measure ρ_n, which is computed between the objects in the top-L positions, the ranked lists are updated with a stable sorting algorithm. Stable sorting is used in order to keep the position in the case of a tie. An updated set of ranked lists $\mathcal{T}_n$ is obtained as output.

8.1.3 Re-Ranking by Hypergraph Embeddings

The contextual representation model used for data elements and how to exploit it to compute more effective similarity measures is a fundamental task in rank-based manifold learning. In this work, we use a hypergraph model based on ranking information inspired by [251, 120]. The hypergraph establishes relations among a set of objects, allowing the representation of high-order similarity relationships. The proposed RFE method computes contextual embeddings based on hypergraph information and defines an iterative re-ranking procedure based on the comparison of such embeddings.

- **Hypergraph Embeddings**

Formally, a hypergraph model is defined by a tuple $H = (V, E_h, w)$, where V represents a finite set of vertices and E_h denotes the set of hyperedges. The hyperedges set E_h can be defined as the family of subsets of V such that $\bigcup_{e_i \in E_h} = V$. A hyperedge e_i is said to be incident to a vertex v_j if $v_j \in e_i$. For each hyperedge e_i, a positive weight $w(e_i)$ is assigned, which denotes the confidence of the relationships established by the hyperedge e_i. In this chapter, please note that in addition to adopting the hypergraph definition described in Section 2.6, some symbols have been modified to avoid misunderstandings that may arise in the RFE formulation. For example, to denote the hypergraph model, H is used instead of $\mathbf{H_G}$.

Each vertex $v_i \in V$ represents an object in the collection: $o_i \in \mathcal{C}$. For each object, a hyperedge is created by exploiting first and second-order neighborhood information. As a reminder, the neighborhood set of an object o_q, denoted as $\mathcal{N}(q, k)$, is defined in Section 2.2.3. A hyperedge e_i is defined based on the neighborhood set of o_i and its respective neighbors. Formally, let $o_x \in \mathcal{N}(i, k)$ be a neighbor of o_i and let $o_j \in \mathcal{N}(x, k)$ be a neighbor of o_x, the hyperedge e_i is defined as:

$$e_i = \mathcal{N}(i, k) \bigcup_{o_x \in \mathcal{N}(i,k)} \mathcal{N}(x, k). \tag{8.5}$$

Consequently, each image o_i is now also represented by a hyperedge e_i. Since the number of hyperedges is equal to the number of vertices, the obtained hypergraph can be represented by a square incidence matrix $\mathbf{H}_m$ of size $|E_h| \times |V|$, where elements $\mathbf{H}_m$ are defined as:

$$h_m(e_i, v_j) = \begin{cases} r(e_i, v_j), & \text{if } v_j \in e_i, \\ 0, & \text{otherwise.} \end{cases} \tag{8.6}$$

Row i of h_m tells which vertices belong to hyperedge e_i and the score $r(e_i, v_j)$ indicates the degree of belonging of the vertex v_j to hyperedge e_i. The score r is computed according to the number and relevance of mentions to v_j in the hyperedge e_i and is defined as:

$$r(e_i, v_j) = \sum_{o_x \in \mathcal{N}(i,k) \wedge o_j \in \mathcal{N}(x,k)} w_p(i, x) \times w_p(x, j), \tag{8.7}$$

where $w_p(i, x)$ is a function that assigns a weight of relevance to o_x according to the position in the ranked list τ_i. Notice that the score r incorporates information from first and second-order ranking references, i.e., from neighbors and neighbors of neighbors. The weight assigned to o_x according to the position of the ranked list τ_i is defined by a log-based function as:

$$w_p(i, x) = 1 - \log_k \tau_i(x). \tag{8.8}$$

The function $w_p(i, x)$ reaches the maximum value of 1, which is assigned to the first position of the ranked lists and corresponds to the query image. For the subsequent positions in the ranked lists, the function decays fast.

While the hyperedge e_i provides a more comprehensive contextual representation for the object o_i, it can also be susceptible to noise in certain circumstances. As it considers second-order similarity relationships, non-relevant objects in rankings of neighbors can generate undesired references in the hyperedge e_i. With the aim of filtering out such cases, we include a consistency check among hyperedges to obtain a more precise representation.

The main idea consists of verifying for each element in the hyperedge e_i how it is referenced by other hyperedges. Most objects in e_i are expected to be relevant and compose a consistent set of high-similarity among each other. Thus, a given relevant object $o_j \in e_i$ is expected to be referenced with high scores in the other hyperedges which represent most of the elements in e_i. On the other hand, a noisy and non-relevant object $o_n \in e_i$ is not expected to be referenced in the same hyperedges.

In this way, the filtered score for a given object $o_j \in e_i$ is computed by multiplying scores in e_i by the score of o_j in hyperedges of elements referenced in e_i, which can be obtained by a matrix $\mathbf{H}$ computed as

$$\mathbf{H} = \mathbf{H}_m{}^2. \tag{8.9}$$

The computation of matrix $\mathbf{H}$ defines the embeddings provided by the hypergraph model to represent each object, which we denote as *h-embeddings*. For an object o_i, its respective h-embedding can be defined by the correspondent row of matrix $\mathbf{H}$, such that:

$$\mathbf{h}_i = [h_{i1}, h_{i2}, \ldots, h_{in}], \tag{8.10}$$

where h_{ij} defines the similarity of object o_j in the hyperedge e_i, also denoted as $h(i, j)$.

The definition of the hypergraph also includes the confidence of each hyperedge, given by the function $w(e_i)$. A highly effective hyperedge is expected to contain a consistent set of vertices. Therefore, it is expected to contain only a few vertices with high score values given by $h(e_i, \cdot)$. Hence, the weight $w(e_i)$ is defined as:

$$w(e_i) = \sum_{j \in \mathcal{N}_h(i,k)} h(i, j), \tag{8.11}$$

where $\mathcal{N}_h(i,k)$ is a neighborhood set defined among the elements with top $h(e_i,\cdot)$ score values in the hyperedge. The $\mathcal{N}_h$ set containing the vertices with the highest values of $h(e_i,\cdot)$ is formally defined as:

$$\mathcal{N}_h(q,k) = \{\mathcal{S} \subseteq e_q, |\mathcal{S}| = k \wedge \forall o_i \in \mathcal{S}, o_j \in e_q - \mathcal{S} : h(q,i) > h(q,j)\}. \tag{8.12}$$

Based on the previous equations, we can define a function $f_h(\cdot)$ that, given a set of ranked lists $\mathcal{T}_n$ as input, computes a hypergraph H and its respective *h-embeddings* given by the matrix $\mathbf{H}$. The function is defined as follows:

$$(H, \mathbf{H}) = f_h(\mathcal{T}_n). \tag{8.13}$$

In fact, the matrix $\mathbf{H}$ and the weight of edges $w(.)$ contain the main similarity information encoded in the hypergraph model. Both structures are exploited by the proposed RFE method and used in this work. Firstly, the information encoded in matrix $\mathbf{H}$ is exploited to define a contextual similarity measure used for re-ranking. Additionally, all the weights $w(e_i)$ are scaled using min-max normalization to keep all values within the [0, 1] interval.

- **Hypergraph-based Re-Ranking**

While similar objects present similar ranked lists, it is expected that the respective h-embeddings are also similar. Once the similarity information is encoded in the matrix $\mathbf{H}$, a similarity measure between two embeddings $\mathbf{h}_i$ and $\mathbf{h}_j$ can be computed by its product $\mathbf{h}_i\mathbf{h}_j$. This operation can be modeled for all the objects by multiplying the matrix $\mathbf{H}$ by its transpose, with the objective of obtaining the affinity matrix $\mathbf{A}$, defined as follows:

$$\mathbf{A} = \mathbf{H}\mathbf{H}^T. \tag{8.14}$$

The elements of matrix $\mathbf{A}$ given by a_{ij} denote the similarity between objects o_i, o_j. The matrix $\mathbf{A}$ contains most of the similarity information extracted based on the hypergraph, such that it can be used to define a more effective similarity measure ρ_h. In addition, the proposed measure also considers residual similarity information, given by the original ranking position. The measure is defined as:

$$\rho_h(i,j) = \frac{a_{ij}}{\tau_i(j)}. \tag{8.15}$$

Based on the similarity computed by the function ρ_h, an updated set of ranked lists $\mathcal{T}_h^{(t)}$ is obtained by applying a stable sorting algorithm. The ranked lists, in turn, can be used to compute a novel hypergraph, and the procedure can be iteratively repeated, such that the superscript $^{(t)}$ denotes the iteration.

After a certain number of T iterations, the set of ranked lists $\mathcal{T}_h^{(T)}$ is provided to the function f_h, which returns a matrix $\mathbf{H}_a$ and an updated hypergraph H_a, used in next steps of the rank flow. The index a is used to indicate that they were obtained based on the affinity matrix:

$$(H_a, \mathbf{H}_a) = f_h(\mathcal{T}_h{}^{(T)}). \tag{8.16}$$

8.1.4 Re-Ranking by Cartesian Product

A Cartesian product step is used to expand the similarity information contained in the updated set of hyperedges E_h^a. Inspired by [251, 332], the procedure exploits high-order similarity relationships represented on hyperedges to compute more effective pairwise measures. Formally, given two hyperedges $e_q, e_i \in E_h^a$, the Cartesian product between them can be defined as:

$$e_q \times e_i = \{(v_x, v_y) : \ v_x \in e_q \wedge v_y \in e_i\}. \tag{8.17}$$

The notation ${e_q}^2$ is used aiming to indicate the Cartesian product between elements of the same hyperedge e_q, such that $e_q \times e_q = {e_q}^2$. For each pair of vertices $(v_i, v_j) \in {e_q}^2$ a pairwise relationship $p : E_h^a \times V \times V \rightarrow \mathbb{R}^+$ is established.

A value p is computed based on the weight $w(e_q)$, which indicates the level of confidence of the hyperedge that originated the association. As previously mentioned, the weight $w(e_i)$ can be interpreted as the confidence estimations of associations encoded on hyperedge e_i. The degrees of association of v_i and v_j are defined by:

$$p(e_q, v_i, v_j) = w(e_q) \times h(e_q, v_i) \times h(e_q, v_j). \tag{8.18}$$

A pairwise similarity measure based on the Cartesian product is defined considering relationships contained in all the hyperedges. This formulation presents the idea of exploiting the co-occurrence of v_i and v_j in different hyperedges, performing a sum of all the values of $p(\cdot, v_i, v_j)$:

$$\rho_c(i,j) = \sum_{e_q \in E \wedge (v_i, v_j) \in {e_q}^2} p(e_q, v_i, v_j). \tag{8.19}$$

Based on the similarity function ρ_c, a more effective set of ranked lists $\mathcal{T}_c$ is computed by a stable sorting algorithm. The ranked lists set $\mathcal{T}_c$ is provided to the function f_h that computes an updated hypergraph and h-embeddings. The index c is used to indicate that they were obtained after the Cartesian product step:

$$(H_c, \mathbf{H}_c) = f_h(\mathcal{T}_c). \tag{8.20}$$

8.1.5 Graph over Hypergraph and Connected Components

Although the hypergraph model provides an effective tool to represent regional similarity information, it does not represent the similarity among objects in the same class or cluster that are more distant on the dataset manifold. In order to represent such information, a high-confident graph is defined based on h-embeddings computed after Cartesian product operations. The Connect Components are extracted from this graph and are used to represent class information and the global structure of similarity relationships encoded in the dataset.

- **Graph Definition**

Formally, the graph is defined as $G = (V, E)$, such that the set of vertices $V = \mathcal{C}$, where each node represents a collection object. The set of edges E is computed based on information provided by the hypergraph representation. Firstly, a set of candidate edges $\mathcal{E}_c$ is defined based on the neighborhood set of each object as:

$$\mathcal{E}_c = \bigcup_{q \in V} \bigcup_{i \in \mathcal{N}(q,k)} \{(q,i)\}. \tag{8.21}$$

In order to select the most confident edges, the set of candidates is ranked. The ranked list τ_c is defined as a permutation of the set of candidate edges $\mathcal{E}_c$. The permutation τ_c is the bijection of the set $\mathcal{E}_c$ onto the set $[n_k] = \{1, 2, \ldots, n_k\}$, The position of the pair (q, i) in the ranked list is denoted by $\tau_c((q, i))$. The permutation is defined such that if (q, i) is ranked before (j, l), e.g, $\tau_c((q, i)) < \tau_c((j, l))$, then $s_c(q, i) \geq s_c(j, l)$. The function s_c is a similarity measure attributed to pairs based on the similarity between h-embeddings and confidence of the hyperedge, defined as:

$$s_c(i, j) = \mathbf{h}_{c_i} \mathbf{h}_{c_j}^T \times w(e_i) \times w(e_j), \tag{8.22}$$

where the pair (i, j) identifies a pair of hyperedges $e_i, e_j \in E_h^c$, and E_h^c denotes a set of hyperedges of the hypergraph H_c. Once ranked, a threshold should be established to define the number of edges that are created. The threshold t_c is defined as:

$$t_c = \frac{\sum_{e_q \in E_h^c} w(e_q)}{2 \times n}. \tag{8.23}$$

Since the weights $w(\cdot)$ are within the range $[0, 1]$ due to normalization, t_c lies within the interval $[0, 0.5]$ due to the division by $2 \times n$. The threshold t_c is used to define the edge set E a follows:

$$E = \{(o_q, o_i) \mid (q, i) \in \mathcal{E}_c \land \tau_c((q, i)) < \text{round}(n_k \times t_c)\}, \tag{8.24}$$

where round$(n_k \times t_c)$ is a function that returns the nearest integer that corresponds to the number of edges that are selected as part of E. The threshold t_c can be understood as a percentage of the total number of candidate edges (n_k) selected.

The process of building the graph can be understood as a function f_g that receives as input a hypergraph H_c and a matrix $\mathbf{H}_c$ (output of the Cartesian product) and computes a graph G:

$$G = f_g(H_c, \mathbf{H}_c). \tag{8.25}$$

- **Connected Components**

Based on the defined graph, its respective Connected Components (CC) are extracted. Formally, each CC is defined as a set of objects $\mathcal{C}_i$. Given two objects o_i, $o_j \in \mathcal{C}_l$, there is a path (edge) between o_i, o_j. Search algorithms in graphs (e.g. Depth and Breadth-First) and the Tarjan algorithm can be used to compute the CCs. The output for the dataset is provided by the set of connected components $\mathcal{S} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$, such that $\bigcup_{\mathcal{C}_i \in \mathcal{S}} = \mathcal{S}$ and $\bigcap_{\mathcal{C}_i \in \mathcal{S}} = \emptyset$.

The connected components are sets of similar objects and it is expected that such structures encode the information of sets or classes of the dataset. Following this reasoning, an embedding is created based on the *h-embeddings* of the elements that are part of it. Given a connected component q, the cc-embedding $\mathbf{c}_q$ is defined as:

$$\mathbf{c}_q = \sum_{o_i \in \mathcal{C}_q} \mathbf{h}_{c_i}. \tag{8.26}$$

Once the Connected Components (CCs) encode information associated with the representation of classes, the similarity to such CC embeddings can be exploited for computing a more globally contextual similarity measure. In this way, a novel embedding is computed for each object according to its similarity to the CC embeddings. Formally, let $\mathbf{e}_q$ be an embedding of an object of index q. The computation of the value of position i of this vector (embedding) is done as follows:

$$\mathbf{e}_q[i] = \mathbf{h}_{c_q} \mathbf{c}_i^T, \tag{8.27}$$

where i identifies the connected component $\mathcal{C}_i \in \mathcal{S}$ and $\mathbf{c}_i$ denotes the embedding that corresponds to this CC. In this way, the embeddings can be computed for each element of the dataset.

- **Re-Ranking by Connected Components**

The re-ranking by CCs exploits information about elements in the same CC. In this way, the elements that present high similarity values in the same CC, have their similarities increased. The first step of this process consists of defining the k elements with the highest values in each connected component. A neighborhood set $\mathcal{N}_c(q,k)$ is defined for each element of index q considering a constant k:

$$\mathcal{N}_c(q,k) = \{\mathcal{S} \subseteq \mathcal{C}, |\mathcal{S}| = k \wedge \forall o_i \in \mathcal{S}, o_j \in \mathcal{C} - \mathcal{S} : \mathbf{c}_q[i] > \mathbf{c}_q[j]\}. \tag{8.28}$$

The ranked list τ_{c_q} can be defined as the permutation of objects that have the k highest values in the embedding $\mathbf{c}_q$. The permutation is defined as the bijection of the set $\mathcal{N}_c(q,k)$ to the set $[k] = \{1, 2, \ldots, k\}$. The position of an object o_i in the ranked list computed by the embedding of the connect component $\mathbf{c}_q$ is defined as $\tau_{c_q}(i)$. If o_i is ranked before o_j in a ranked list, this means, $\tau_{c_q}(q,i) < \tau_{c_q}(q,j)$, therefore $\mathbf{c}_q[i] \geq \mathbf{c}_q[j]$.

The re-ranking by CCs exploits three complementary information: (*i*) the similarity between embeddings; (*ii*) the object belonging to the same connected component and; (*iii*) the residual information of rank position. The similarity $\rho_e(i,j)$ is defined in order to combine such information, formally defined as:

$$\rho_e(i,j) = \sum_{o_i,o_j \in \mathcal{N}_c(q,k)} \frac{\left(1 \Big/ \left(1 + \sqrt{\tau_{c_q}(q,i)^2 + \tau_{c_q}(q,j)^2}\right)\right) \times \mathbf{e}_i \mathbf{e}_j^T}{\tau_i(j)}. \tag{8.29}$$

Based on the similarity function ρ_e, a set of ranked lists $\mathcal{T}_e$ is obtained by a stable sorting algorithm. The set of ranked lists $\mathcal{T}_e$ is provided to the function f_h that computes a new hypergraph H and matrix $\mathbf{H}$. The index e is used to indicate that they were obtained after the step of the connected components:

$$(H_e, \mathbf{H}_e) = f_h(\mathcal{T}_e). \tag{8.30}$$

8.1.6 Embeddings for Classification

The class information encoded in the re-ranking by CCs can be useful for other machine learning tasks. In this way, novel representations are computed for dataset objects and used as embeddings for semi-supervised classifiers. Given the ranked lists $\mathcal{T}_e$ and the hypergraph H_e obtained in the previous step, we obtain a graph with the updated connected components following the same equations defined in Section 8.1.5. Thus, the updated graph is defined as follows:

$$G_e = f_g(H_e, \mathbf{H}_e). \tag{8.31}$$

The new connected components, considering the component $\mathbf{c}$ after the step of CC (index e) for the element q, are obtained as follows:

$$\mathbf{c}_{e_q} = \sum_{o_i \in \mathcal{C}_{e_q}} \mathbf{h}_{e_i}. \tag{8.32}$$

Finally, each of the positions of the embedding vector, which is going to be used for classification, is computed as follows:

$$\mathbf{e}_{e_q}[i] = \mathbf{h}_{e_q} \mathbf{c}_{e_i}^T, \tag{8.33}$$

where the index e indicates that the variables were obtained after the re-ranking by the connected components. The contextual embedding $\mathbf{e}_{e_q}$ is used as features by semi-supervised classifiers.

8.1.7 Unseen Queries

The formulation proposed by RFE considered a pre-existing dataset, where all the elements of the dataset can be taken as queries. However, RFE also allows to perform queries with elements that do not belong to the dataset, in a formulation known in the literature as unseen queries. To make this possible, RFE follows a strategy proposed in [378] by decoupling off-line procedures (for the whole dataset) of on-line procedures (for the unseen query).

In an offline setting, the conventional steps of the method (normalization, re-ranking by embeddings, Cartesian product, re-ranking by connected components) are normally executed for all the known elements in the dataset. So, when a new external query (unseen query) needs to be evaluated, the k most similar elements are computed for each of them and a h-embedding is generated for the new query. The cosine distance between the query embedding and pre-computed embeddings in the whole dataset is used to rank the unseen query, producing the ranked lists for such elements.

8.1.8 Rank Aggregation

The RFE can also be exploited to fuse different features, in rank aggregation tasks. Different ranked lists sets $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_d\}$ are used as input with the objective of computing a more effective output set $\mathcal{T}_r$. The normalization step is performed individually for each of the rankers and the values are accumulated in a single sparse matrix M_f, once only top-L positions are considered. New ranked lists $\mathcal{T}_f$ are obtained by the sorting objects based on scores given by the matrix M_f. After that, the RFE (which can be understood as a function f_r) is executed for the ranked lists $\mathcal{T}_f$ and the list $\mathcal{T}_r$ is obtained as result:

$$\mathcal{T}_r = f_m(\mathcal{T}_f). \tag{8.34}$$

8.2 Experimental Evaluation

This section discusses the experimental evaluation conducted to assess the effectiveness of the proposed method. Section 8.2.1 describes the experimental settings. Section 8.2.2 discusses the impact of parameters while Section 8.2.3 presents an ablation study that includes an analysis of the impact of each step in our proposed method. Section 8.2.4 and 8.2.5 present the results on unsupervised retrieval and semi-supervised classification tasks, respectively. The results for unseen queries are described in Section 8.2.6. Sections 8.2.7 and 8.2.8 compare RFE with other state-of-the-art approaches for retrieval and classification, respectively. Finally, Section 8.2.9 presents visual analyses for both tasks.

8.2.1 Experimental Protocol

A broad experimental evaluation was conducted on 10 different image datasets. The datasets vary in size from 400 to 72,000 images. In this work, there are two different experimental scenarios: *(i)* unsupervised image retrieval, which was assessed on all 10 datasets; and *(ii)* semi-supervised image classification conducted on the Flowers and Corel5k datasets. The retrieval category encompasses not only general-purpose image datasets but also person Re-ID datasets (i.e., CUHK03, Market, Duke). More information about the datasets and descriptors is presented in Section 4.2.

Due to the highly diverse aspects of each dataset, we employed different evaluation measures in each case to enable comparisons with other approaches. In the classification task, we used accuracy as the evaluation measure. In contrast, for the retrieval task, other measures were used, with Mean Average Precision (MAP) being the most common. For Re-ID datasets, the R1 (which, in this case, is equivalent to Precision@1) was included, since it is commonly reported in the literature. For the UKbench dataset, which has the smallest number of images per class (only 4), the NS-Score was used. The NS-Score is the average of correct images at the top-4 positions of the ranked lists. Additional information about effectiveness measures can be found in Section 4.1.

We adopted the evaluation protocol for each dataset based on common practices in the literature. For most of them, all the images were considered as queries, except for Holidays [127] and Re-ID ones, where a different protocol was adopted [429, 422, 428]. For Holidays, there is a specific set of queries [127]. Regarding Re-ID, each dataset has a set of queries and a corresponding gallery set [429, 422, 428], which is the set of images that are ranked in relation to the query. The size of the ranked lists was set to $L = 400$ for most datasets, while the larger ones, such as the Re-ID datasets and ALOI, used $L = 2000$.

The semi-supervised classification relies on Graph Convolutional Networks (GCNs), which are stochastic. Since the results of the executions vary, we report an average of 5 executions on 10 different folds. This was adopted for our method and all the baselines. For unsupervised retrieval, the executions are deterministic.

8.2.2 Parametric Space Analysis

Initially, an experiment was conducted to visualize the impact of parameter α in the reciprocal sigmoid function, which is used in order to compute the rank normalization. This is the first step of our proposed approach, described in Section 8.1.2. The normalization mainly relies on Equation 8.4, which defines a reciprocal sigmoid function (σ). Figure 8.2 presents the values for Equation 8.4 (σ in y-axis) as the Rank Position ($\tau_x(y)$ in x-axis) varies. Different values of alpha were considered. The figure reveals that α is responsible for changing the steepness of the sigmoid curve, which refers to how quickly the output of the function changes as the input (i.e., the rank position) increases. However, it is challenging to determine an appropriate value of α based solely on this plot.

Based on this issue, an analysis was conducted with the objective of identifying default parameters. Figure 8.3 presents the impact of parameters α and T (number of iterations) on the MAP results for two datasets (i.e., Flowers and Corel5k). The CNN-ResNet [110] was considered for this experiment. Since we are not evaluating the parameter k in this case, we set it to the number of elements per class ($k = 100$). This is done to keep the focus of the analysis on α and T. The surface shows that the lowest values of α and T are more appropriate. Notice that the set of parameters $(T, \alpha) = (2, 0.1)$ is close to the best results in all cases (a, b, and c). Therefore, we used these values for all subsequent experiments.

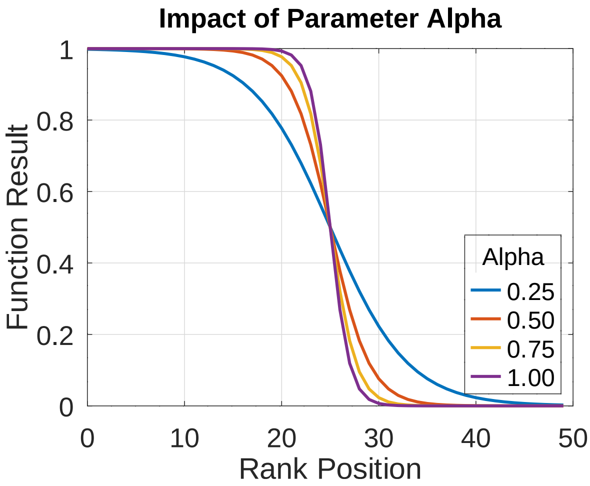
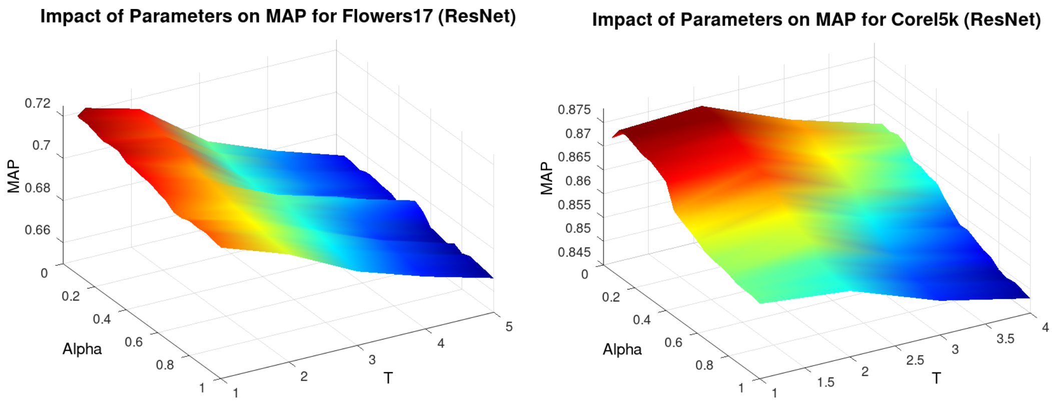


Figure 8.2 – Impact of parameter α in function σ (Equation 8.4) as the rank position varies.



(a) Flowers

(b) Corel5k

Figure 8.3 – Impact of parameters α and T (number of iterations) on MAP for two datasets.

8.2.3 Ablation Study

An ablation study was conducted to analyze the effectiveness of each step of the proposed method on 6 different datasets. We evaluated the retrieval results incrementally from Steps 1 to 4, as discussed in Section 8.1. Step (0) corresponds to the original features, Step (1) involves ranked lists normalization, Step (2) performs re-ranking by hypergraph embeddings, Step (3) computes re-ranking by Cartesian product, and Step (4) re-ranks by connected components. In this case, we excluded Step (5), which generates embeddings, as it is only necessary for semi-supervised classification.

Figure 8.4 presents the effectiveness results for every step of the proposed approach. For each dataset, two descriptors were evaluated. The descriptors considered were SWIN-TF [202], VIT-B16 [77], Inner Distance Shape Context (IDSC) [190], Contour Features Descriptor (CFD) [244], OSNET-AIN [436], and OSNET-IBN [436]; which are among the top-performing ones. The experiment was conducted using the best value of k in each case. Notice that the values consistently increase along the performed steps, indicating the relevance of each step. However, the datasets Holidays and Ukbench (c and e) revealed a different behavior, where Step 4 slightly decreases the MAP. This is probably caused by the fact that different from others, these datasets have a small number of images per class. Therefore, all the subsequent retrieval results presented in the next sections include Steps 1-4, except for the UKBench and Holidays datasets, which use Steps 1-3.

8.2.4 Retrieval Results

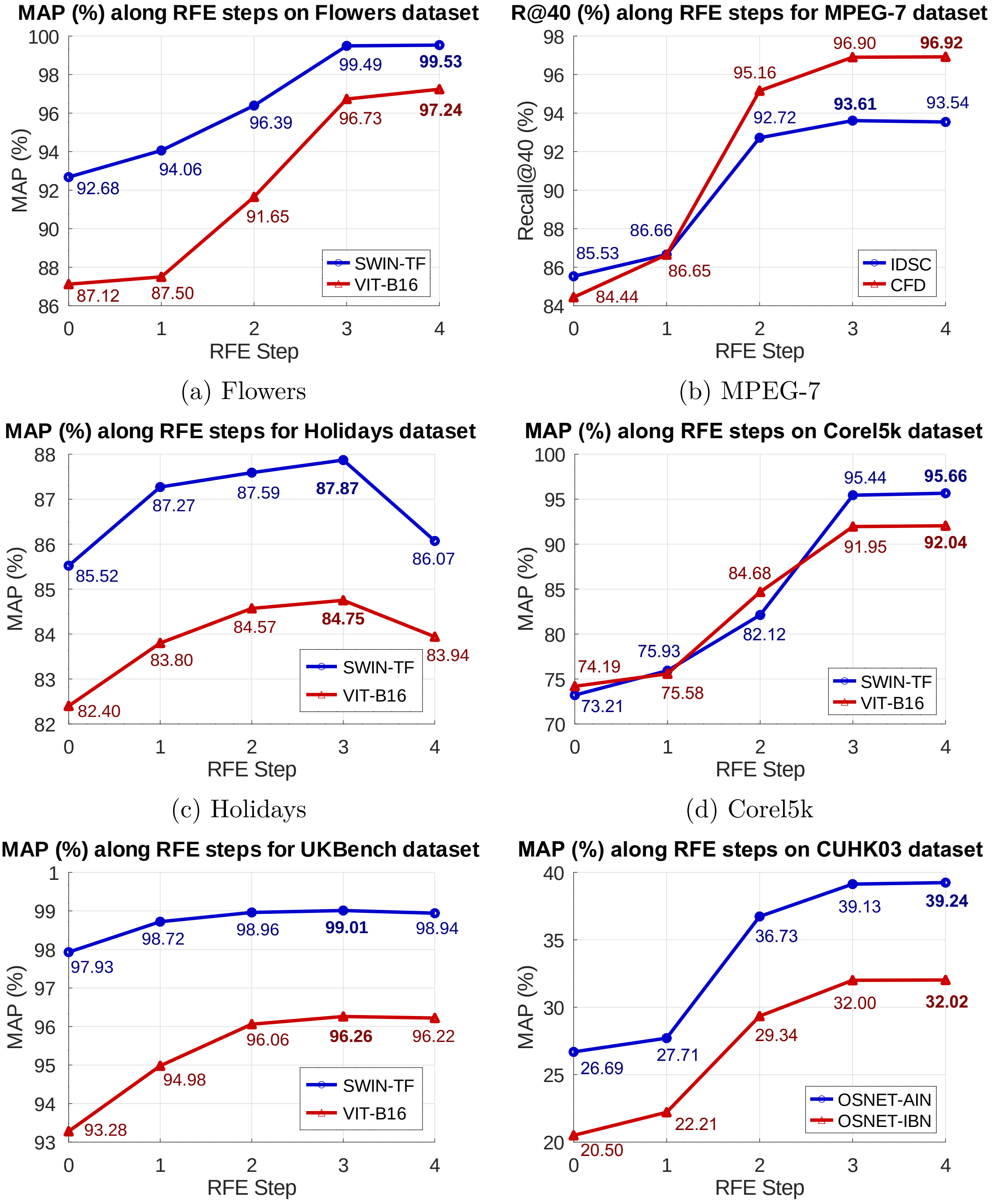
In image retrieval tasks, there are two different scenarios, which are both included in our evaluation: *(i)* standard re-ranking, where only one descriptor (feature) is considered; and *(ii)* rank-aggregation, which combines one or more features. For all experiments, we considered two variations for the parameter k (size of the neighborhood set): a default value [8] and the best value. The best k is reported considering the executions with k in range $[5, 120]$ with increments of 5. In general, the results revealed that our method is very robust to the change of k.

Firstly, we evaluate RFE on Flowers, Corel5k, and ALOI datasets; which are general-purpose image datasets that use the same protocol and evaluation measure. Table 8.1 presents the results. For standard re-raking, a relative gain was reported considering the improvement in relation to the original input descriptor. Since many descriptors are combined in rank aggregation, a gain is not reported in these scenarios. Notice that for all the cases, significant gains were obtained (up to +50.84%), and the fusion was able to improve the results even further. The best result for each dataset is highlighted in bold and marked with a gray background. For the three datasets, the best MAP is above 95%.

The same set of experiments was conducted for two datasets commonly used as image retrieval benchmarks: Holidays and UKbench. Since they have a small number

[8] The default values are: $k = 60$ for Flowers and Corel5k; $k = 5$ for Holidays and UKBench; and $k = 20$ for all the others.

of images per class, the best k is reported considering all the executions with k in the range $[1, 20]$ with increments of 1. Tables 8.2 and 8.3 present the results for Holidays and Ukbench, respectively. As can be seen, expressive gains were obtained for both datasets and measures. For single descriptor executions, positive gains were obtained in all the cases, achieving gains up to +7.42%. For NS-Score, the results are very close to the maximum value, which is 4. It is also possible to notice a correlation between MAP and NS-Score.



(a) Flowers

(b) MPEG-7

(c) Holidays

(d) Corel5k

(e) UKBench

(f) CUHK03

Figure 8.4 – Ablation study on six datasets considering two descriptors each. The graphs present the effectiveness values (MAP or R@40 depending on the dataset) for each step of the proposed approach. The best value for each plot is highlighted in bold.

Table 8.1 – Retrieval results of the proposed method (RFE) on general-purpose image datasets (Flowers, Corel5k, and ALOI). The results are reported for MAP (%) evaluation measure considering re-ranking (single descriptor) and rank-aggregation (fusion of descriptors). The best values for each dataset are highlighted in bold with a gray background.

Descriptors	**Original MAP**	**Method w/ default k**	**Method w/ best k**	**Relative Gain**
		Flowers		
		Re-Ranking		
CNN-DPNet [51]	49.06	69.47	69.95 (k=70)	+42.58%
CNN-ResNet [110]	50.00	72.32	72.62 (k=75)	+45.23%
CNN-SENet [117]	40.85	61.26	61.26 (k=60)	+49.96%
CNN-Xception [52]	45.27	66.65	66.81 (k=65)	+47.57%
T2T-VIT24T [399]	38.03	54.99	55.03 (k=70)	+44.73%
VIT-B16 (VIT) [77]	87.12	92.28	97.24 (k=80)	+11.61%
SWIN-TF (STF) [202]	92.68	97.96	99.53 (k=85)	+7.39%
		Rank-Aggregation		
ResNet+DPNet	—	80.07	80.13 (k=75)	—
VIT+ResNet	—	94.63	97.67 (k=80)	—
VIT+STF	—	**98.07**	**99.65 (k=85)**	—
VIT+ResNet+STF	—	97.64	99.28 (k=90)	—
		Corel5k		
		Re-Ranking		
CNN-DPNet [51]	63.69	81.58	85.48 (k=100)	+34.22%
CNN-ResNet [110]	63.46	84.11	87.97 (k=100)	+38.61%
CNN-SENet [117]	55.57	78.77	83.38 (k=100)	+50.06%
CNN-Xception [52]	52.92	76.33	79.82 (k=90)	+50.84%
T2T-VIT24T [399]	58.97	80.46	84.10 (k=100)	+42.62%
VIT-B16 (VIT) [77]	74.19	90.02	92.04 (k=100)	+24.06%
SWIN-TF (STF) [202]	73.21	93.55	95.66 (k=105)	+30.70%
		Rank-Aggregation		
ResNet+DPNet	—	87.66	91.22 (k=100)	—
VIT+ResNet	—	93.28	95.01 (k=100)	—
VIT+STF	—	**95.39**	**96.79 (k=100)**	—
VIT+ResNet+STF	—	95.20	96.79 (k=100)	—
		ALOI		
		Re-Ranking		
CNN-DPNet [51]	79.09	94.45	96.32 (k=30)	+21.79%
CNN-ResNet [110]	81.97	94.79	96.37 (k=30)	+17.57%
CNN-SENet [117]	78.41	93.91	95.87 (k=30)	+22.27%
CNN-Xception [52]	76.07	93.40	95.36 (k=30)	+25.36%
T2T-VT24T [399]	76.90	93.46	95.36 (k=30)	+24.00%
VIT-B16 (VIT) [77]	79.40	93.55	95.40 (k=30)	+20.16%
SWIN-TF (STF) [202]	89.97	96.68	97.81 (k=30)	+8.71%
		Rank-Aggregation		
ResNet+DPNet	—	95.71	97.06 (k=30)	—
VIT+ResNet	—	95.70	97.13 (k=30)	—
VIT+STF	—	96.07	97.53 (k=30)	—
VIT+ResNet+STF	—	**96.59**	**97.73 (k=30)**	—

Table 8.2 – Retrieval results of the proposed method (RFE) on the Holidays dataset. The results are reported for MAP (%) evaluation measure considering re-ranking (single descriptor) and rank-aggregation (fusion of descriptors). The best values are highlighted in bold with a gray background.

Descriptors	**Original MAP**	**Method w/ default k**	**Method w/ best k**	**Relative Gain**
		Re-Rank		
CNN-DPNet [51]	70.58	74.64	75.00 (k=6)	+6.25%
CNN-OLDFP [222]	88.46	89.58	90.11 (k=6)	+1.87%
CNN-ResNet [110]	74.87	77.15	77.37 (k=4)	+3.33%
CNN-SENet [117]	71.59	74.36	74.36 (k=5)	+3.88%
CNN-Xception [52]	64.93	68.24	68.48 (k=6)	+5.46%
T2T-VIT24T [399]	69.04	73.98	74.03 (k=6)	+7.23%
VIT-B16 (VIT) [77]	82.40	84.75	84.75 (k=5)	+2.85%
SWIN-TF (STF) [202]	85.52	87.87	87.87 (k=5)	+2.75%
		Rank-Aggregation		
VIT+ResNet	—	86.11	86.22 (k=6)	—
VIT+OLDFP	—	**91.64**	**91.97 (k=4)**	—
ResNet+OLDFP	—	88.08	88.33 (k=4)	—
OLDFP+STF	—	90.84	90.88 (k=4)	—
VIT+ResNet+OLDFP	—	89.98	90.35 (k=4)	—
VIT+OLDFP+STF	—	90.90	91.52 (k=4)	—

Table 8.3 – Retrieval results of the proposed method (RFE) on the UKBench dataset. The results are reported for both NS-Score and MAP evaluation measures considering re-ranking (single descriptor) and rank-aggregation (fusion of descriptors). The best values are highlighted in bold with a gray background.

Evaluation Measure	**NS-Score**				**MAP (%)**			
Descriptors	**Original NS-Score**	**Method w/ default k**	**Method w/ best k**	**Relative Gain**	**Original MAP**	**Method w/ default k**	**Method w/ best k**	**Relative Gain**
		Re-Ranking				**Re-Ranking**		
CNN-DPNet [51]	3.46	3.71	3.72 (k=6)	+7.42%	90.47	94.58	94.67 (k=6)	+4.65%
CNN-OLDFP [222]	3.85	3.93	3.93 (k=5)	+2.24%	97.74	98.92	98.92 (k=5)	+1.21%
CNN-ResNet [110]	3.67	3.85	3.85 (k=6)	+4.94%	94.54	97.31	97.31 (k=5)	+2.93%
CNN-SENet [117]	3.56	3.76	3.76 (k=5)	+5.52%	92.15	95.55	95.55 (k=5)	+3.69%
CNN-Xception [52]	3.49	3.75	3.75 (k=6)	+7.60%	90.83	95.35	95.35 (k=6)	+4.99%
T2T-VIT24T [399]	3.48	3.75	3.75 (k=5)	+7.78%	90.26	95.40	95.40 (k=5)	+5.69%
VIT-B16 [77]	3.62	3.80	3.80 (k=6)	+5.00%	93.28	96.26	96.26 (k=5)	+3.19%
SWIN-TF [202]	3.86	3.94	3.94 (k=6)	+2.01%	97.93	98.98	99.01 (k=6)	+1.10%
		Rank-Aggregation				**Rank-Aggregation**		
VOC+OLDFP	—	3.90	3.90 (k=6)	—	—	98.22	98.22 (k=5)	—
VOC+ResNet	—	3.92	3.93 (k=6)	—	—	98.76	98.79 (k=6)	—
VOC+VIT-B16	—	3.92	3.92 (k=6)	—	—	98.69	98.77 (k=7)	—
OLDFP+ResNet	—	3.94	3.95 (k=6)	—	—	99.13	99.13 (k=5)	—
OLDFP+VIT-B16	—	3.93	3.94 (k=6)	—	—	98.94	98.99 (k=6)	—
ResNet+VIT-B16	—	3.91	3.91 (k=5)	—	—	98.45	98.45 (k=5)	—
OLDFP+SWIN-TF	—	**3.97**	**3.97 (k=6)**	—	—	**99.53**	**99.57 (k=6)**	—
VOC+OLDFP+ResNet	—	3.94	3.94 (k=6)	—	—	99.07	99.07 (k=5)	—
VOC+OLDFP+VIT-B16	—	3.94	3.95 (k=6)	—	—	99.09	99.13 (k=6)	—
VOC+ResNet+VIT-B16	—	3.94	3.95 (k=6)	—	—	99.13	99.15 (k=6)	—
OLDFP+ResNet+VIT-B16	—	3.94	3.94 (k=6)	—	—	99.07	99.08 (k=6)	—
OLDFP+ResNet+SWIN-TF	—	3.96	3.96 (k=6)	—	—	99.40	99.41 (k=6)	—
VOC+OLDFP+ResNet+VIT-B16	—	3.95	3.95 (k=6)	—	—	99.20	99.28 (k=7)	—
VOC+OLDFP+VIT-B16+SWIN-TF	—	3.96	3.96 (k=6)	—	—	99.36	99.43 (k=6)	—

We also assessed RFE for person Re-ID (i.e., CUHK03, Market, and Duke datasets). These datasets are usually more challenging. They involve identifying and matching individuals across different camera views or even across different locations and times.

People's appearances can vary significantly due to changes in lighting, pose, clothing, and accessories. These factors can make it difficult to match the same person in different images. Table 8.4 reports the results on these datasets. Since R1 is also commonly used for Re-ID evaluation, it was also included. The R1 corresponds to the first value of the CMC (Cumulative Matching Characteristics) curve, which indicates the number of ranked lists that have an image that corresponds to the same individual in the first position after the query image (which, in this case, is equivalent to Precision@1).

The best k is reported considering all the executions with k in the range $[5, 50]$ with increments of 5. Notice that significant gains were obtained in all the cases (up to +65.88%), which were also improved by the rank-aggregation in most cases. These results reveal the potential of our approach in dealing not only with general-purpose scenarios but also with other challenging and more specific ones such as Re-ID.

Table 8.4 – Retrieval results of the proposed method (RFE) on three person Re-ID datasets (CUHK03, Market, and Duke). The results are reported for both R1 and MAP evaluation measures considering re-ranking (single descriptor) and rank-aggregation (fusion of descriptors). The best values are highlighted in bold with a gray background (MAP as the criteria).

Evaluation Measure	**R1 (%)**				**MAP (%)**			
Descriptors	**Original R1**	**Method w/ default k**	**Method w/ best k**	**Relative Gain**	**Original MAP**	**Method w/ default k**	**Method w/ best k**	**Relative Gain**
	CUHK03							
	Re-Ranking				**Re-Ranking**			
HACNN [177]	8.36	12.80	12.80 (k=20)	+53.03%	9.33	14.27	14.41 (k=15)	+54.42%
MLFN [39]	9.47	13.69	13.79 (k=15)	+45.63%	9.85	15.14	15.18 (k=15)	+54.11%
OSNet-AIN [436]	26.39	**36.67**	**36.89 (k=15)**	+39.76%	26.69	**39.12**	**39.24 (k=15)**	+47.00%
OSNet-IBN [436]	20.31	29.65	29.82 (k=15)	+46.85%	20.50	31.94	32.02 (k=15)	+56.18%
ResNet50 [110]	12.24	17.84	18.37 (k=15)	+50.15%	12.74	19.77	19.77 (k=20)	+55.18%
	Rank-Aggregation				**Rank-Aggregation**			
OSNet-AIN+OSNet-IBN	—	36.19	37.16 (k=15)	—	—	38.51	39.13 (k=15)	—
OSNet-AIN+ResNet50	—	33.54	33.54 (k=20)	—	—	35.40	35.40 (k=20)	—
OSNet-IBN+ResNet50	—	29.56	29.56 (k=20)	—	—	31.40	31.40 (k=20)	—
OSNet-AIN+OSNet-IBN+ResNet50	—	33.91	33.91 (k=20)	—	—	35.94	35.94 (k=20)	—
	Market							
	Re-Ranking				**Re-Ranking**			
HACNN [177]	49.23	52.20	52.82 (k=15)	+7.30%	22.29	31.93	32.10 (k=25)	+44.02%
MLFN [39]	46.59	49.58	49.76 (k=15)	+6.82%	21.11	30.65	30.89 (k=25)	+46.30%
OSNet-AIN [436]	69.95	70.99	70.99 (k=20)	+1.49%	42.33	57.38	58.21 (k=25)	+37.52%
OSNet-IBN [436]	66.45	67.25	67.90 (k=15)	+2.19%	36.31	52.71	53.23 (k=25)	+46.60%
ResNet50 [110]	46.59	51.72	51.90 (k=15)	+11.41%	21.92	34.09	34.81 (k=25)	+58.82%
	Rank-Aggregation				**Rank-Aggregation**			
OSNet-AIN+OSNet-IBN	—	**72.42**	**72.42 (k=20)**	—	—	**58.55**	**59.51 (k=25)**	—
OSNet-AIN+ResNet50	—	67.34	67.34 (k=20)	—	—	52.19	52.88 (k=25)	—
OSNet-IBN+ResNet50	—	64.61	64.61 (k=20)	—	—	49.45	50.40 (k=25)	—
OSNet-AIN+OSNet-IBN+ResNet50	—	68.20	68.53 (k=15)	—	—	54.35	55.11 (k=25)	—
	Duke							
	Re-Ranking				**Re-Ranking**			
HACNN [177]	42.19	50.31	50.99 (k=25)	+20.85%	24.37	39.32	40.42 (k=25)	+65.88%
MLFN [39]	48.65	56.06	56.73 (k=25)	+16.61%	28.00	44.00	45.39 (k=25)	+62.13%
OSNet-AIN [436]	71.14	75.67	76.84 (k=25)	+8.01%	51.68	66.60	68.31 (k=30)	+32.19%
OSNet-IBN [436]	67.41	73.88	75.00 (k=25)	+11.25%	44.66	63.60	64.81 (k=25)	+45.12%
ResNet50 [110]	52.29	60.50	62.57 (k=30)	+19.66%	31.00	48.77	50.67 (k=25)	+63.45%
	Rank-Aggregation				**Rank-Aggregation**			
OSNet-AIN+OSNet-IBN	—	**76.21**	**77.69 (k=25)**	—	—	**67.46**	**69.21 (k=25)**	—
OSNet-AIN+ResNet50	—	72.80	74.55 (k=30)	—	—	63.71	65.50 (k=25)	—
OSNet-IBN+ResNet50	—	72.26	74.10 (k=30)	—	—	62.65	64.09 (k=25)	—
OSNet-AIN+OSNet-IBN+ResNet50	—	74.69	76.17 (k=25)	—	—	65.74	67.02 (k=30)	—

8.2.5 Classification Results

The proposed approach is capable of generating embeddings that can be utilized in various applications beyond retrieval. In this section, we employ RFE for semi-supervised classification on two general-purpose image datasets (i.e., Flowers and Corel5k). The process of embedding generation is unsupervised and encompasses all the steps of the proposed approach (from 1 to 5). Our hypothesis is that the RFE embeddings can be used to train semi-supervised classifiers, resulting in improved accuracy. We employed recent Graph Convolutional Neural Networks (GCNs) models along with the traditional Support Vector Machine (SVM) with a polynomial kernel. The GCNs can operate on graphs, and they have become increasingly popular due to their ability to handle complex relationships between data points, which cannot be easily modeled using traditional machine learning methods. The RFE embeddings were evaluated by applying z-score normalization, followed by concatenation with the original features. Subsequently, dimensionality reduction to 200 dimensions was performed using Principal Component Analysis (PCA) [9].

Tables 8.5 and 8.6 present the results on Flowers and Corel5k datasets, respectively. In all the classifiers, the default parameters were used, proposed by the original authors. The GCNs were trained considering 50 epochs and $k = 40$ for the input kNN graphs. Our study compares the accuracy of classifiers that used the original features with those that used embeddings generated by the proposed RFE. We highlight in bold the best result for each classifier and in red the best for each dataset. The results demonstrate that the embeddings generated by our proposed approach are effective and have the potential to improve results across various classifiers. Notably, positive gains were obtained for all methods and features.

Table 8.5 – Semi-supervised classification (accuracy) on Flowers dataset for different features. We compare the training that used the original features with the one that used embeddings generated by RFE. The best result for each classifier is highlighted in bold and the best for each dataset is highlighted in red.

					GCN		
Mode	**Descriptor**	**SVM [54]**	**NET [146]**	**GAT [344]**	**SGC [363]**	**APPNP [147]**	**ARMA [30]**
Original	**ResNet [110]**	82.467%	69.386%	71.211%	78.649%	72.186%	60.475%
	DPNet [51]	79.812%	72.954%	18.874%	76.292%	70.539%	56.539%
	SENet [117]	76.193%	68.895%	63.18%	72.835%	66.797%	60.649%
RFE Embeddings	**ResNet [110]**	**82.565%**	**82.593%**	**82.966%**	84.948%	**83.974%**	**75.160%**
	DPNet [51]	80.131%	80.003%	41.237%	81.603%	81.029%	67.784%
	SENet [117]	76.716%	76.618%	73.454%	77.559%	77.260%	70.382%
Relative Gain	**ResNet [110]**	+0.12%	+19.03%	+16.51%	+8.01%	+16.33%	+24.28%
	DPNet [51]	+0.40%	+9.66%	+118.49%	+6.96%	+14.87%	+19.89%
	SENet [117]	+0.69%	+11.21%	+16.26%	+6.49%	+15.66%	+16.05%

[9] Z-Score normalization and PCA were performed with scikit-learn using default parameters.

Table 8.6 – Semi-supervised classification (accuracy) on Corel5k dataset for different features. We compare the training that used the original features with the one that used embeddings generated by RFE. The best result for each classifier is highlighted in bold and the best for each dataset is highlighted in red.

			GCN				
Mode	**Descriptor**	**SVM [54]**	**NET [146]**	**GAT [344]**	**SGC [363]**	**APPNP [147]**	**ARMA [30]**
Original	**ResNet [110]**	89.504%	78.066%	87.68%	90.288%	86.679%	73.621%
	DPNet [51]	87.662%	84.733%	18.349%	87.389%	85.653%	72.883%
	SENet [117]	88.613%	88.627%	87.292%	90.404%	88.76%	83.447%
RFE Embeddings	**ResNet [110]**	**89.602%**	90.008%	91.003%	91.54%	91.507%	89.212%
	DPNet [51]	87.933%	89.488%	52.374%	90.515%	91.061%	85.135%
	SENet [117]	88.776%	**91.299%**	**91.441%**	**91.97%**	**92.198%**	**90.924%**
Relative Gain	**ResNet [110]**	+0.11%	+15.3%	+3.79%	+1.39%	+5.57%	+21.18%
	DPNet [51]	+0.31%	+5.61%	+185.43%	+3.58%	+6.31%	+16.81%
	SENet [117]	+0.18%	+3.01%	+4.75%	+1.73%	+3.87%	+8.96%

8.2.6 Unseen queries

Encountering scenarios where query images are not included in the dataset being evaluated is not uncommon. These are referred to as external or unseen queries. To assess the proposed approach in such cases, we conducted experiments on the Flowers, Corel5k, and ALOI datasets, which are presented in Table 8.7.

We generated a set of unseen queries by randomly removing elements from the original dataset. To ensure a balanced analysis, we generated 10 samples per dataset, with each sample containing one element from each class. The reported MAP (both original and RFE) reflects the effectiveness of the approach in handling unseen queries, where the improvement is visible for all datasets and features.

Table 8.7 – Evaluation of RFE on unseen queries considering MAP (%). The reported results are the average of 10 executions, each conducted on a different set of unseen queries randomly sampled from the dataset.

Dataset	**Descriptor**	**Original**	**RFE**
Flowers	CNN-ResNet	52.3226	65.4526
	VIT-B16	89.0063	93.3823
	SWIN-TF	93.0988	95.3603
Corel5k	CNN-ResNet	63.2227	76.3823
	VIT-B16	75.2124	84.8642
	SWIN-TF	72.3914	82.5962
ALOI	CNN-ResNet	82.5268	88.4239
	VIT-B16	80.1258	85.8109
	SWIN-TF	89.7562	93.1862

8.2.7 Comparison with State-of-the-art for Unsupervised Image Retrieval

This section aims to present the comparisons of the best results obtained by the proposed RFE (reported in Section 8.2.4) in relation to recent baselines and state-of-the-art approaches on unsupervised image retrieval.

Table 8.8 presents the results for ORL and MPEG-7 datasets, which are two traditional benchmark datasets. These datasets are used for comparison with different diffusion methods. The ORL consists of images of faces, while the MPEG-7 is composed of images of shapes and contours. In order to keep consistency with the baselines, the same features were used for all the approaches: the IDSC [190] for MPEG-7 and raw images for ORL. The result with the original features is reported as "Our Baseline". The best values are highlighted in bold for each dataset. Notice that RFE achieved the best result for ORL and comparable ones for MPEG-7.

Table 8.8 – State-of-the-art (SOTA) comparison with other variants of diffusion processes on the ORL (R@15) and the MPEG-7 (R@40) datasets.

Methods	**ORL**	**MPEG-7**
Baseline [18]	62.35	85.40
SD [347]	71.67	83.09
LCDP [383]	74.25	89.45
TPG [385]	73.90	89.06
MR [434]	77.05	89.26
MR* [434]	77.58	92.61
GDP [74]	77.42	90.96
RDP (Y=I) [18]	78.53	**93.77**
RDP (Y=W) [18]	79.27	**93.78**
Our Baseline	74.32	85.40
RFE **(our method)**	**90.62** **(k=10)**	**93.54** **(k=20)**

The state-of-the-art comparison also encompasses the Flowers, Corel5k, and ALOI datasets; which is shown in Table 8.9. Our method outperformed all other approaches, achieving the best results on all three datasets. The values reveal the effectiveness of RFE for both small and large datasets (Flowers and ALOI contain 13060 and 10200 images, respectively), with MAP always above 96.79%. This is a really significant result since the baselines also consider rank-aggregation of different features, especially Unsupervised Genetic Algorithm Framework for Rank Selection and fusion (UGAF-RSF) [327] and Unsupervised Selective Rank Fusion (USRF) [329] that combine more than 10 features.

Tables 8.10 and 8.11 compare the RFE results to state-of-the-art methods on Holidays and Ukbench datasets, respectively. These datasets are widely used as benchmarks for many retrieval algorithms. We compare RFE to at least 15 approaches for each dataset. Notice, that the results achieved by RFE are higher than the baselines in both cases. We achieved an NS-Score of 3.97 (the maximum possible value is 4.00).

Table 8.12 presents the results of different approaches on the Re-ID datasets considering both R1 and MAP. Our results (RFE) are marked with a gray background and correspond to the best ones according to Table 8.4. The abbreviations in parentheses indicate the datasets used for training (C03 = CUHK03, M = Market1501, D =

DukeMTMC, MT = MSMT17). For example, the use of (D, M) indicates that the reported result corresponds to training done either on Duke or on the Market dataset.

Table 8.9 – State-of-the-art comparison on Flowers, Corel5k, and ALOI datasets (MAP %).

Method	**Flowers**	**Corel5k**	**ALOI**
CPRR [332]	—	—	76.90
RL-Sim [240]	—	—	78.84
RL-Recom [335]	—	—	80.35
LHRR [251]	—	73.34	88.42
BFSTree [253]	—	53.00	91.15
RDPAC [252]	—	56.00	91.31
UGAF-RSF [327]	80.92	91.45	—
USRF [329]	81.71	90.32	—
RFE (Our Method)	**99.65**	**96.79**	**97.73**

Table 8.10 – State-of-the-art comparison on Holidays dataset (MAP).

MAP for state-of-the-art methods				
Jégou *et al.* [127]	Tolias *et al.* [315]	Paulin *et al.* [238]	Qin *et al.* [268]	Zheng *et al.* [425]
75.07%	82.20%	82.90%	84.40%	85.20%
Sun *et al.* [299]	Zheng *et al.* [423]	Pedronette *et al.* [241]	Arandjelovic *et al.* [12]	Li *et al.* [178]
85.50%	85.80%	86.16%	87.50%	89.20%
Razavian *et al.* [271]	Pedronette *et al.* [253]	Gordo *et al.* [104]	Valem *et al.* [329]	Valem *et al.* [328]
89.60%	90.02%	90.30%	90.51%	90.51%
Liu *et al.* [203]	Pedronette *et al.* [251]	Pedronette *et al.* [252]	Yu *et al.* [398]	Berman *et al.* [26]
90.89%	90.94%	91.25%	91.40%	91.80%

RFE (Our Method)
91.97%

Table 8.11 – State-of-the-art comparison on UKBench dataset (NS-Score).

N-S-Scores for state-of-the-art methods				
Qin *et al.* [267]	Zhang *et al.* [413]	Zheng *et al.* [424]	Bai *et al.* [16]	Xie *et al.* [371]
3.67	3.83	3.84	3.86	3.89
Lv *et al.* [210]	Liu *et al.* [203]	Pedronette *et al.* [241]	Bai *et al.* [20]	Liu *et al.* [159]
3.91	3.92	3.93	3.93	3.93
Valem *et al.* [328]	Bai *et al.* [17]	Valem *et al.* [329]	Valem *et al.* [327]	Chen *et al.* [50]
3.93	3.94	3.94	3.95	3.96

RFE (Our Method)
3.97

The results reported on Market were trained on Duke and the results reported on Duke were trained on Market. None of the presented methods were trained using labels of the target dataset. The abbreviations were omitted for multi-source baselines, but they can be consulted in their papers. The best results for each dataset are highlighted in bold. Notice, that our results are among the best in all the cases and are above all of the baselines for DukeMTMC considering MAP.

Table 8.12 – State-of-the-art (SOTA) comparison for person Re-ID datasets considering MAP (%) and R-01 (%). The abbreviations in parentheses indicate the datasets used for training (C03 = CUHK03, M = Market1501, D = DukeMTMC, MT = MSMT17). For example, the use of (D, M) indicates that the reported result corresponds to training done either on Duke or on the Market dataset. The results reported on Market were trained on Duke and the results reported on Duke were trained on Market. None of the presented methods were trained using labels of the target dataset.

Method	**Year**	**Datasets**					
		Market1501		**DukeMTMC**		**CUHK03**	
		R1	**MAP**	**R1**	**MAP**	**R1**	**MAP**
Unsupervised Methods							
ARN [181]	2018	70.3	39.4	60.2	33.4	—	—
EANet [118]	2018	66.4	40.6	45.0	26.4	51.4	31.7
TAUDL [170]	2018	63.7	41.2	61.7	43.5	44.7	31.2
ECN [431]	2019	75.1	43.0	63.3	40.4	—	—
UTAL [171]	2019	69.2	46.2	62.3	44.6	**56.3**	**42.3**
SSL [189]	2020	71.7	37.8	52.5	28.6	—	—
HCT [402]	2020	80.0	56.4	69.6	50.7	—	—
CAP [353]	2021	**91.4**	**79.2**	**81.1**	67.3	—	—
IICS [376]	2021	89.5	72.9	80.0	64.4	—	—
Domain Adaptive Methods							
HHL (D,M) [430]	2018	62.2	31.4	46.9	27.2	—	—
HHL (C03) [430]	2018	56.8	29.8	42.7	23.4	—	—
ATNet (D,M) [197]	2019	55.7	25.6	45.1	24.9	—	—
CSGLP (D,M) [273]	2019	63.7	33.9	56.1	36.0	—	—
ISSDA (D,M) [306]	2019	81.3	63.1	72.8	54.1	—	—
ECN++ (D,M) [432]	2020	84.1	63.8	74.0	54.4	—	—
MMCL (D,M) [348]	2020	84.4	60.4	72.4	51.4	—	—
Cross-Domain Methods (single-source)							
EANet (C03) [118]	2018	59.4	33.3	39.3	22.0	—	—
EANet (D,M) [118]	2018	61.7	32.9	51.4	31.7	—	—
SPGAN (D,M) [71]	2018	43.1	17.0	33.1	16.7	—	—
DAAM (D,M) [121]	2019	42.3	17.5	29.3	14.5	—	—
AF3 (D,M) [195]	2019	67.2	36.3	56.8	37.4	—	—
AF3 (MT) [195]	2019	68.0	37.7	66.3	46.2	—	—
PAUL (MT) [380]	2019	68.5	40.1	72.0	53.2	—	—
Cross-Domain Methods (multi-source)							
CAMEL [396]	2017	54.5	26.3	—	—	31.9	—
EMTL [370]	2018	52.8	25.1	39.7	22.3	—	—
Baseline by [153]	2019	80.5	56.8	67.4	46.9	29.4	27.4
Our Proposed Method							
Our Method		**72.42**	**59.51**	**77.69**	**69.21**	**36.89**	**39.24**

8.2.8 Comparison with State-of-the-art for Semi-Supervised Image Classification

This section compares the semi-supervised image classification results reported in Section 8.2.5 to various state-of-the-art approaches. Table 8.13 presents the comparisons considering different features (CNN-ResNet [110] and CNN-SENet [117]). The best result for each feature and dataset is highlighted in bold. The gray rows indicate the results that correspond to our method. We employed the same protocol adopted for RFE in all baselines: 5 executions of 10 folds. The only exception is CoMatch [169], where only 3 executions were reported for Corel5k due to the long time required to train this approach. Different from others, CoMatch takes images as input. However, it uses CNN-ResNet as its backbone.

Table 8.13 – Accuracy comparison (%) for baselines on Flowers and Corel5k datasets. We compared our approach with semi-supervised classification baselines. The methods are compared with different input features. The results of our method are highlighted with a gray background; the best results for each pair of features and dataset are marked in bold.

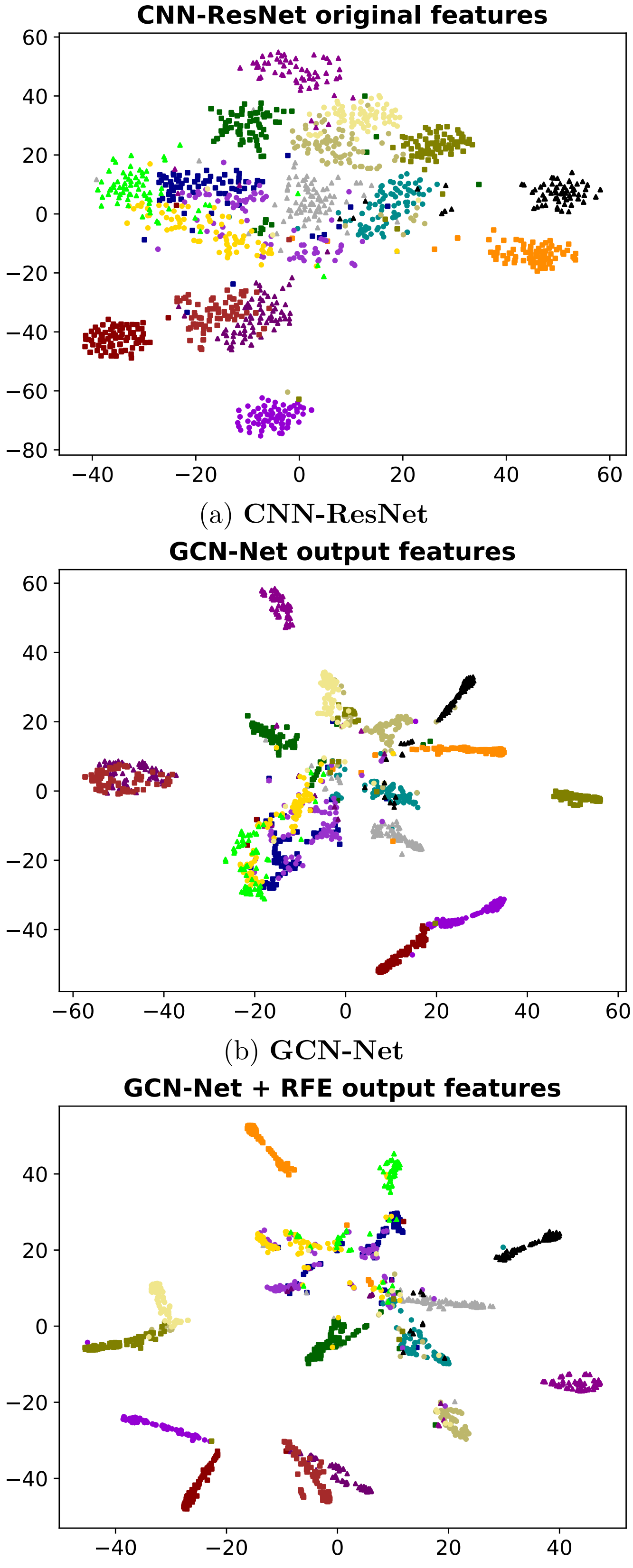
Method	**Input**	**Flowers**	**Corel5k**
CoMatch [169]	**Images**	82.55	*85.70*
kNN	**ResNet Features**	63.67	76.80
SVM [54]		80.54	88.73
OPF [8]		71.77	83.56
SL-Perceptron		75.44	83.56
ML-Perceptron		78.88	87.10
PseudoLabel+SGD [162]		82.69	89.76
LS+kNN [433]		73.49	83.98
LS+SVM [433, 54]		73.53	83.26
LS+OPF [433, 8]		72.66	82.32
LS+SL-Perceptron [433]		72.34	82.38
LS+ML-Perceptron [433]		73.03	82.53
GNN-LDS [90]		54.98	62.69
GNN-KNN-LDS [90]		79.32	88.94
WSEF [264]		**85.12**	**91.68**
RFE (Our Method)		84.95	91.54
kNN	**SENet Features**	48.71	58.78
SVM [54]		73.30	85.89
OPF [8]		64.00	81.33
SL-Perceptron		71.84	82.28
ML-Perceptron		72.62	86.90
PseudoLabel+SGD [162]		76.87	89.85
LS+kNN [433]		58.05	72.16
LS+SVM [433, 54]		59.84	72.79
LS+OPF [433, 8]		59.25	72.20
LS+SL-Perceptron [433]		59.27	72.19
LS+ML-Perceptron [433]		59.39	72.24
GNN-LDS [90]		52.24	65.80
GNN-KNN-LDS [90]		73.69	89.95
WSEF [264]		76.16	89.74
RFE (Our Method)		**77.56**	**92.20**

For all the methods, we considered the default parameters and implementation provided by the original authors or the one in *Python Sklearn*. Regarding parameters, we used $k = 20$ for methods that require a size for the neighborhood set (i.e., kNN, GNN-LDS, GNN-KNN-LDS, and WSEF). The Label Spreading (LS) [433] was used combined with different classifiers once it can be used to generate pseudo-labels for further expanding the training set. The results achieved by RFE are the best ones for the SENet features and very comparable to the best for the ResNet features.

8.2.9 Visual Analysis

In addition to the numerical analyses, qualitative experiments are also important for understanding the results achieved by the proposed approach. For better visualization of the improvements provided by RFE in the semi-supervised classification experiments, Figure 8.5 illustrates feature spaces on Flowers dataset with CNN-ResNet descriptor for three different cases: *(a)* features extracted by the CNN-ResNet descriptor; *(b)* GCN-Net output features after being trained on the CNN-ResNet features; and *(c)* GCN-Net output features after being trained on the CNN-ResNet features combined to the RFE embeddings. The t-SNE method was used to compute the coordinates in the 2D space. While each dot represents a different element of the dataset, each combination of color and shape corresponds to a distinct class. From *(a)* to *(b)*, it is evident that training with GCN was able to improve the separability between classes due to the ability of GCN to leverage the structure of graph data to aggregate and transform neighboring information. From *(b)* to *(c)*, the separability was further enhanced by the RFE embeddings, which encode the similarity of the elements to the connected components which were computed based on different contextual information, including the hypergraph structure. Notice that *(c)* presents the best correspondence among the visual groups formed by the dots and the original dataset classes. This evinces our hypothesis that the RFE embeddings improve the classification of GCNs.

Experiments were also conducted to visualize the performance of RFE in retrieval tasks. Figure 8.6 presents examples of ranked lists before and after the execution of our proposed method. These results were obtained on different datasets (CNN-ResNet for Flowers and Corel5k; and OSNET-AIN for DukeMTMC) with the default parameters and k. The query images are presented with green borders and the incorrect ones with red borders. The examples cover diverse scenarios, encompassing challenges such as similar images between different classes, occlusions, lightning, and viewpoint variances. Despite these challenges, RFE clearly demonstrated significant improvements across all queries.

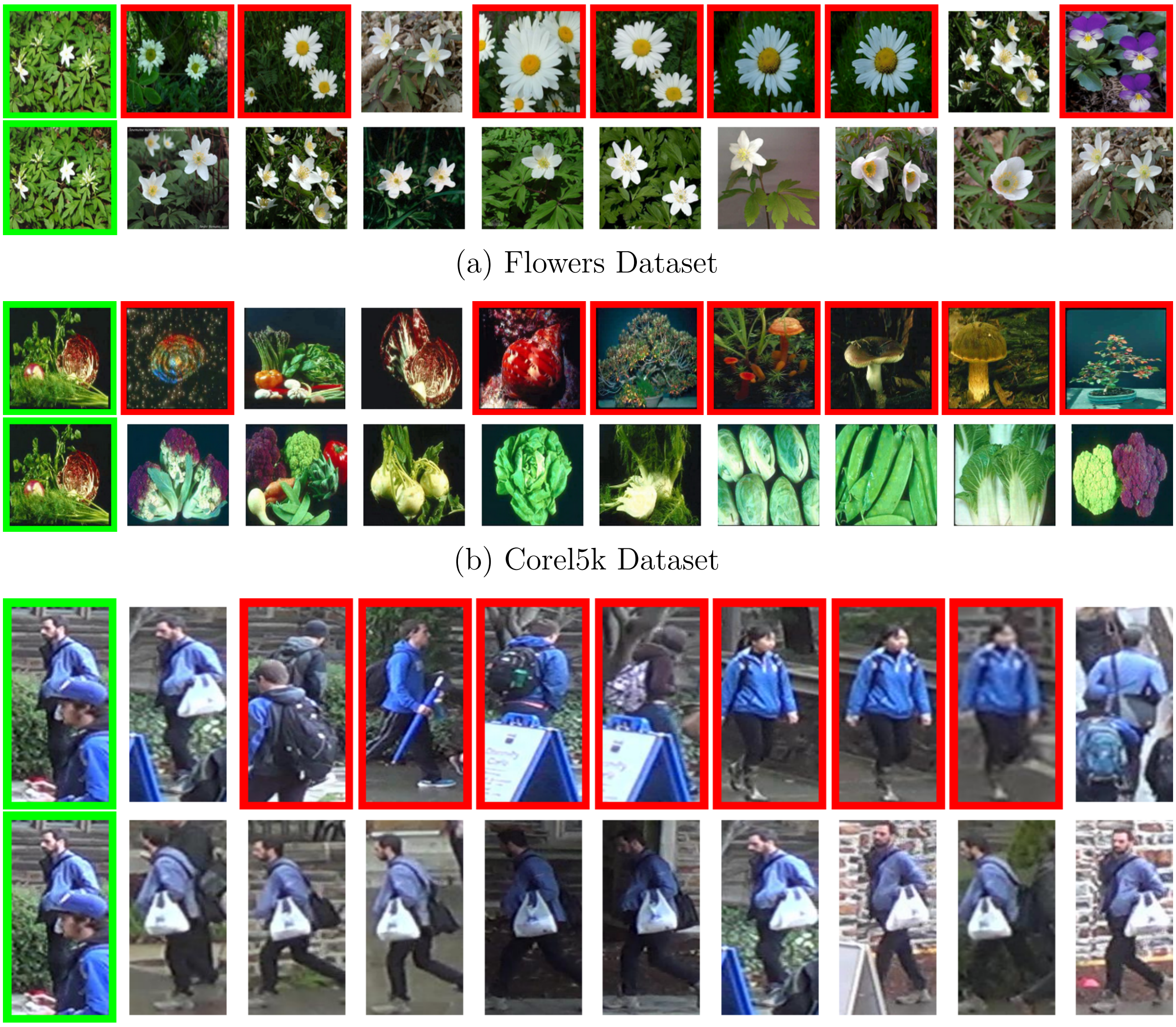


(a) **CNN-ResNet**

(b) **GCN-Net**

(c) **GCN-Net + RFE**

Figure 8.5 – Feature space illustrations computed by t-SNE on the Flowers dataset with the CNN-ResNet descriptor. It shows the (a) original feature space, (b) feature space obtained with the GCN, and (c) feature space obtained by the GCN using the RFE (our proposed approach) embeddings.



(a) Flowers Dataset

(b) Corel5k Dataset

(c) DukeMTMC Dataset

Figure 8.6 – Examples of ranked lists before and after RFE was applied for three datasets. Query images are highlighted with green borders and wrong results are with red borders.

9 Contextual Manifold Learning on Graph Convolutional Networks (Manifold-GCN)

Graph Convolutional Networks (GCNs) present an effective and emerging representation learning strategy. One of the core concepts involves convolution operations in a non-Euclidean domain defined by graph-structured data. In practice, a new representation is learned by aggregating feature representations from the graph-based neighborhood [410, 146]. The graph data is inherently available in some domains but needs to be inferred or constructed in others [90]. Consequently, several methods have been proposed for graph-structured data as citation datasets [344, 363, 147, 30, 45, 172, 19], but only a few approaches have been proposed for image and multimedia data [42, 379, 193, 405]. In scenarios where the data are not inherently represented as graphs, such structures can be constructed to reflect similarity relationships. In most cases, the most direct approach is to create a k-nearest neighbor graph. However, the GCN models are highly sensitive to the input graph, in the sense that a more effective classification depends on the edges between nodes of the same class. Therefore, defining a graph capable of encoding contextual information and representing more effective similarity relationships assumes a key role.

In this scenario, this chapter presents a novel GCN-based approach, the Manifold-GCN, for image classification in semi-supervised scenarios, where labeled data is limited. Deep features are extracted for image representation employing transfer learning by CNNs and Vision Transformers (ViT) models. Ranking structures are computed and used as input by unsupervised manifold learning algorithms based on these extracted features. Manifold learning approaches aim to capture and exploit the intrinsic manifold structure to compute a more effective distance/similarity measure [133]. In this work, we consider recent unsupervised manifold learning methods to provide more effective similarity measures using rank-based formulations.

The manifold learning methods produce more effective ranking results, i.e., improved neighbor sets, which are exploited for building the input graph of the GCN model. In addition to constructing kNN graphs, the use of reciprocal kNN graphs is proposed. The main hypothesis of this chapter is that the use of manifold learning to improve the graph structure provided as the input of the Graph Convolutional Network (GCN) can further improve the classification results obtained. This work proposes and validates this hypothesis on different manifold learning and recent GCN approaches.

We can highlight the main contributions of our work as follows: *(i)* novel ways to learn the graph structures that improve GCN image classification; *(ii)* the use of reciprocal kNN graph in order to provide a more reliable graph for GCNs. There are very few works that employ kNN graphs [90] or manifold learning [45] for GCNs. In [90] the traditional kNN graph is employed and [45] uses manifold learning, but in both works no image data is considered. Other few works have recently employed GCN models on image classification [42, 379, 193, 405]. However, to the best of our knowledge, this is the first work that exploits both manifold learning and reciprocal kNN graphs for GCN-based semi-supervised image classification. In addition, it combines powerful contextual modeling given by GCN models with effective representations given by CNNs and ViT features.

There are many applications of the proposed approach. The improvement of classification results using GCNs may benefit many different areas, especially when there is limited labeled data. For example: person re-identification [137] and diagnosis of diseases [1]. The Manifold-GCN can be employed in scenarios where the graph data is not previously available by building the graph from the features and employing manifold learning.

A wide experimental evaluation was conducted in order to assess the effectiveness of the proposed approach. The experimental results were obtained on 3 public datasets. We evaluated the impact of different GCN models combined with different manifold learning methods. The experimental results demonstrate the effectiveness of the proposed approach and the gains of combining manifold learning and reciprocal kNN graphs.

This chapter is organized as follows: Section 9.1 describes our proposed approach, the Manifold-GCN. Section 9.2 reports the experimental evaluation.

9.1 Proposed Method

In this work, we propose the Manifold-based Graph Convolutional Network (Manifold-GCN), a semi-supervised framework based on the combined use of manifold learning and GCN models for image classification in scenarios with limited labeled data. The initial representations were obtained by deep features extracted by CNN and ViT models trained on a transfer learning setting. Given the representations, the central idea consists of exploiting contextual similarity measures given by unsupervised manifold learning methods for computing a more effective graph. The similarity information encoded in the graph is exploited by GCN models for learning novel representations used for classification.

Figure 9.1 illustrates the main steps that compose our strategy. Each step is identified by a number (top of boxes) and a function (bottom of boxes). In **(1)**, a feature vector is extracted for representing each image. In **(2)**, representations are processed in order to obtain ranked lists, which encode the similarity information. Unsupervised manifold learning methods are used to analyze contextual similarity information and compute more effective rankings in **(3)**. In **(4)**, the outputs of the manifold learning methods are modeled as kNN graphs or reciprocal kNN graphs. In **(5)**, the graph and features are jointly provided to the GCN models for semi-supervised training. The embeddings obtained for each of the elements of the dataset can be used for classification, through a softmax operation. Each of the main steps of the framework is described in the next subsections.

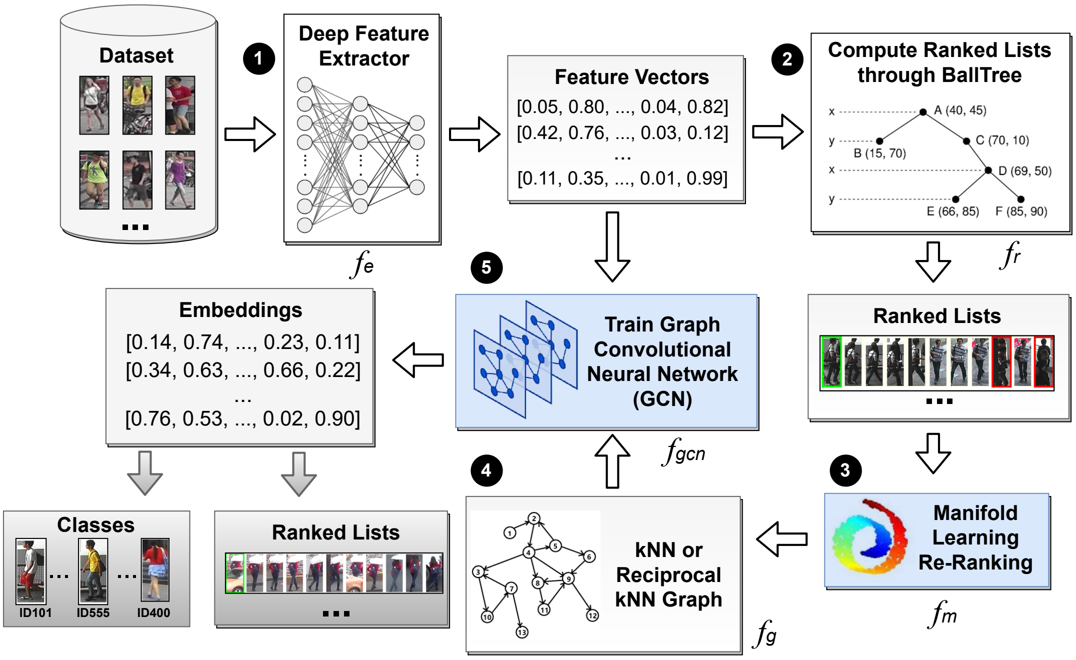


Figure 9.1 – Workflow of our proposed Manifold-GCN framework for image classification. The steps of the approach are numbered.

9.1.1 Similarity Measurement and Ranking Model

In the proposed approach, the similarity information is encoded on ranking structures. Let us consider a ranking task in which, given a query image, an ordered list of images from the collection is returned according to the similarity to the query. Formally, given a query image o_q, a ranked list τ_q=$(o_1, o_2, \ldots, o_L)$ in response to the query, where L denotes the length of the list. The ranked list τ_q can be defined as a permutation of a set $\mathcal{C}_L$ which contains the L most similar images to image o_q in the collection $\mathcal{C}$. The permutation τ_q is a bijection from the set $\mathcal{C}_L$ onto the set $[L] = \{1, 2, \ldots, L\}$. The $\tau_q(o_i)$ notation denotes the position (or rank) of image o_i in the ranked list τ_q.

The ranked list τ_q can be computed based on the comparison between image representations. Let δ: $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$ be a distance function that computes the distance between two images according to their corresponding feature vectors. The Euclidean distance is often used as the distance function. Formally, the distance between two images o_i, o_j is defined by $\delta(\mathbf{x}_i, \mathbf{x}_j)$.

For a given query, a ranked list can be obtained by sorting images in increasing order of the distance. In terms of ranking positions, we can say that if image o_i is ranked before image o_j in the ranked list of image o_q, that is, $\tau_q(o_i) < \tau_q(o_j)$, then $\delta(\mathbf{x}_q, \mathbf{x}_i) \leq \delta(\mathbf{x}_q, \mathbf{x}_j)$. Taking every image in the collection as a query image o_q, a set of ranked lists $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_n\}$ can be obtained. In this way, the set $\mathcal{T}$ can be obtained from the feature matrix $\mathbf{X}$ and the ranking task defined by a function f_r, such that $\mathcal{T} = f_r(\mathbf{X})$. Tree-based indexing structures [234] and hashing approaches [99] can be exploited in order to provide efficient implementations for the function f_r. In this work, we consider BallTree [234, 239] structures.

9.1.2 Unsupervised Manifold Learning

How to accurately define distance or similarity among data elements is a challenging and fundamental step in many machine learning tasks. The most common approach is given by pairwise comparisons based on Euclidean-like distance functions. However, pairwise analyses ignore contextual information and complex similarity arrangements encoded in the structural information of the dataset manifold. Aiming at addressing such drawbacks, many contextual similarity approaches take into account the structure of datasets in order to compute more global and effective similarity measures.

Manifold Learning is a wide term that has many different definitions in the literature. In general, manifold Learning approaches aim to capture and exploit the intrinsic manifold structure to compute a more effective distance/similarity measure [133]. Recently, unsupervised manifold learning approaches based on ranking information have achieved relevant advances in contextual similarity measurement [251, 252, 253].

In fact, the set of ranked lists $\mathcal{T}$ encodes rich similarity information about the image collection. The main objective of rank-based manifold learning methods is to exploit such information to capture the structure of the dataset manifold. Therefore, this step consists of the use of unsupervised manifold learning methods for processing the original ranked lists, providing more effective ranking results which are subsequently modeled as graphs to be submitted to a GCN model.

Formally, the manifold learning methods can be defined as a function f_m that receives a set of ranked lists $\mathcal{T}$ as input and returns a set of ranked lists $\mathcal{T}_m$ as output, which is expected to be more effective than the original:

$$\mathcal{T}_m = f_m(\mathcal{T}). \tag{9.1}$$

Once defined under a common formulation, three different manifold learning algorithms were considered to instantiate the proposed approach (described in Section 9.1.6).

9.1.3 Graph Building

The improved set of ranked lists computed by the manifold learning methods is used to build a graph. The motivation is based on the conjecture that more effective similarity information can be extracted and encoded in the graph by exploiting the processed ranked lists. Let $G = (V, \mathbf{X}, E)$ be the graph defined in Section 2.5.2. We propose to compute the edge set E as a function of the set of ranked lists $\mathcal{T}_m$, such that $E = f_g(\mathcal{T}_m)$.

This work considers two distinct approaches to define the function f_g. The similarity information encoded in the ranked lists is modeled through different neighborhood set formulations. Both approaches are discussed in the following.

- *Traditional kNN Graph:* The kNN graph is based on the natural neighborhood set. Given an element o_q, the natural neighborhood set $\mathcal{N}(o_q, k)$ contains the k most similar elements to o_q, which can be formally defined as:

$$\mathcal{N}(o_q, k) = \{\mathcal{X} \subseteq \mathcal{C}, |\mathcal{X}| = k \ \wedge \ \forall o_i \in \mathcal{X}, o_j \in \mathcal{C} - \mathcal{X} : \tau_q(o_i) < \tau_q(o_j)\}. \tag{9.2}$$

Therefore, the edge set E of the kNN graph can be defined as:

$$E = \{(o_q, o_j) \mid o_j \in \mathcal{N}(o_q, k)\}. \tag{9.3}$$

In other words, each element has an edge to the k most similar elements.

- *Reciprocal kNN Graph:* the reciprocal kNN graph is based on the reciprocal neighborhood set [267], which requires a stronger bidirectional similarity relationship. Different from the natural neighborhood set, which is not symmetrical, the reciprocal neighborhood set is symmetrically defined as:

$$\mathcal{N}_r(o_q, k) = \{obj_i | obj_i \in \mathcal{N}(o_q, k) \wedge o_q \in \mathcal{N}(o_i, k)\}. \tag{9.4}$$

The edge set E for the reciprocal kNN set can be defined as:

$$E = \{(o_q, o_j) \mid o_j \in \mathcal{N}_r(o_q, k)\}. \tag{9.5}$$

Thus, we can interpret that there are edges between the elements o_q and o_j if they are reciprocal neighbors in the top-k positions of their ranked lists.

For both kNN and reciprocal kNN approaches, the edge set E can be represented by a non-negative adjacency matrix $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$, which can be defined as:

$$a_{ij} = \begin{cases} 1, & (o_i, o_j) \in E \\ 0, & \text{otherwise.} \end{cases} \tag{9.6}$$

The adjacency matrix $\mathbf{A}$ is used as input by GCN models, as discussed in the next section.

9.1.4 Graph Convolutional Networks

Graph Convolutional Networks (GCN), originally introduced in [146], aim at learning novel and more effective representations (embeddings) for each graph node. It is done by iteratively aggregating the embeddings of its neighbors, encoding the graph structure directly in a neural network model. The original model proposed in [146] is a two-layer GCN model that uses the graph represented by the adjacency matrix $\mathbf{A}$ for semi-supervised node classification.

The network model can be depicted as a function both on the feature data $\mathbf{X}$ and on the adjacency matrix $\mathbf{A}$, as:

$$\mathbf{Z} = f_{gcn}(\mathbf{X}, \mathbf{A}), \tag{9.7}$$

where $\mathbf{Z}$ denotes an embedding matrix, such that $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n]^T \in \mathbb{R}^{n \times c}$ and $\mathbf{z}_i$ is a c-dimensional embedded representation learned for the node v_i; where n is the dataset size and c corresponds to the number of classes.

The degree matrices are computed as a pre-processing step, defined as $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-1/2}$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\tilde{\mathbf{D}}$ is the degree matrix of $\tilde{\mathbf{A}}$. Then, the function $f_{gcn}(\cdot)$ which represents the two-layer GCN model assumes the form:

$$\mathbf{Z} = f(\mathbf{X}, \mathbf{A}) = softmax(\hat{\mathbf{A}}\ ReLU(\hat{\mathbf{A}}\mathbf{X}\mathbf{W}^{(0)})\mathbf{W}^{(1)}). \tag{9.8}$$

The matrix $\mathbf{W}^{(0)} \in \mathbb{R}^{d \times H}$ defines the neural network weights for an input-to-hidden layer with H feature maps, while $\mathbf{W}^{(1)} \in \mathbb{R}^{H \times c}$ is a hidden-to-output matrix. Both matrices $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$ are trained using gradient descent, considering the cross-entropy error over all labeled nodes. $v_l \in V_L$.

The activation function is applied row-wise and is defined as $softmax(z_i) = \frac{exp(z_i)}{\sum_i exp(z_i)}$, where z_i is the position i of embedding $\mathbf{z}_i$.

The softmax yields the probability distribution over the c class labels for each row, i.e., the probability values sum up to 1 for each row. Given an image o_i, the learned embedded representation $\mathbf{z}_i$ is then used for classification tasks by applying an argmax over the output of the softmax.

9.1.5 GCN Models

The original GCN [146] model and more 4 variants [147, 363, 30] are used in the proposed Manifold-GCN approach. The GCN models employed are:

- **Graph Convolution Network (GCN)** [146]: The first GCN proposed, introducing the idea of convolutions applied to graph domains, often known as GCN-Net or simply GCN.
- **Simple Graph Convolution (SGC)** [363]: A simplification of the conventional GCN models which removes the non-linearities and collapses weight matrix between consecutive layers.
- **Graph Attention Networks (GAT)** [344]: Employs auto-attention layers with the idea of solving the main shortcomings of the previous GCN models. The layers are stacked in a way that it is possible to specify different weights for nodes of the same neighborhood without requiring costly operations.
- **Approximate Personalized Propagation of Neural Predictions (APPNP)** [147]: A model that combines a GCN with the PageRank algorithm, deriving a propagation strategy based on a modified PageRank approach.
- **Auto-Regressive Moving Average (ARMA) Filter Convolutions** [30]: A GCN variant that defines convolutional layers based on filters of Auto-Regressive Moving Average type.

9.1.6 Manifold Learning Methods

Manifold learning can be broadly understood as the process of non-linear dimensionality reduction by performing distance learning for a set of features. In fact, images are commonly represented as points in a high-dimensional feature space. However, it has been shown that data samples often live in a much lower dimensional intrinsic space [133]. Therefore, how to capture and exploit the intrinsic manifold structure to compute a more effective distance/similarity measure becomes a key task in many areas [133]. In this work, we consider recent unsupervised manifold learning methods to provide more effective similarity measures using rank-based approaches. Three of them are considered:

- **Log-based Hypergraph of Ranking References (LHRR)** [251]: An algorithm that models the input ranked lists as hypergraphs and exploits the relations between the elements in the dataset.
- **BFS-Tree of Ranking References (BFSTREE)** [253]: It uses a breadth-first tree structure that models the similarity information between the elements in the ranked lists, which is employed with the objective of analyzing the implicit relations between the elements of the dataset. The tree structure allows a representation of the top-k elements such that the weights of the edges are computed based on the correlations among the ranked lists.

- **The Rank-based Diffusion Process with Assured Convergence (RDPAC)** [252]: It performs a diffusion process to exploit the information contained in the ranked lists. It also presents formal proof for the convergence of the diffusion process. The asymptotic complexity of the algorithm is low, which allows its use in many different scenarios with a great number of data elements.

9.2 Experimental Evaluation

This section discusses the experimental evaluation conducted to assess the effectiveness of the proposed Manifold-GCN. Section 9.2.1 discusses the experimental protocol. The semi-supervised image classification results are presented in Section 9.2.2. Section 9.2.4 shows visualizations of feature space improvements, while Section 9.2.5 reports a comparison with both traditional and recent state-of-the-art methods, Section 9.2.6 reports the run-time for each step of the proposed approach.

9.2.1 Experimental Protocol

The Manifold-GCN was evaluated for semi-supervised classification and retrieval. Three public datasets were considered for classification (Flowers [229], Corel5k [194], and CUB200 [346]) and three for retrieval in person re-identification (CUHK03 [176], Market1501 [422], and DukeMTMC [428]). A diverse set of deep features was considered, including CNNs and Vision Transformers. More details about the evaluation measures, datasets, and descriptors can be found in Sections 4.1 and 4.2, respectively.

The proposed Manifold-GCN consists of two steps: manifold learning and semi-supervised classification. For the manifold learning approach, all the data is used for the distance learning process, which is completely unsupervised; no labels are used. In the second step, the semi-supervised classification by the GCN, we perform cross-validation that, in our case, consists of a 10-fold split where one fold is used for training and the rest is used for testing. For each of the 10 executions (one for every fold being considered as training), 90% is considered as testing data (unlabeled data). We highlight that, since we are running 10 executions by changing the folds, every dataset element will be considered as training or test at least once. Therefore, each reported value corresponds to the mean of 50 executions (number of executions multiplied by the number of folds).

Regarding the training and test splits, the same protocol was applied to both our approach and all baselines. Since the GCNs used by the Manifold-GCN are transductive methods, both the training and testing data are considered during training. However, only the training data is labeled. Model predictions are always made on the test set, which is unlabeled. It is important to observe that only 10% of the data is used for training, which creates a challenging semi-supervised setting where the amount of labeled data is significantly smaller than the unlabeled data.

For all the GCNs, the Adam optimizer with a learning rate of 10^{-5} was used, except for Cub200, in which we used a learning rate of 10^{-4}. Regarding the number of neurons, we used 256. The only exceptions are GCN-SGC, which does not have this parameter; and GCN-GAT which has a number of heads, which was set to 32. The training processes consisted of 200 epochs, using input graphs with $k = 40$. In the same way, the manifold learning methods also have a parameter k, which is different from the graph k. For the method k, we also used $k = 40$.

9.2.2 Classification Results

The proposed approach was evaluated on a wide diversity of semi-supervised classification scenarios, considering 3 distinct datasets (Flowers [229], Corel5k [194], and CUB200 [346]). For each dataset, 4 to 5 deep learning features trained on a transfer learning setting were used, considering both CNNs and Vision Transformers approaches. For classification, 5 GCN models are evaluated considering both the traditional and reciprocal kNN graphs. The impact of the re-ranking step is also assessed, evaluating the classification results with and without this step, considering 3 distinct rank-based manifold learning methods. In the semi-supervised scenario, the mean of 5 executions for 10 folds was performed.

Tables 9.1, 9.2 and 9.3 present the results for the datasets Flowers, Corel5k, and CUB200, respectively. The best result for each feature/GCN is highlighted in bold. The gray highlight is used to indicate the best result for the corresponding GCN. The blue color indicates the best result for the dataset (the best result in the whole table).

Some interesting observations can be made from the experimental results. In general, it can be noticed that the reciprocal kNN graph outperforms the traditional kNN graph. It can be observed that the use of manifold learning methods outperforms the scenarios without its use. Moreover, the combination of reciprocal kNN graph and manifold learning methods leads to the best results for all GCN models (gray highlight) and datasets (in blue).

Among the features, VIT-B16 yielded the best results. Therefore, there is a correlation that shows that the better the feature, the better the classification result. In this case, the best feature is VIT-B16. For GCN models and manifold learning methods, the diversity is higher, but GCN-SGC and RDPAC achieved the best results in most of the scenarios. We also can highlight the remarkable gains obtained on all evaluated datasets and features from the original (kNN without re-ranking) to the proposed approach (reciprocal kNN with re-ranking). For CUB200, the most challenging dataset, the accuracy of GCN-APPNP was improved from 55.24% to 75.59%.

Our method was also evaluated considering the weighted F-Measure. Figure 9.2 reports the results for GCN-SGC on the traditional kNN graph (on the left) and the Reciprocal kNN graph (on the right). For every graph, we see that using manifold learning improves the results of the traditional GCN.

Table 9.1 – Impact of manifold learning approaches (LHRR, RDPAC, BFSTREE) and Reciprocal Graph (Rec.) on the classification accuracy (%) of 5 different GCN models on **Flowers dataset**. The best results for each feature and GCN model are highlighted in bold, the best results for each GCN model are marked with a gray background, and the best result for the entire dataset is highlighted in blue. In all the cases, the best results used manifold learning and the Reciprocal Graph.

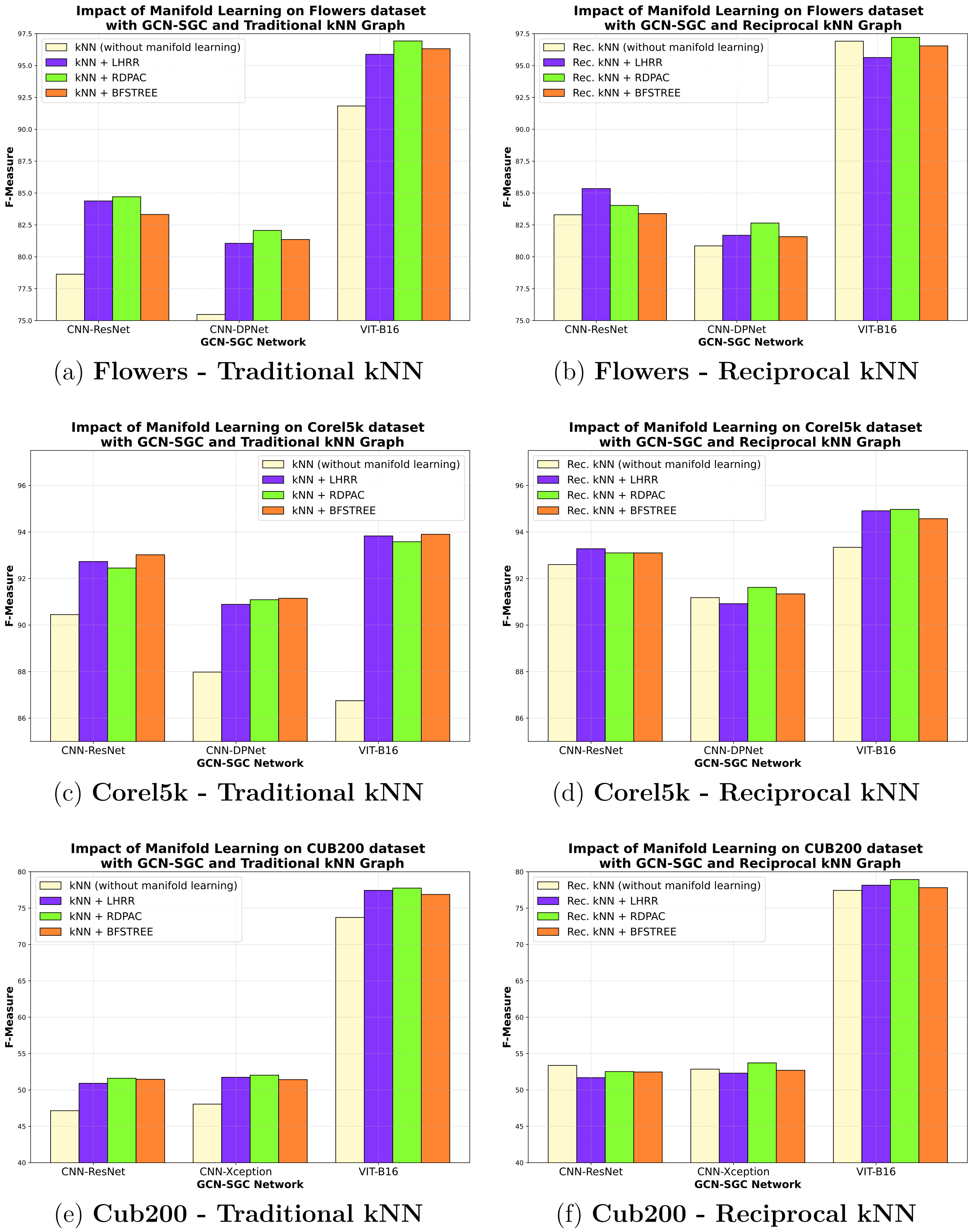
Classifier Specification			**Feature**				
GCN	**Graph**	**Re-Rank**	**CNN-ResNet** [110]	**CNN-DPNet** [51]	**CNN-SENet** [117]	**T2T-VIT24** [399]	**VIT-B16** [77]
GCN-Net	kNN	—	79.08 ± 0.3039	76.94 ± 0.3688	72.72 ± 0.2052	69.75 ± 0.0827	92.72 ± 0.1324
	kNN	LHRR	84.37 ± 0.3239	80.76 ± 0.1372	73.89 ± 0.133	72.03 ± 0.1131	95.88 ± 0.0567
	kNN	RDPAC	83.91 ± 0.1279	81.24 ± 0.2597	74.76 ± 0.2245	74.60 ± 0.1353	96.86 ± 0.0702
	kNN	BFSTREE	83.12 ± 0.1784	81.39 ± 0.1222	74.83 ± 0.1284	72.49 ± 0.3283	96.33 ± 0.0695
	Rec.	—	83.89 ± 0.1973	81.19 ± 0.264	76.23 ± 0.1913	75.82 ± 0.2096	97.07 ± 0.0606
	Rec.	LHRR	**84.67 ± 0.0988**	80.64 ± 0.1749	73.97 ± 0.1383	72.40 ± 0.1927	95.39 ± 0.1583
	Rec.	RDPAC	84.20 ± 0.1975	**82.27 ± 0.1659**	**76.61 ± 0.1968**	**75.87 ± 0.1877**	**97.16 ± 0.0168**
	Rec.	BFSTREE	82.97 ± 0.1623	81.20 ± 0.1141	74.80 ± 0.2034	73.26 ± 0.1008	96.52 ± 0.0538
GCN-SGC	kNN	—	79.64 ± 0.1023	77.09 ± 0.1139	73.00 ± 0.0941	70.05 ± 0.0802	92.84 ± 0.0655
	kNN	LHRR	84.41 ± 0.0835	80.36 ± 0.0661	74.04 ± 0.0599	72.11 ± 0.113	95.85 ± 0.0285
	kNN	RDPAC	84.19 ± 0.0659	81.12 ± 0.0645	75.06 ± 0.0627	75.18 ± 0.0743	96.95 ± 0.0133
	kNN	BFSTREE	83.33 ± 0.0533	81.54 ± 0.0410	75.08 ± 0.0565	72.86 ± 0.0879	96.42 ± 0.0396
	Rec.	—	83.99 ± 0.0478	81.32 ± 0.0314	76.16 ± 0.0415	75.69 ± 0.0828	96.93 ± 0.0464
	Rec.	LHRR	**84.91 ± 0.0665**	80.75 ± 0.0694	74.60 ± 0.0510	72.95 ± 0.0563	95.47 ± 0.0171
	Rec.	RDPAC	84.53 ± 0.0580	**82.53 ± 0.1335**	**76.93 ± 0.0376**	**76.43 ± 0.0499**	**97.11 ± 0.0163**
	Rec.	BFSTREE	83.43 ± 0.0200	81.58 ± 0.1169	75.03 ± 0.0313	73.58 ± 0.026	96.63 ± 0.0337
GCN-GAT	kNN	—	80.67 ± 0.2144	65.60 ± 0.9961	74.64 ± 0.3048	67.33 ± 0.9069	93.65 ± 0.228
	kNN	LHRR	84.52 ± 0.3202	76.15 ± 1.4547	75.48 ± 0.2365	73.37 ± 0.2824	95.33 ± 0.2522
	kNN	RDPAC	84.02 ± 0.1058	77.39 ± 1.2703	75.29 ± 0.3550	75.40 ± 0.4316	97.09 ± 0.0572
	kNN	BFSTREE	83.04 ± 0.1844	77.19 ± 1.833	75.82 ± 0.2086	73.26 ± 0.3184	96.41 ± 0.0465
	Rec.	—	83.67 ± 0.1965	77.42 ± 0.6762	76.74 ± 0.3398	74.82 ± 0.2978	96.99 ± 0.0558
	Rec.	LHRR	**84.82 ± 0.2194**	79.63 ± 0.6337	75.22 ± 0.2648	73.32 ± 0.3684	95.21 ± 0.2575
	Rec.	RDPAC	84.40 ± 0.1488	**79.69 ± 1.0373**	**77.18 ± 0.2940**	**76.90 ± 0.3418**	**97.22 ± 0.0557**
	Rec.	BFSTREE	82.79 ± 0.2926	78.74 ± 0.2682	75.94 ± 0.2681	73.85 ± 0.2632	96.55 ± 0.0881
GCN-APPNP	kNN	—	77.25 ± 0.1692	76.38 ± 0.238	71.0 ± 0.4051	69.45 ± 0.3072	90.24 ± 0.2128
	kNN	LHRR	84.58 ± 0.2621	82.53 ± 0.2443	76.83 ± 0.1622	74.32 ± 0.2989	96.05 ± 0.0421
	kNN	RDPAC	85.35 ± 0.2205	83.32 ± 0.1287	76.89 ± 0.3673	77.87 ± 0.0660	97.28 ± 0.0303
	kNN	BFSTREE	84.22 ± 0.1638	83.34 ± 0.0875	77.94 ± 0.3084	75.65 ± 0.2505	96.73 ± 0.0763
	Rec.	—	83.91 ± 0.1181	82.20 ± 0.2160	77.74 ± 0.1645	77.11 ± 0.1485	97.24 ± 0.0470
	Rec.	LHRR	**85.88 ± 0.1896**	82.55 ± 0.2138	76.60 ± 0.2479	75.40 ± 0.2458	95.68 ± 0.1083
	Rec.	RDPAC	85.41 ± 0.2304	**83.99 ± 0.1276**	**78.82 ± 0.1466**	**78.01 ± 0.1307**	**97.43 ± 0.0699**
	Rec.	BFSTREE	83.75 ± 0.2099	83.14 ± 0.1915	77.83 ± 0.1826	75.85 ± 0.2098	96.89 ± 0.0632
GCN-ARMA	kNN	—	78.69 ± 0.2471	76.01 ± 0.295	73.18 ± 0.4015	70.47 ± 0.2548	91.27 ± 0.1731
	kNN	LHRR	84.64 ± 0.3211	81.90 ± 0.4272	76.09 ± 0.1451	74.26 ± 0.2543	95.66 ± 0.1726
	kNN	RDPAC	85.05 ± 0.1643	82.38 ± 0.3741	76.18 ± 0.3637	76.75 ± 0.2599	96.88 ± 0.0698
	kNN	BFSTREE	83.72 ± 0.0791	81.96 ± 0.3477	76.81 ± 0.1272	75.03 ± 0.1647	96.24 ± 0.0812
	Rec.	—	83.32 ± 0.3713	80.86 ± 0.1282	76.96 ± 0.3041	76.11 ± 0.3851	96.66 ± 0.1140
	Rec.	LHRR	**85.36 ± 0.3818**	82.17 ± 0.3283	75.92 ± 0.2516	74.64 ± 0.3728	95.13 ± 0.2118
	Rec.	RDPAC	84.97 ± 0.2524	**83.14 ± 0.3078**	**77.89 ± 0.2358**	**77.81 ± 0.3271**	**97.02 ± 0.0944**
	Rec.	BFSTREE	84.06 ± 0.2612	82.21 ± 0.1901	76.88 ± 0.1897	75.10 ± 0.3000	96.47 ± 0.1171

Table 9.2 – Impact of manifold learning approaches (LHRR, RDPAC, BFSTREE) and Reciprocal Graph (Rec.) on the classification accuracy (%) of 5 different GCN models on **Corel5k dataset**. The best results for each feature and GCN model are highlighted in bold, the best results for each GCN model are marked with a gray background, and the best result for the entire dataset is highlighted in blue. In all the cases, the best results used manifold learning.

Classifier Specification			**Feature**				
GCN	**Graph**	**Re-Rank**	**CNN-ResNet**	**CNN-DPNet**	**CNN-SENet**	**T2T-VIT24**	**VIT-B16**
			[110]	[51]	[117]	[399]	[77]
GCN-Net	kNN	—	89.34 ± 0.0950	86.49 ± 0.0998	89.17 ± 0.0956	89.02 ± 0.1452	89.93 ± 0.2878
	kNN	LHRR	91.40 ± 0.0906	88.94 ± 0.1958	90.19 ± 0.1392	90.68 ± 0.0957	94.57 ± 0.121
	kNN	RDPAC	91.46 ± 0.1402	89.05 ± 0.1054	90.65 ± 0.0483	91.77 ± 0.1246	94.29 ± 0.139
	kNN	BFSTREE	**92.03 ± 0.1165**	89.28 ± 0.1858	91.19 ± 0.1102	91.78 ± 0.0432	94.30 ± 0.3362
	Rec.	—	91.68 ± 0.1064	**89.62 ± 0.1114**	**91.81 ± 0.1159**	92.19 ± 0.0908	93.42 ± 0.1987
	Rec.	LHRR	91.68 ± 0.0224	88.48 ± 0.1268	90.58 ± 0.0901	91.50 ± 0.0684	94.63 ± 0.139
	Rec.	RDPAC	92.00 ± 0.1434	89.55 ± 0.0944	90.93 ± 0.1654	91.96 ± 0.0705	**94.76 ± 0.1577**
	Rec.	BFSTREE	92.00 ± 0.0954	89.33 ± 0.1221	91.32 ± 0.0833	**92.43 ± 0.0401**	94.39 ± 0.2771
GCN-SGC	kNN	—	89.62 ± 0.0321	86.78 ± 0.0256	89.81 ± 0.0426	88.95 ± 0.0482	93.36 ± 0.0401
	kNN	LHRR	91.19 ± 0.0262	88.74 ± 0.0242	89.90 ± 0.044	90.49 ± 0.0518	95.20 ± 0.0219
	kNN	RDPAC	91.47 ± 0.0216	88.95 ± 0.0632	90.70 ± 0.0403	91.77 ± 0.0521	94.76 ± 0.078
	kNN	BFSTREE	91.98 ± 0.0246	89.23 ± 0.0453	91.40 ± 0.0061	91.71 ± 0.0444	95.26 ± 0.0759
	Rec.	—	91.98 ± 0.0133	89.83 ± 0.0415	**92.15 ± 0.0164**	**92.75 ± 0.0908**	95.49 ± 0.0107
	Rec.	LHRR	91.73 ± 0.0508	88.70 ± 0.0669	90.73 ± 0.0235	91.68 ± 0.0305	95.57 ± 0.017
	Rec.	RDPAC	92.00 ± 0.0247	**89.84 ± 0.1057**	90.85 ± 0.0396	92.31 ± 0.072	95.50 ± 0.020
	Rec.	BFSTREE	**92.04 ± 0.009**	89.49 ± 0.0627	91.30 ± 0.0257	92.54 ± 0.0591	95.30 ± 0.0479
GCN-GAT	kNN	—	90.48 ± 0.1727	83.28 ± 0.33	91.13 ± 0.1107	90.7 ± 0.1187	91.3 ± 0.1764
	kNN	LHRR	92.21 ± 0.1328	88.59 ± 0.4012	91.28 ± 0.2208	92.2 ± 0.0839	94.56 ± 0.1777
	kNN	RDPAC	91.86 ± 0.1403	89.78 ± 0.2723	91.41 ± 0.1429	92.82 ± 0.0956	94.46 ± 0.2555
	kNN	BFSTREE	**92.42 ± 0.1008**	89.61 ± 0.362	91.95 ± 0.1382	93.09 ± 0.1337	94.58 ± 0.2226
	Rec.	—	92.02 ± 0.0917	89.0 ± 0.2638	**92.23 ± 0.0844**	92.81 ± 0.113	93.64 ± 0.2373
	Rec.	LHRR	92.19 ± 0.1057	89.17 ± 0.2074	91.18 ± 0.1451	92.41 ± 0.1456	94.55 ± 0.1918
	Rec.	RDPAC	92.22 ± 0.0858	**90.48 ± 0.1718**	91.48 ± 0.1021	93.02 ± 0.1334	**94.89 ± 0.1492**
	Rec.	BFSTREE	92.30 ± 0.1128	90.01 ± 0.2374	91.88 ± 0.1081	**93.35 ± 0.1537**	94.75 ± 0.1385
GCN-APPNP	kNN	—	89.72 ± 0.2031	87.68 ± 0.0785	89.92 ± 0.0992	89.86 ± 0.0731	86.89 ± 0.2487
	kNN	LHRR	92.6 ± 0.0625	90.81 ± 0.1043	91.49 ± 0.1307	91.83 ± 0.0952	94.53 ± 0.1144
	kNN	RDPAC	92.69 ± 0.1161	90.75 ± 0.179	91.81 ± 0.098	92.58 ± 0.0588	94.12 ± 0.2213
	kNN	BFSTREE	93.04 ± 0.0872	**91.01 ± 0.1026**	92.35 ± 0.0535	92.83 ± 0.031	94.37 ± 0.0855
	Rec.	—	92.69 ± 0.05	90.70 ± 0.1301	**92.79 ± 0.0429**	93.56 ± 0.0669	93.53 ± 0.1042
	Rec.	LHRR	92.88 ± 0.1058	89.99 ± 0.0869	91.78 ± 0.0694	92.63 ± 0.0817	94.95 ± 0.2116
	Rec.	RDPAC	92.82 ± 0.046	90.95 ± 0.1134	91.92 ± 0.0738	93.17 ± 0.0804	**95.13 ± 0.1095**
	Rec.	BFSTREE	**93.08 ± 0.0727**	90.78 ± 0.1317	92.39 ± 0.0269	**93.70 ± 0.0653**	94.72 ± 0.1564
GCN-ARMA	kNN	—	88.58 ± 0.312	86.47 ± 0.0729	89.11 ± 0.1061	89.16 ± 0.0571	85.48 ± 0.3945
	kNN	LHRR	91.58 ± 0.1185	89.84 ± 0.1565	90.98 ± 0.1738	91.46 ± 0.0963	90.66 ± 0.5051
	kNN	RDPAC	91.72 ± 0.1775	90.09 ± 0.2858	91.08 ± 0.1226	92.28 ± 0.0545	92.62 ± 0.4067
	kNN	BFSTREE	92.23 ± 0.1447	90.31 ± 0.1195	91.7 ± 0.0869	92.24 ± 0.0682	92.28 ± 0.3061
	Rec.	—	91.14 ± 0.137	89.24 ± 0.2139	91.31 ± 0.1887	91.84 ± 0.0774	90.48 ± 0.1707
	Rec.	LHRR	91.77 ± 0.1541	89.24 ± 0.1428	91.07 ± 0.126	91.78 ± 0.1145	92.39 ± 0.2078
	Rec.	RDPAC	92.05 ± 0.1403	**90.41 ± 0.1645**	91.47 ± 0.1202	92.49 ± 0.2056	92.80 ± 0.1896
	Rec.	BFSTREE	**92.27 ± 0.0377**	90.14 ± 0.1897	**91.71 ± 0.1753**	**92.90 ± 0.1446**	**92.74 ± 0.2083**

Table 9.3 – Impact of manifold learning approaches (LHRR, RDPAC, BFSTREE) and Reciprocal Graph (Rec.) on the classification accuracy (%) of 5 different GCN models on **CUB200 dataset**. The best results for each feature and GCN model are highlighted in bold, the best results for each GCN model are marked with a gray background, and the best result for the entire dataset is highlighted in blue. In all the cases, the best results used manifold learning.

Classifier Specification			**Feature**			
GCN	**Graph**	**Re-Rank**	**CNN-ResNet** [110]	**CNN-SENet** [51]	**CNN-Xception** [52]	**VIT-B16** [77]
GCN-Net	kNN	—	40.76 ± 0.7467	35.8 ± 0.0634	46.66 ± 0.019	64.39 ± 0.4486
	kNN	LHRR	49.16 ± 0.3119	36.17 ± 0.1153	51.13 ± 0.0738	70.42 ± 0.671
	kNN	RDPAC	49.44 ± 0.1092	36.84 ± 0.0578	51.18 ± 0.0284	72.71 ± 0.1506
	kNN	BFSTREE	49.18 ± 0.1011	37.10 ± 0.0482	50.62 ± 0.0639	71.54 ± 0.1888
	Rec.	—	49.46 ± 0.3279	**39.42 ± 0.113**	50.76 ± 0.0713	68.85 ± 0.3055
	Rec.	LHRR	51.23 ± 0.0788	36.5 ± 0.0728	51.92 ± 0.0546	73.49 ± 0.1879
	Rec.	RDPAC	**51.57 ± 0.0999**	38.57 ± 0.0712	**53.12 ± 0.0596**	**74.39 ± 0.3061**
	Rec.	BFSTREE	50.80 ± 0.0291	37.8 ± 0.0538	51.82 ± 0.0658	73.58 ± 0.3939
GCN-SGC	kNN	—	47.55 ± 0.0329	36.48 ± 0.0684	48.60 ± 0.0072	74.23 ± 0.0385
	kNN	LHRR	51.22 ± 0.0184	35.88 ± 0.0137	52.36 ± 0.0125	77.84 ± 0.0519
	kNN	RDPAC	51.88 ± 0.0315	37.75 ± 0.0148	52.98 ± 0.0103	78.16 ± 0.0453
	kNN	BFSTREE	51.66 ± 0.016	37.70 ± 0.01	52.21 ± 0.0095	77.31 ± 0.0563
	Rec.	—	53.71 ± 0.0362	**40.31 ± 0.0255**	54.0 ± 0.0054	78.03 ± 0.0428
	Rec.	LHRR	51.99 ± 0.0251	36.74 ± 0.0162	53.12 ± 0.0153	78.54 ± 0.0177
	Rec.	RDPAC	**52.85 ± 0.0164**	38.91 ± 0.0073	**54.59 ± 0.0036**	**79.27 ± 0.0325**
	Rec.	BFSTREE	52.68 ± 0.0308	38.65 ± 0.023	53.54 ± 0.0041	78.12 ± 0.0344
GCN-GAT	kNN	—	41.84 ± 0.2901	32.5 ± 0.205	42.45 ± 0.1848	59.53 ± 0.5668
	kNN	LHRR	48.86 ± 0.1593	34.78 ± 0.1155	48.8 ± 0.246	64.02 ± 0.4082
	kNN	RDPAC	49.05 ± 0.1145	35.9 ± 0.1158	49.03 ± 0.1037	68.78 ± 0.2495
	kNN	BFSTREE	48.77 ± 0.1427	35.98 ± 0.1457	48.3 ± 0.1084	68.1 ± 0.2488
	Rec.	—	45.46 ± 0.1879	33.02 ± 0.1206	45.88 ± 0.16	64.82 ± 0.2582
	Rec.	LHRR	50.19 ± 0.0904	35.28 ± 0.1364	50.17 ± 0.1073	70.31 ± 0.0762
	Rec.	RDPAC	**50.95 ± 0.0632**	**37.55 ± 0.1087**	**51.29 ± 0.1577**	**72.94 ± 0.1716**
	Rec.	BFSTREE	49.89 ± 0.1871	36.67 ± 0.129	49.87 ± 0.1245	71.73 ± 0.1775
GCN-APPNP	kNN	—	29.16 ± 0.6867	30.27 ± 0.3694	42.68 ± 0.0826	55.24 ± 0.5689
	kNN	LHRR	47.0 ± 0.1836	34.91 ± 0.1598	48.77 ± 0.0979	66.57 ± 0.572
	kNN	RDPAC	47.19 ± 0.0701	35.29 ± 0.1195	47.72 ± 0.094	69.92 ± 0.2262
	kNN	BFSTREE	46.59 ± 0.2154	35.28 ± 0.0718	47.14 ± 0.0895	70.86 ± 0.2702
	Rec.	—	48.51 ± 0.1192	38.02 ± 0.0461	47.51 ± 0.0452	68.29 ± 0.0935
	Rec.	LHRR	**51.99 ± 0.0800**	37.45 ± 0.0768	51.43 ± 0.084	74.61 ± 0.0991
	Rec.	RDPAC	51.82 ± 0.1028	**39.15 ± 0.1601**	**52.17 ± 0.0865**	**75.59 ± 0.2139**
	Rec.	BFSTREE	50.6 ± 0.0848	38.21 ± 0.0358	50.26 ± 0.1301	74.15 ± 0.1837
GCN-ARMA	kNN	—	38.74 ± 0.4527	32.96 ± 0.1626	42.91 ± 0.1465	60.26 ± 0.4398
	kNN	LHRR	47.58 ± 0.2387	34.56 ± 0.0799	49.26 ± 0.2191	67.21 ± 0.2825
	kNN	RDPAC	47.77 ± 0.2075	35.4 ± 0.1474	49.88 ± 0.1479	71.16 ± 0.2337
	kNN	BFSTREE	47.12 ± 0.3126	35.6 ± 0.1385	48.78 ± 0.0991	70.13 ± 0.4433
	Rec.	—	44.37 ± 0.1739	34.25 ± 0.1559	46.95 ± 0.3062	64.55 ± 0.3184
	Rec.	LHRR	49.29 ± 0.0987	35.22 ± 0.0891	50.38 ± 0.1318	70.05 ± 0.653
	Rec.	RDPAC	**49.81 ± 0.2090**	**37.12 ± 0.1276**	**51.63 ± 0.1155**	**73.29 ± 0.34**
	Rec.	BFSTREE	48.92 ± 0.2721	36.38 ± 0.1331	50.41 ± 0.0777	72.17 ± 0.3336



(a) **Flowers - Traditional kNN**

(b) **Flowers - Reciprocal kNN**

(c) **Corel5k - Traditional kNN**

(d) **Corel5k - Reciprocal kNN**

(e) **Cub200 - Traditional kNN**

(f) **Cub200 - Reciprocal kNN**

Figure 9.2 – Impact of manifold learning approaches on F-measure results considering GCN-SGC on different datasets and features.

9.2.3 Person Re-ID Results

In view of the promising results obtained, the method was also evaluated for person re-identification. The Re-ID results consider only the GCN-SGC, once this GCN is the one that presented the best results in the majority of the experiments of the general purpose datasets (Flowers, Corel5k, and CUB200). For training, 200 epochs were used and a learning rate of 10^{-5}. The features were pre-processed with PCA to obtain feature vectors with 10 positions that are provided as input for the GCN network. The ranked lists were obtained from the BallTree method with Euclidean distance for the embeddings obtained as the output of the GCN-SGC. All the ranked lists have the same size (3000 positions).

Tables 9.4, 9.5, and 9.6 present the results for the datasets CUHK03 [176], Market1501 [422] and DukeMTMC [428], respectively. The measures MAP and R1 were used, once they are commonly used in Re-ID literature. The default dataset protocol was used in all the cases. The reported results represent the mean and standard deviation of 5 executions considering the training set (*train*), test (*gallery*), and query (*probe*) proposed by the authors of the datasets. Once again, it is noticeable that the reciprocal kNN graph in combination with re-ranking methods produced the best results, which evince the effectiveness of our proposed approach.

Table 9.4 – Results (%) for GCN-SGC on CUHK03 dataset.

Classifier Specification		**CNN-ResNet** [110]		**OSNET-AIN** [436]	
Graph	**Re-Rank**	**MAP**	**R1**	**MAP**	**R1**
kNN	—	12.08 ± 0.0390	12.60 ± 0.2819	23.18 ± 0.0544	24.07 ± 0.3397
kNN	LHRR	18.59 ± 0.0364	18.46 ± 0.1093	34.25 ± 0.0527	33.58 ± 0.0962
kNN	BFSTREE	18.54 ± 0.0907	18.40 ± 0.1994	34.83 ± 0.0662	34.03 ± 0.2731
kNN	RDPAC	18.05 ± 0.0861	17.63 ± 0.1418	33.62 ± 0.0799	32.87 ± 0.0793
Rec.	—	14.67 ± 0.0573	14.71 ± 0.1664	26.77 ± 0.1767	27.22 ± 0.0953
Rec.	LHRR	19.02 ± 0.0376	18.68 ± 0.1328	35.95 ± 0.0519	**35.19 ± 0.1743**
Rec.	BFSTREE	18.97 ± 0.0814	18.89 ± 0.1552	35.58 ± 0.1056	34.81 ± 0.1622
Rec.	RDPAC	**19.58 ± 0.0781**	**19.13 ± 0.1600**	**35.99 ± 0.0703**	35.00 ± 0.1744

Table 9.5 – Results (%) on Market1501 dataset.

Classifier Specification		**CNN-ResNet** [110]		**OSNET-AIN** [436]	
Graph	**Re-Rank**	**MAP**	**R1**	**MAP**	**R1**
kNN	—	19.78 ± 0.0724	37.4 ± 0.2341	40.8 ± 0.1112	57.43 ± 0.0893
kNN	LHRR	33.26 ± 0.0433	48.05 ± 0.1069	56.56 ± 0.0786	69.28 ± 0.0356
kNN	BFSTREE	33.21 ± 0.0741	50.02 ± 0.1293	56.02 ± 0.0595	69.82 ± 0.0482
kNN	RDPAC	32.10 ± 0.0671	47.99 ± 0.0869	54.83 ± 0.055	67.95 ± 0.0742
Rec.	—	27.03 ± 0.0813	44.43 ± 0.1611	47.54 ± 0.0994	63.43 ± 0.1197
Rec.	LHRR	**34.16 ± 0.0737**	49.31 ± 0.1197	57.37 ± 0.0836	69.41 ± 0.1105
Rec.	BFSTREE	34.05 ± 0.0651	**50.5 ± 0.0983**	**57.48 ± 0.0869**	**70.30 ± 0.0827**
Rec.	RDPAC	33.6 ± 0.0317	49.07 ± 0.0403	56.83 ± 0.054	69.39 ± 0.0774

Table 9.6 – Results (%) for GCN-SGC on DukeMTMC dataset.

Classifier Specification		**CNN-ResNet** [110]		**OSNET-AIN** [436]	
Graph	**Re-Rank**	**MAP**	**R1**	**MAP**	**R1**
kNN	—	30.66 ± 0.0698	49.41 ± 0.0523	52.68 ± 0.0909	67.72 ± 0.2399
kNN	LHRR	47.81 ± 0.0565	58.29 ± 0.1408	64.09 ± 0.0482	72.89 ± 0.1502
kNN	BFSTREE	46.06 ± 0.0386	57.09 ± 0.158	62.21 ± 0.0409	72.13 ± 0.1502
kNN	RDPAC	45.41 ± 0.0918	56.84 ± 0.0832	61.54 ± 0.0369	71.01 ± 0.1218
Rec.	—	38.89 ± 0.0806	52.85 ± 0.2548	57.67 ± 0.0435	69.98 ± 0.066
Rec.	LHRR	**48.69 ± 0.0539**	**59.37 ± 0.1402**	65.61 ± 0.0536	**74.22 ± 0.1085**
Rec.	BFSTREE	48.15 ± 0.0804	58.7 ± 0.1459	65.66 ± 0.0514	73.82 ± 0.2246
Rec.	RDPAC	48.54 ± 0.0409	58.84 ± 0.1170	**65.83 ± 0.0372**	73.67 ± 0.1121

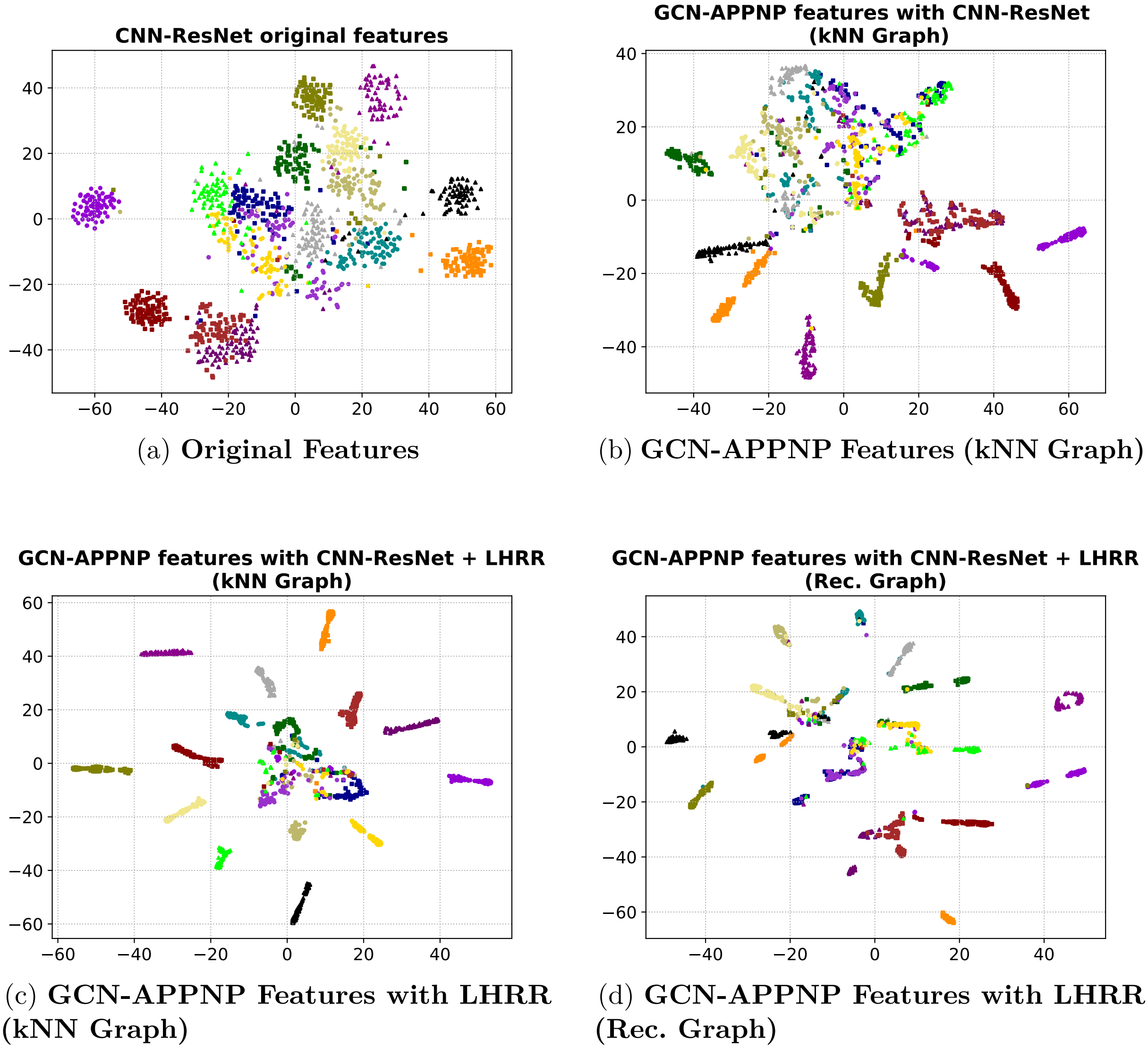
9.2.4 Visualization Results

In order to visualize the effectiveness of our approach, an experiment was conducted showing the distribution of features in a 2D space, after being processed by t-Distributed Stochastic Neighbor Embedding (t-SNE) [214]. Figure 9.3 shows the results for (a) the original CNN-ResNet features; (b) the GCN output with kNN graph; (c) the GCN output with kNN graph and manifold learning; (d) the GCN output with Reciprocal graph and manifold learning. The Flowers dataset was chosen for this visualization due to the small number of classes, which makes it easier to visualize the improvements. Each class is represented by a different combination of shape and color. Notice that the distribution of classes is further improved when the Manifold-GCN is applied (c and d), which is consistent with our main hypothesis.

9.2.5 Comparison with Other Approaches

For comparison purposes, a wide variety of supervised and semi-supervised classification approaches were considered, both traditional and more recent ones. A brief description of the employed baselines, the implementations, and the parameters used are presented in the following:

- *k* **Nearest Neighbor (kNN)**: A traditional approach that computes the distance to the other elements in the dataset and selects the k closest ones. The sklearn implementation was used, with $k = 20$.
- **Support Vector Machine (SVM)** [54]: It is a traditional method that consists of finding the hyperplane that best separates the data into the correct classes in a high-dimensional space. The sklearn implementation was used, with default parameters and a Radial Basis Function (RBF) kernel.
- **Single-Layer and Multi-Layer Perceptron**: The sklearn implementation was used for both, with the Stochastic Gradient Descent (SGD) optimizer.



(a) **Original Features**

(b) **GCN-APPNP Features (kNN Graph)**

(c) **GCN-APPNP Features with LHRR (kNN Graph)**

(d) **GCN-APPNP Features with LHRR (Rec. Graph)**

Figure 9.3 – t-SNE visualizations that show the feature space improvement when manifold learning and reciprocal graph were applied. Experiments were conducted on the Flowers dataset and CNN-ResNet features. Each class is represented by a different shape and color.

- **Optimum-Path Forest (OPF)** [236, 8]: It builds a graph where each node is an element of the dataset and the edges are weighted by their Euclidean distance. The algorithm computes the optimum path between the nodes in order to classify them into a given class. The pyOPF [10] implementation was used, with the default parameters.

- **Pseudo-label** [162]: The method is semi-supervised and is used to assign labels to unlabeled data. In this work, a public implementation [11] was used along with the Logistic Regression classifier that employed the Stochastic Gradient Descent (SGD) optimizer and Squared Hinge loss with $\alpha = 10^{-5}$ for training.

[10] https://github.com/marcoscleison/PyOPF

[11] https://github.com/anirudhshenoy/pseudo_labeling_small_datasets

- **Label Spreading (LS)** [433]: A semi-supervised algorithm that attributes labels to elements according to the labels of their neighbors, given a certain degree of similarity. For this process, it uses an affinity matrix based on a normalized graph Laplacian. The sklearn implementation was used, considering a Radial Basis Function (RBF) kernel with $\alpha = 0.4125$, $\gamma = 0.1$, and a maximum of 100 iterations. This method is used to expand the training set and is used along with the other classifiers.

- **Learning Discrete Structures for Graph Neural Networks (GNN-LDS and GNN-KNN-LDS)** [90]: This Graph Neural Network (GNN) learns both a graph and embeddings from the input features. It approximately solves a bilevel program that learns a discrete probability distribution on the edges of the graph. The authors claim that this is the first method that simultaneously learns the graph and the parameters of a GNN for semi-supervised classification. The approach presents two variants: (*i*) GNN-LDS; and (*ii*) GNN-KNN-LDS which initializes by computing a kNN graph. Both were used as baselines with their default parameters proposed in the implementation [12] provided by the original authors. For the kNN graph, $k = 20$ was used.

- **Weakly Supervised Framework Experiments Framework (WSEF)** [264]: The method generates pseudo-labels by applying different rank correlation measures (e.g., Jaccard, Spearman). The approach is mainly based on the idea that elements that have ranked lists with a high intersection with others probably belong to the same class. The implementation [13] provided by the authors was used considering Rank Biased Overlap (RBO) [358] correlation measure, $k = 40$ in combination with SVM.

- **CoMatch** [169]: The method is based on concepts of graph-based self-supervised learning. The approach is trained to produce similar embeddings for the same image with different augmentations. CoMatch jointly optimizes three losses: (*i*) a supervised classification loss on labeled data, (*ii*) an unsupervised classification loss on unlabeled data, and (*iii*) a graph-based contrastive loss on unlabeled data. It takes images as input instead of features. This version employs ResNet [110] as the backbone. We considered the implementation provided by the authors [14], with default parameters (the ones used for ImageNet [70] in their code). We trained with a batch size of 25 and 400 epochs for all datasets. Except for the CUB200 dataset, which is larger, we used a batch size of 50 and 300 epochs.

We also compared our results with three CNN-based classifiers, considering image data as input. The images were provided with a size of 100x100 pixels in batches of

[12] `https://github.com/lucfra/LDS-GNN`

[13] `https://github.com/UDLF/WSEF`

[14] `https://github.com/salesforce/CoMatch`

size 32. For the other methods, the input consists of feature vectors obtained from deep features trained through transfer learning. Notice that the CNNs used as baselines require more labeled data in comparison to other methods and were evaluated on a supervised cross-validation scenario (9 folds for training, 1 fold for testing). Except for CNN classifiers, all other methods were evaluated on semi-supervised scenarios (1 fold for training, 9 folds for testing).

Table 9.7 presents the comparison with both traditional and recent state-of-the-art baselines in relation to our approach on Flowers, Corel5k, and CUB200 datasets. Most results are the mean of 5 executions of 10 folds, with some exceptions which are indicated in italic text. Some methods require long running times on larger datasets (i.e., LDS and CoMatch). For GNN-KNN-LDS, KNN-LDS, and CoMatch, the results on CUB200 correspond to 1 execution. For CoMatch, the mean of 3 executions is reported for the Corel5k dataset. The best result for each feature is highlighted in bold and the best for each dataset is highlighted in red. The gray rows indicate the results that correspond to our method.

The proposed method revealed superior results compared to the baselines in most of the cases. The only exception is Flowers with VIT-B16 features where WSEF shows the best results (97.82% accuracy). However, our Manifold-GCN is very close with 97.43% accuracy.

9.2.6 Efficiency Results

We conducted an experiment to measure the run-time (in seconds) for running each of the manifold learning methods and GCN models. The experiments were executed on a machine with an Intel(R) Core(TM) i7-10700F CPU @ 2.90GHz, 32 GB RAM, NVIDIA GeForce RTX 3060 GPU with 12GB VRAM running Ubuntu 20.04 with Linux kernel 5.15.0-52-generic. Table 9.8 reports the average and standard deviation of 5 executions of 10 folds on each dataset and for the two types of graphs (kNN and Reciprocal kNN).

Manifold Learning (M.L.) performs the pre-processing of the GCN graph. Since these methods are not currently parallelized, they run all on the CPU. Parallelization of these approaches is out of the scope of this work, but rank-based methods can be parallelized with data parallelism as shown in other papers [335, 332]. While the training involves both the GCN initialization and the learning process, testing is responsible for computing the classification of all the queries. Both training and testing are performed on the GPU.

Notice that the execution times are very low, which indicates that the method is fast even for the more robust GCNs. Also, most compared methods have a costly training process. An example is CoMatch (2021) which requires huge training times: 40 minutes on

Flowers; 88.4 minutes on Corel5k; 260 minutes on CUB200. These times are the average of executions for 100 epochs. However, CoMatch is generally recommended to be trained for 400 epochs. These values are much higher than our proposed approach.

Table 9.7 – Accuracy comparison (%) for baselines on Flowers, Corel5k, and CUB200 datasets. For every dataset, we compared our approach with both supervised and semi-supervised baselines. The methods are compared with different input features. The results of our method are highlighted with a gray background; the best results for each pair of features and dataset are marked in bold, and the best for each dataset are in red.

Method	**Year**	**Input**	**Training Split**	**Flowers**	**Corel5k**	**CUB200**
MobileNet	2017	**Images**	Supervised	86.66	90.90	35.20
ResNet50	2015			85.97	91.52	31.10
CNN-Xception	2016			90.24	93.32	44.25
CoMatch	2021	**Images**	Semi-Supervised	82.55	*85.70*	*38.29*
kNN	—	**ResNet Features**	Semi-Supervised	63.67	76.80	36.67
SVM	1995			80.54	88.73	48.84
OPF	2009			71.77	83.56	38.59
SL-Perceptron	—			75.44	83.56	39.91
ML-Perceptron	—			78.88	87.10	32.24
PseudoLabel+SGD	2013			82.69	89.76	21.67
LS+kNN	2004			73.49	83.98	36.99
LS+SVM	2004			73.53	83.26	38.70
LS+OPF	2004			72.66	82.32	39.28
LS+SL-Perceptron	2004			72.34	82.38	39.21
LS+ML-Perceptron	2004			73.03	82.53	39.68
GNN-LDS	2019			54.98	62.69	—
GNN-KNN-LDS	2019			79.32	88.94	*37.78*
WSEF+SVM+RBO	2021			85.12	91.68	52.17
SGC+Rec.+RDPAC	Ours			84.53	92.00	**52.85**
Manifold-GCN (best result)	Ours			**85.88**	**93.08**	**52.85**
kNN	—	**SENet Features**	Semi-Supervised	48.71	58.78	22.23
SVM	1995			73.30	85.89	35.32
OPF	2009			64.00	81.33	30.94
SL-Perceptron	—			71.84	82.28	36.39
ML-Perceptron	—			72.62	86.90	32.15
PseudoLabel+SGD	2013			76.87	89.85	20.96
LS+kNN	2004			58.05	72.16	20.00
LS+SVM	2004			59.84	72.79	24.82
LS+OPF	2004			59.25	72.20	25.38
LS+SL-Perceptron	2004			59.27	72.19	25.41
LS+ML-Perceptron	2004			59.39	72.24	25.72
GNN-LDS	2019			52.24	65.80	—
GNN-KNN-LDS	2019			73.69	89.95	—
WSEF+SVM+RBO	2021			76.16	89.74	36.49
SGC+Rec.+RDPAC	Ours			76.93	90.85	38.91
Manifold-GCN (best result)	Ours			**78.82**	**92.79**	**40.31**
kNN	—	**VIT-B16 Features**	Semi-Supervised	91.91	81.19	56.62
SVM	1995			96.75	91.92	75.61
OPF	2009			96.50	90.02	73.27
SL-Perceptron	—			75.79	82.15	70.84
ML-Perceptron	—			92.59	74.41	12.02
PseudoLabel+SGD	2013			96.84	89.07	30.19
LS+kNN	2004			95.74	89.63	66.15
LS+SVM	2004			94.49	87.59	66.81
LS+OPF	2004			94.22	86.14	66.68
LS+SL-Perceptron	2004			93.71	86.31	65.45
LS+ML-Perceptron	2004			95.13	87.68	62.81
GNN-LDS	2019			72.03	56.33	*22.75*
GNN-KNN-LDS	2019			96.66	88.56	*52.42*
WSEF+SVM+RBO	2021			**97.82**	94.00	78.64
SGC+Rec.+RDPAC	Ours			97.11	95.50	**79.27**
Manifold-GCN (best result)	Ours			97.43	**95.57**	**79.27**

Table 9.8 – Execution time (in seconds) for manifold learning methods and GCN approaches for both training and testing.

		Flowers	**Corel5k**	**CUB200**
M.L.	LHRR	1.10 ± 0.0012	6.21 ± 0.0017	20.16 ± 0.0108
	RDPAC	4.66 ± 0.0195	41.74 ± 0.0158	104.18 ± 0.5091
	BFSTREE	9.34 ± 0.0046	37.94 ± 0.1712	95.09 ± 0.0704
Train	GCN-Net (kNN)	0.76 ± 0.0187	2.23 ± 0.0178	6.95 ± 0.0141
	GCN-Net (Rec.)	0.61 ± 0.0016	1.57 ± 0.0018	4.40 ± 0.0003
	GCN-SGC (kNN)	0.15 ± 0.0005	0.20 ± 0.0011	0.54 ± 0.0006
	GCN-SGC (Rec.)	0.14 ± 0.0003	0.19 ± 0.0022	0.51 ± 0.0002
	GCN-GAT (kNN)	3.41 ± 0.0021	11.89 ± 0.0031	30.62 ± 0.0095
	GCN-GAT (Rec.)	2.44 ± 0.0025	7.90 ± 0.0029	19.35 ± 0.0043
	GCN-APPNP (kNN)	0.77 ± 0.0088	3.49 ± 0.0032	27.95 ± 0.0025
	GCN-APPNP (Rec.)	0.75 ± 0.0002	2.44 ± 0.0032	17.11 ± 0.0032
	GCN-ARMA (kNN)	3.84 ± 0.0064	14.45 ± 0.0215	47.8 ± 0.0146
	GCN-ARMA (Rec.)	2.80 ± 0.0051	9.94 ± 0.0013	30.51 ± 0.0113
Test	GCN-Net (kNN)	0.06 ± 0.0366	0.18 ± 0.0382	0.40 ± 0.0327
	GCN-Net (Rec.)	0.05 ± 0.0013	0.18 ± 0.002	0.44 ± 0.0029
	GCN-SGC (kNN)	0.04 ± 0.0015	0.16 ± 0.0011	0.38 ± 0.0051
	GCN-SGC (Rec.)	0.05 ± 0.0015	0.18 ± 0.0018	0.44 ± 0.0005
	GCN-GAT (kNN)	0.04 ± 0.001	0.15 ± 0.0009	0.38 ± 0.004
	GCN-GAT (Rec.)	0.05 ± 0.0015	0.18 ± 0.002	0.44 ± 0.0032
	GCN-APPNP (kNN)	0.05 ± 0.0015	0.15 ± 0.0008	0.38 ± 0.0045
	GCN-APPNP (Rec.)	0.05 ± 0.0015	0.18 ± 0.0021	0.44 ± 0.0026
	GCN-ARMA (kNN)	0.04 ± 0.0015	0.15 ± 0.0008	0.39 ± 0.0036
	GCN-ARMA (Rec.)	0.04 ± 0.0018	0.18 ± 0.0023	0.44 ± 0.0004

10 Contextual Contrastive Loss (CCL)

Machine learning models heavily rely on loss functions, which assume a fundamental role in optimization steps by defining a quantitative measure for prediction errors and guiding the learning process. For classification, the cross-entropy loss is the most commonly used metric for training in supervised learning scenarios [143]. The idea behind cross-entropy loss is to quantify the difference between probability distributions. Despite its widespread use, it exhibits limitations, particularly in its ability to generalize effectively to new, unseen data. It also struggles with issues like class imbalance, noisy labels [418, 296], and the potential for poor margins [84, 199].

Metric learning and contrastive learning were proposed as solutions to the limitations of cross-entropy loss by focusing on learning effective feature representations that emphasize the relationships and distances between data points, rather than merely categorizing individual examples [143, 47]. Metric learning focuses on learning a distance function over pairs of objects. This distance function aims to quantify how similar or dissimilar these objects are to each other. The primary goal is to ensure that similar objects are closer together while dissimilar objects are farther apart in the learned metric space [143, 47].

One of the most well-known methods for self-supervised contrastive learning is the Simultaneous Contrastive Learning of Representations [47] (SimCLR), which is a pioneer in the field. However, since it does not consider labeled data, the Supervised Contrastive Learning [143] (SupCon) was proposed, which can be seen as a supervised version of SimCLR. Although significant progress has been made with contrastive losses, these methods rely solely on comparing the similarity between pairs of embeddings, ignoring contextual information.

In this research, the concept of *contextual information* refers to the process of exploiting the neighboring elements of a data sample to compute more semantically meaningful similarity measures. Some works exploit neighborhood analysis for different purposes, showing the relevance of this information in the context of learning. The Simple Siamese (SimSiam) [49] is compared to SimCLR by employing a kNN classifier on their latent features. The Adaptive Neighborhood Metric Learning (ANML) [292] identifies and removes inseparable similar and dissimilar samples in the training procedure. There is also an example of application [311] that integrates nearest-neighbor hyperparameters with triplet learning to enhance classification performance through a local-margin triplet loss and local mining strategy. Another approach employs neighborhood information in graphs to regularize learning [136], but without using a contrastive loss. However, few methods directly integrate contextual similarity information into the contrastive loss [441, 82, 183].

In this chapter, a novel loss function is proposed, the Contextual Contrastive Loss (CCL), based on the supervised contrastive loss [143, 47] and contextual information, successfully exploited for image retrieval [250, 245]. The proposed CCL improves the learned similarity by taking advantage of contextual neighborhood information for comparing elements during the training process. Among the main contributions, we can mention: *(i)* A novel loss is proposed, named Contextual Contrastive Loss (CCL), based on the supervised contrastive loss [143, 47] and contextual information [250, 245]; *(ii)* Different from other methods that demand constant feature updates, ours only requires updates once per epoch, utilizing those created during each iteration, causing no significant overhead during training; *(iii)* The neighborhood sets are computed once and do not need to be recomputed during the training process; *(iv)* A dynamic neighborhood size is proposed to initially enforce the regrouping of larger regions in space, and then progressively focuses on fine-grained regions as the training progresses, which smooths convergence; *(v)* Results reveal superior results compared to the original contrastive loss [143, 47] on image classification datasets, especially in cases where there are few labeled data and a smaller number of epochs, which shows the potential of our approach in resource-constrained scenarios.

Despite sharing some similarities with the proposed CCL, the kNN Contrastive Loss [441] is distinctly different: *(i)*: It is designed for classification in dialogue systems, specifically considering out-of-domain (OOD) samples, as opposed to image classification; *(ii)*: The kNN Contrastive Loss computes the average contrastive loss for an element and its k neighbors. It iterates for the k neighbors before the contrastive loss logarithmic function. In contrast, our loss formulation is notably different, replacing the similarity function with the square of three components and featuring symmetry; *(iii)*: The methodologies diverge in managing neighborhood lists and features. Our method requires only occasional updates of certain features once per epoch and does not necessitate updating the neighborhood set throughout the training process.

The remaining of this chapter is organized as follows: Section 10.1 discusses the Supervised Contrastive loss (SupCon) [143] and its main formulations, which the CCL is based on. Section 10.2 describes the CCL approach and its steps. Section 10.3 presents the experimental evaluation and results.

10.1 Supervised Contrastive Loss

In this section, we describe the supervised contrastive loss proposed by [143], which is used as the inspiration for our approach. This loss is an extension of the self-supervised [47] batch contrastive approaches to a fully supervised setting. This extension allows the model to use label information more effectively. The general idea involves grouping data samples that belong to the same class closer together in the embedding space while pushing apart groups of samples from different classes. The objective is to enhance the model's ability to distinguish between different classes based on the learned representations (features).

The learning process consists of the use of batches, which contain pairs of images. For each image, two augmentations (i.e., views) are generated. Given a set of N_b randomly sampled sample/label pairs, $\{\boldsymbol{x}_k, \boldsymbol{y}_k\}_{k=1 \ldots N_b}$, the corresponding batch used for training consists of $2N_b$ pairs, $\{\tilde{\boldsymbol{x}}_\ell, \tilde{\boldsymbol{y}}_\ell\}_{\ell=1 \ldots 2N_b}$, where $\tilde{\boldsymbol{x}}_{2k}$ and $\tilde{\boldsymbol{x}}_{2k-1}$ are two random augmentations of $\boldsymbol{x}_k$ $(k = 1 \ldots N_b)$ and $\tilde{\boldsymbol{y}}_{2k-1} = \tilde{\boldsymbol{y}}_{2k} = \boldsymbol{y}_k$. In this work, we consider only multiviewed batches (size $2N_b$), which present two augmentations for each image.

For a multiviewed batch, let $i \in I \equiv \{1 \ldots 2N_b\}$ be the index of an arbitrary augmented sample, and let $j(i)$ be the index of the other augmented sample originating from the same source sample. The set of indices of all positives in a multiviewed batch distinct from i is defined by Equation 10.1 and $|P(i)|$ is its cardinality.

$$P(i) \equiv p \in A(i) : \tilde{\boldsymbol{y}}_p = \tilde{\boldsymbol{y}}_i, \tag{10.1}$$

$A(i)$ refers to the set of all elements in the batch except the image i called the anchor.

Based on these definitions, the work of [143] presents two different supervised contrastive losses, presented by Equations 10.2 and 10.3, respectively.

$$\mathcal{L}_{\text{out}}^{\text{sup}} = \sum_{i \in I} \mathcal{L}_{\text{out}\ ,i}^{\text{sup}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \psi\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \psi\right)} \tag{10.2}$$

$$\mathcal{L}_{\text{in}}^{\text{sup}} = \sum_{i \in I} \mathcal{L}_{\text{in}\ ,i}^{\text{sup}} = \sum_{i \in I} - \log \left\{ \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \psi\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \psi\right)} \right\} \tag{10.3}$$

Here, $\boldsymbol{z}_i$ is the embedding generated by the model during the learning process for the data sample i. The index i is called the anchor. The similarity of embeddings is computed using the dot product operation. The scalar parameter $\psi \in \mathbb{R}^+$, known as temperature, controls how tightly or loosely the model should group embeddings of the same class versus those of different classes.

However, besides similar, the two loss functions (Equations 10.2 and 10.3) are not equivalent. In $\mathcal{L}_{\text{in}}^{\text{sup}}$, the summation of positives is located inside of the log, while in $\mathcal{L}_{\text{out}}^{\text{sup}}$, it is outside. As mathematically discussed and experimentally evaluated in [143], $\mathcal{L}_{\text{out}}^{\text{sup}}$ presents the best results and $\mathcal{L}_{\text{out}}^{\text{sup}} >= \mathcal{L}_{\text{in}}^{\text{sup}}$ by the Jensen's Inequality [129].

10.2 Proposed Method

The proposed contextual loss is based on the supervised contrastive loss [143], more specifically the one defined by Equation 10.2. Among the various factors that significantly impact the performance of a loss function, the similarity measurement is a crucial one. Accurately measuring the similarity between elements helps to quantify the difference between the predicted values and the actual values. This measurement guides the learning process, enabling the model to make more accurate predictions.

10.2.1 Pairwise Similarity and Contextual Information

Originally, the similarity between two elements of indexes i and p is computed considering the dot product of their embeddings [143] denoted by $\boldsymbol{z}_i$ and $\boldsymbol{z}_p$. This operation is equivalent to cosine similarity if both embeddings are normalized. Normalizing a vector means dividing each component of the vector by the magnitude of the vector, resulting in a vector of length one. Therefore, we can define a function that computes the similarity between embeddings as done in the original loss: $\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_p) = \boldsymbol{z}_i \cdot \boldsymbol{z}_p$.

Pairwise measures have been widely employed in various cases. However, they are limited in multiple scenarios since they often ignore contextual similarity information [246]. The concept of *"contextual information"* is overly used in the literature with different meanings. In this work, the term *"contextual information"* is used to describe the process of exploiting the information given by the closest neighboring elements of a given item to calculate a more accurate similarity measure. The following subsection provides a definition and discussion of the neighborhood set.

10.2.2 Neighborhood Definition

Let $\mathcal{C} = \{obj_1, obj_2, \ldots, obj_n\}$ be an image collection. Let $\boldsymbol{z}_i$ denote an embedding for the image obj_i in a metric space $\mathbb{R}^d$, where d is the size of the embedding (number of dimensions). Based on the comparison between embeddings, an ordered list of nearest neighbors can be computed. Let sim: $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a function that computes the similarity between two images according to their corresponding embeddings (i.e., cosine distance). Formally, the cosine similarity between two images obj_i, obj_j is defined by $\text{sim}(\boldsymbol{z}_i,\ \boldsymbol{z}_j)$.

For a given anchor $obj_i \in \mathcal{C}$, the set of the k nearest neighbors (kNN) of obj_i, denoted by $NN_k(obj_i)$, contains the k most similar images to obj_i in the collection $\mathcal{C}$. Let $|NN_k(x_i)| = k$, where $|\cdot|$ denotes the cardinality of the set. For every $x_j \in NN_k(x_i)$ and every $x_l \notin NN_k(x_i)$, it holds that $d(x_i, x_j) \leq d(x_i, x_l)$. Additionally, we define $NN_k^{\mathcal{Y}}(obj_i)$ as the subset of $NN_k(obj_i)$ where each image belongs to the same class $\mathcal{Y}$ as obj_i. This subset can be expressed as: $NN_k^{\mathcal{Y}}(obj_i) = \{x \in NN_k(obj_i) \mid \text{class}(x) = \mathcal{Y}\}$. This definition ensures that $NN_k^{\mathcal{Y}}(obj_i)$ exclusively contains images from class $\mathcal{Y}$.

10.2.3 Contextual Similarity and Symmetry Discussion

Based on the idea of using contextual information to improve the similarity between elements and the neighborhood definition, we propose a novel contextual similarity measure:

$$\text{sim}_{\text{ctx}}\left(\boldsymbol{z}_p, \boldsymbol{z}_i, k\right) = \frac{1}{|NN_k^{\mathcal{Y}}(i)|} \times \sum_{j \in NN_k(i)} \text{sim}\left(\boldsymbol{z}_p, \boldsymbol{z}_j\right), \tag{10.4}$$

where $\boldsymbol{z}_p$ and $\boldsymbol{z}_i$ are the embeddings being compared and $k \in \mathbb{R}^+$ is a scalar value that defines the neighborhood size. The function *sim* is the dot product operation between the two embeddings, defined by $\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_p) = \boldsymbol{z}_i \cdot \boldsymbol{z}_p$. However, the result of sim_{ctx} for the pairs $(\boldsymbol{z}_p, \boldsymbol{z}_i)$ and $(\boldsymbol{z}_i, \boldsymbol{z}_p)$ is not symmetric, which is an important aspect in this scenario. Therefore, to ensure symmetry, we propose to sum the symmetric pairs, each raised to the power of 2:

$$\text{sim}_{\text{ctx}}^{\text{sym}}\left(\boldsymbol{z}_p, \boldsymbol{z}_i, k\right) = \text{sim}_{\text{ctx}}\left(\boldsymbol{z}_p, \boldsymbol{z}_i, k\right)^2 + \text{sim}_{\text{ctx}}\left(\boldsymbol{z}_i, \boldsymbol{z}_p, k\right)^2. \tag{10.5}$$

The importance of using the squared is further discussed in the next subsections.

10.2.4 Neighborhood Size and Logarithmic Decay

The neighborhood size, which is defined by the scalar $k \in \mathbb{Z}^+$ is of fundamental importance in the approach, since it defines the number of elements to be considered by the contextual similarity in Equations 10.4 and 10.5. However, the optimal value of k tends to vary throughout the training process. In the beginning, larger adjustments are necessary for the network weights, while towards the end, smaller adjustments are required. This can be explained by the inherent convergence of the learning process and also by the decay of the learning rate, which follows a cosine function in this work.

Let k_{start} be the initial value of k for the first epoch, $\xi \in \mathbb{Z}^+$ be the current epoch, and ξ_{total} the total number of epochs to run. The value of k is computed according to a logarithmic decay across epochs, defined as follows: $k = \max\Big(1,\ \text{round}((1 - \log_{\xi_{total}}(\xi)) \cdot k_{\text{start}})\Big)$, where round is a function that returns the nearest integer to a given real number.

10.2.5 Proposed Contextual Contrastive Loss (CCL)

The distance of a point $P(a, b, c)$ in a 3D space to the origin $O(0, 0, 0)$ is given by the square root of the sum of three terms squared (i.e, $\sqrt{a^2 + b^2 + c^2}$). In the context of our proposal, we can use this equation to define the contextual contrastive similarity between i and p: $\text{sim}_{\text{ccl}}\left(\boldsymbol{z}_i, \boldsymbol{z}_p, k\right) = \sqrt{\text{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_p\right)^2 + \text{sim}_{\text{ctx}}\left(\boldsymbol{z}_i, \boldsymbol{z}_p, k\right)^2 + \text{sim}_{\text{ctx}}\left(\boldsymbol{z}_p, \boldsymbol{z}_i, k\right)^2}$. Using all the previous definitions, this can be simplified as:

$$\mathrm{sim}_{\mathrm{ccl}}\left(\boldsymbol{z}_i, \boldsymbol{z}_p, k\right) = \sqrt{\mathrm{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_p\right)^2 + \mathrm{sim}_{\mathrm{ctx}}^{\mathrm{sym}}\left(\boldsymbol{z}_p, \boldsymbol{z}_i, k\right)}, \tag{10.6}$$

where the result of $\mathrm{sim}_{\mathrm{ccl}}$ is the same for symmetric pairs.

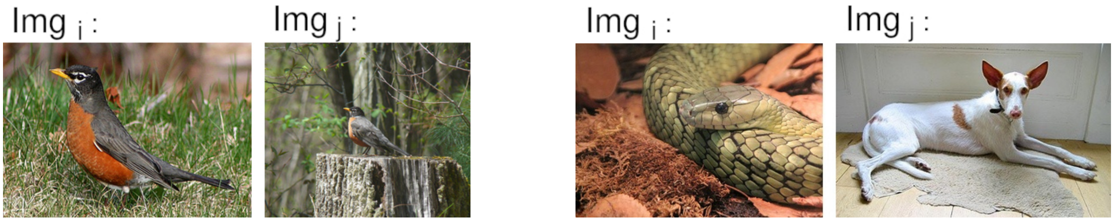
With $\mathrm{sim}_{\mathrm{ccl}}$, the complete equation of our proposed contextual contrastive loss (CCL) is:

$$\mathcal{L}^{\mathrm{ccl}} = \sum_{i \in I} \mathcal{L}_i^{\mathrm{ccl}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\mathrm{sim}_{\mathrm{ccl}}\left(\boldsymbol{z}_i, \boldsymbol{z}_p\right)/\psi\right)}{\sum_{a \in A(i)} \exp\left(\mathrm{sim}_{\mathrm{ccl}}\left(\boldsymbol{z}_i, \boldsymbol{z}_a\right)/\psi\right)}, \tag{10.7}$$

where the variable k is omitted for readability proposes.

Aiming at illustrating the proposed loss, a visualization was created to show the distribution of elements by their distances at the beginning and end of the training process when considering our proposed approach that uses contextual similarity information. Figure 10.1 presents the pairs of images considered as references. Figure 10.2 shows the plots for the same (blue) and different (red) classes. Each plot contains 1000 dots, which correspond to the top-1000 nearest neighbors of obj_i. Each dot represents a distinct image, and its position is determined based on the distance from the reference images.

Initially, the distributions are completely chaotic as shown in both *(a)* and *(c)*. Notice that as training enhances the separability between classes, in *(b)*, the dots tend to align in a line from bottom to top, left to right. Conversely, in *(d)*, they tend to form a line from bottom to top, right to left.



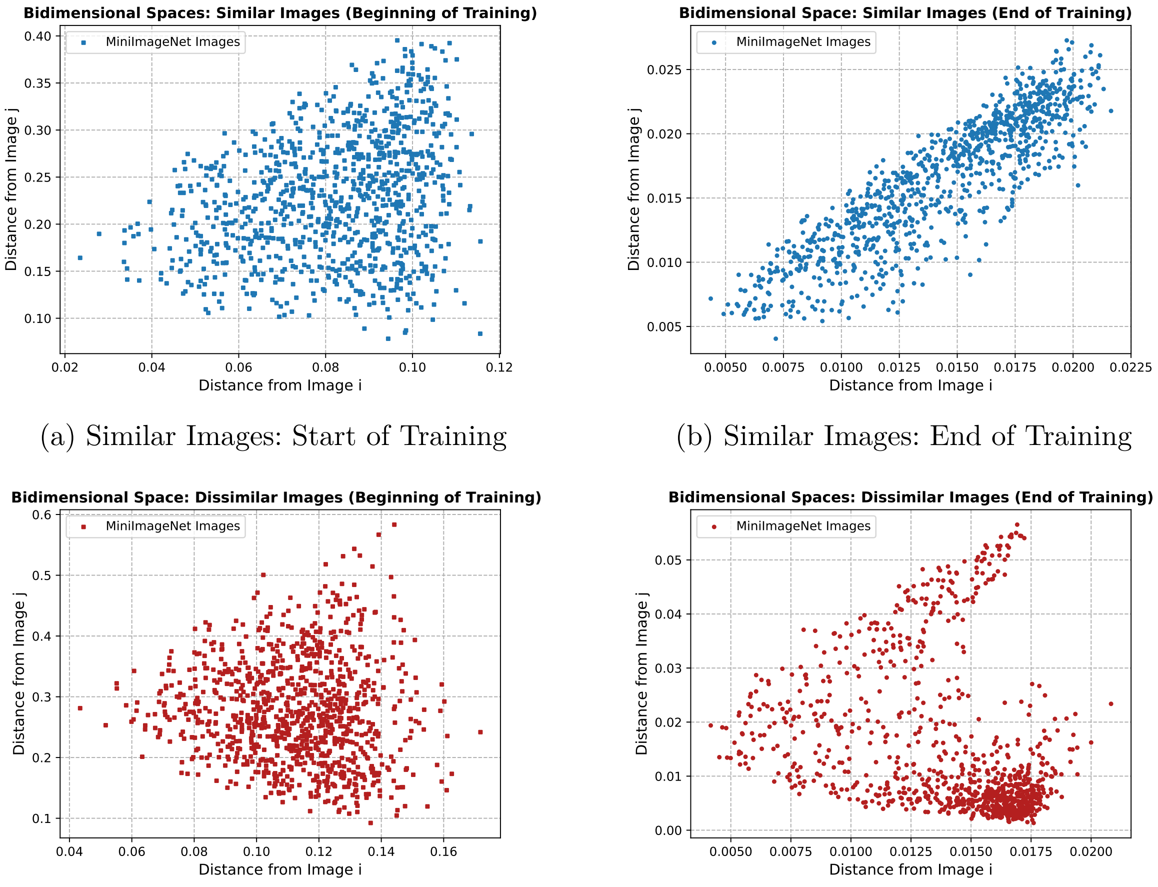
(a) Similar reference images

(b) Dissimilar reference images

Figure 10.1 – MiniImageNet images used as references for the bidimensional space plots.

10.2.6 Proposed Training Workflow

This section explains the workflow of the proposed approach and all its steps from training to testing, including how the proposed CCL is used by the metric learning model. Figure 10.3 presents an overview of the four steps that compose our framework, which is divided into two main categories: *(i)* metric learning: given image data, it learns new embedding representations based on the contrastive loss; and *(ii)* classification: where a linear model is trained using the binary cross-entropy loss to classify the embeddings according to their classes.



(a) Similar Images: Start of Training

(b) Similar Images: End of Training

(c) Dissimilar Images: Beginning of Training

(d) Dissimilar Images: End of Training

Figure 10.2 – Bidimensional space for similar and dissimilar images on the MiniImageNet dataset at the start (10 epochs) and end (300 epochs) of training using the proposed CCL.

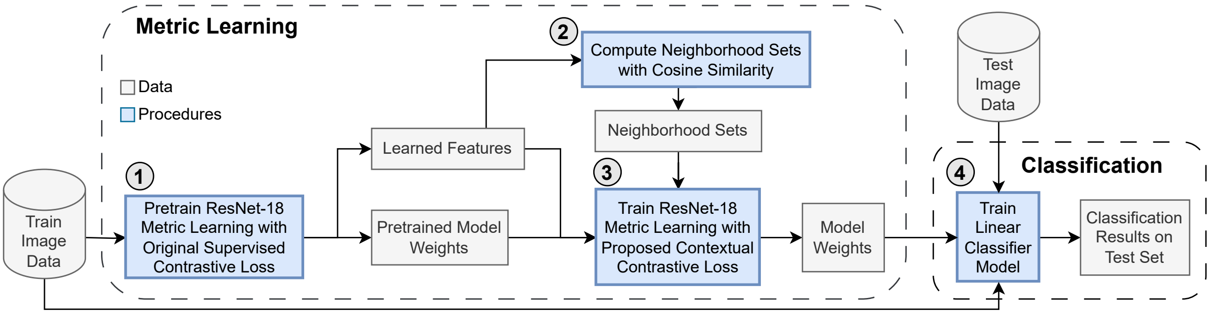

Metric Learning
Data
Procedures
Train Image Data
1
Pretrain ResNet-18 Metric Learning with Original Supervised Contrastive Loss
Learned Features
Pretrained Model Weights
2
Compute Neighborhood Sets with Cosine Similarity
Neighborhood Sets
3
Train ResNet-18 Metric Learning with Proposed Contextual Contrastive Loss
Model Weights
Test Image Data
Classification
4
Train Linear Classifier Model
Classification Results on Test Set

Figure 10.3 – Workflow of the steps of the proposed approach.

The procedures are marked in blue color and the data, that flows (input/output) between procedures are marked in gray. The steps of the workflow, marked in blue, are the following:

1. **Metric Learning Pretraining:** A pretraining is conducted using the metric learning model and the original supervised contrastive loss. The weights of this training are later used to generate the neighborhood set and for training the metric learning

model in step (3). For a fair comparison, this step is included for both the baseline and ours.

2. **Compute Neighborhood Sets:** The neighborhood sets are computed based on the features (i.e., embeddings) extracted by the pretrained model. The neighborhood sets are computed according to the formulation in Section 10.2.2. Our approach is efficient since the neighborhood sets are computed only once and do not need to be recomputed.
3. **Train Metric Learning with CCL:** The metric learning receives RGB images as input and learns embeddings (features) to represent them in a space of d dimensions. In this work, all embeddings are generated with 128 positions. The metric learning step uses the proposed CCL for learning more accurate representations. To calculate the similarity with the nearest neighbors, a set of features is considered. This feature set is updated each epoch with the features generated for the batches in every iteration within that epoch. If an image appears more than once, only the most recent feature from it is used to update the feature set.
4. **Classification:** A linear classification model is trained using the embeddings learned by the metric learning model. This model is used to predict the labels for the test set. The accuracy is computed and reported on the test set.

10.3 Experimental Evaluation

In this section, we describe the protocol and present both the quantitative and qualitative results obtained. Our proposed CCL loss is frequently compared with SupCon [143] because it is based on it. Additionally, we include comparisons with SimCLR [47], which, although unsupervised, was also compared to SupCon [143] in its original publication [143]. Three datasets were considered: Food101 [33], MiniImageNet [345], and CIFAR-100 [150]. These datasets were selected because a higher volume of images and larger classes are typically used to evaluate contrastive learning approaches. More detailed information about the datasets and effectiveness measures is presented in Chapter 4.

Table 10.1 presents the default hyperparameters used for the metric learning model and linear classifier model. Most of the parameters were adopted according to the Supervised Contrastive Loss (SupCon) implementation [15], which CCL is based on. We adopted the same parameters for CCL and SupCon loss to make a fair comparison and to ensure consistency. The parameters that are specific to our approach are marked with a star symbol*.

[15] github.com/HobbitLong/SupContrast

Table 10.1 – Neural network architecture and default hyperparameters utilized in the evaluation.

Parameter	**Metric Learning**	**Downstream Classifier**
Architecture	ResNet-18	Linear Classifier
Loss Function	Contrastive	Cross-entropy
Batch Size	128	128
Epochs (ξ)	100	20
Pre-Training Epochs*	10	—
Neighborhood Size (k)*	50	—
Temperature (ψ)	0.1	—
Image Resolution	Augmented 32×32 Crop	Resized to 64×64
Output Feature Size (d)	128	—
Learning Rate	0.5	5
Cosine Learning Rate Decay	True	True
Learning Rate Warmup	True	True
Weight Decay	10^{-4}	0
Momentum	0.9	0.9
Optimizer	Stochastic gradient descent (SGD)	Stochastic gradient descent (SGD)

Among the parameters, an experiment was conducted to evaluate two crucial ones: the batch size and the neighborhood size (k). These experiments were conducted on the Food101 dataset, which is the largest one, with a random split of 20% of images for training. Batch size plays a crucial role in contrastive learning, which hinges on comparing different data samples to learn distinctive features. A larger batch size provides more diverse sample pairs, enhancing the model's ability to generalize and distinguish between features. However, it must be carefully chosen to balance the quality of the learned representations. Table 10.2 presents the accuracy for different batch sizes for both SupCon [143] and CCL. Notably, there is a significant increase in accuracy when the batch size changes from 64 to 128; beyond this point, the accuracy begins to stabilize. Also, our CCL presented gains in all cases. These results are plotted in Figure 10.4, where the dashed line indicates the default batch chosen.

Table 10.2 – Impact of batch size on accuracy (%) on Food101 dataset, considering a split of 20% for training.

Batch Analysis: Acc. (%) on Food101			
Batch Size	**SupCon [143]**	**CCL (ours)**	**Relative Gain**
64	42.07	44.02	**+4.635%**
128	49.05	53.34	**+8.746%**
192	51.33	54.66	**+6.487%**
256	52.41	52.87	**+0.878%**
Avg.	48.71	51.22	**+5.190%**

Table 10.3 presents the analysis of the parameter k. It is observed that $k = 70$ is the best setting in most cases. However, the variation in results across different k values is small, suggesting that CCL is robust to different choices of k. Also, for 300 epochs, an even smaller k can be considered. Therefore, we adopted $k = 70$ for all cases and $k = 30$ for 300 epochs in the remaining experiments.

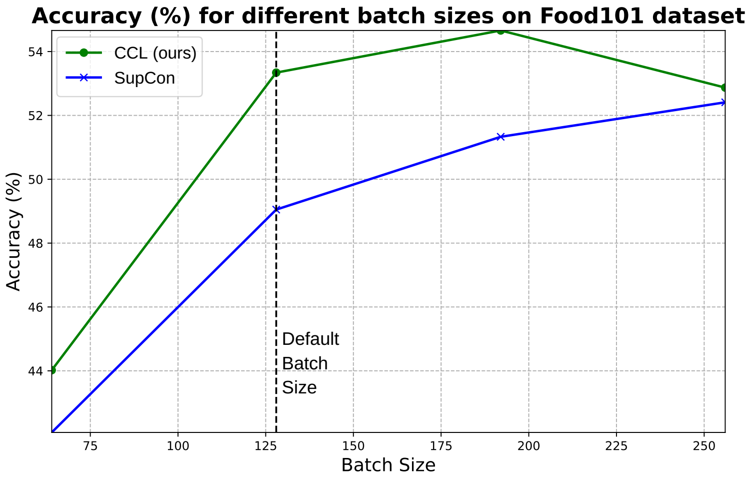


Figure 10.4 – Accuracy (%) on the test set for different batch sizes.

Table 10.3 – Impact of parameter k (neighborhood size) on accuracy (%). Results highlighted in gray deviate less than 0.20 from the best value in bold.

k Analysis: Accuracies (%) on Food101 dataset						
Train	**Epochs**	**SupCon [143]**	**k=30**	**k=50**	**k=70**	**k=90**
20%	**100**	48.32	51.19	53.14	**54.10**	53.78
	200	56.50	58.59	**58.96**	58.80	58.69
	300	58.11	**59.86**	59.40	58.87	58.44
40%	**100**	62.47	64.68	65.65	65.86	**65.95**
	200	67.30	68.27	68.66	**68.72**	68.30
	300	68.02	68.95	**68.97**	68.80	68.59
Average		60.12	61.92	62.46	**62.53**	62.29

With all the parameters and protocol set, an evaluation was conducted considering various training splits (20%, 40%, 60%, and 80%) to assess the robustness of CCL for 100 training epochs when compared to SimCLR [47] and SupCon [143]. For each training percentage, three different splits were randomly generated. These same splits were used when comparing our loss function to others. The results are presented as the mean accuracy and a 95% confidence interval across the three splits. Table 10.4 presents the accuracy results for the three evaluated datasets: Food101, MiniImageNet, and CIFAR-100. The results reveal gains in all cases, especially with fewer training data which is a more challenging scenario.

For the Food101 dataset, the most extensive dataset included in our evaluation, we conducted experiments for 100, 200, and 300 epochs. Table 10.5 shows improvements across all scenarios. These results reveal a significant benefit of our method: it achieves superior performance in situations with limited training data and fewer epochs, which reveals the potential of our method in resource-constrained scenarios. Additionally, CCL with 200 epochs achieves better results than SupCon with 300 epochs in all cases. To better illustrate the advantages of CCL compared to SupCon [143], Figure 10.5 displays the accuracies on the test set during training. For 185 epochs, CCL reaches the accuracy that SupCon achieves in 300.

Table 10.4 – Accuracies (%) achieved for 100 epochs of training, comparing the proposed CCL with other contrastive losses (i.e., SimCLR [47] and SupCon [143]), across four training set sizes on three datasets. The relative gains compare CCL with SupCon [143].

Dataset	**Loss**	**Dataset percentages used for (training,testing)**				**Average**
		(20%, 80%)	**(40%, 60%)**	**(60%, 40%)**	**(80%, 20%)**	**Values**
Food101	SimCLR [47]	31.889 ± 1.974	39.920 ± 0.149	44.246 ± 0.724	47.108 ± 0.937	40.791
	SupCon [143]	48.369 ± 0.515	62.346 ± 0.504	68.649 ± 0.300	71.998 ± 0.459	62.841
	CCL (ours)	53.573 ± 0.347	65.672 ± 0.368	71.074 ± 1.010	73.920 ± 0.935	66.060
	R. Gain	**+10.759%**	**+5.335%**	**+3.532%**	**+2.670%**	**+5.574%**
MiniImageNet	SimCLR [47]	37.909 ± 0.393	48.197 ± 0.244	54.148 ± 1.735	58.427 ± 1.121	49.670
	SupCon [143]	53.466 ± 1.133	67.269 ± 0.537	73.429 ± 0.949	77.454 ± 0.793	67.905
	CCL (ours)	57.231 ± 1.194	69.263 ± 0.104	74.787 ± 0.645	78.217 ± 0.982	69.875
	R. Gain	**+7.042%**	**+2.964%**	**+1.849%**	**+0.985%**	**+3.210%**
CIFAR-100	SimCLR [47]	36.595 ± 2.503	46.018 ± 0.324	51.427 ± 0.426	54.740 ± 1.502	47.195
	SupCon [143]	56.133 ± 1.614	68.089 ± 0.758	73.347 ± 0.545	76.383 ± 0.562	68.488
	CCL (ours)	58.813 ± 0.116	69.748 ± 0.124	74.753 ± 0.496	77.613 ± 1.283	70.232
	R. Gain	**+4.774%**	**+2.437%**	**+1.917%**	**+1.610%**	**+2.685%**
Average Gain		**+7.525%**	**+3.579%**	**+2.433%**	**+1.755%**	**+3.823%**

Table 10.5 – Accuracies (%) achieved on the Food101 dataset when comparing the proposed CCL against SupCon [143], for different training epochs.

Analysis of the number of epochs on the Food101 dataset						
Epochs	**Loss**	**Dataset percentages used for (training, testing)**				**Average**
		(20%, 80%)	**(40%, 60%)**	**(60%, 40%)**	**(80%, 20%)**	**Values**
100	SupCon [143]	48.369 ± 0.515	62.346 ± 0.504	68.649 ± 0.300	71.998 ± 0.459	62.841
	CCL (ours)	53.573 ± 0.347	65.672 ± 0.368	71.074 ± 1.010	73.920 ± 0.935	66.060
	R. Gain	**+10.759%**	**+5.335%**	**+3.532%**	**+2.670%**	**+5.574%**
200	SupCon [143]	56.116 ± 0.836	67.164 ± 0.656	72.102 ± 1.082	74.787 ± 1.418	67.542
	CCL (ours)	58.392 ± 0.636	68.657 ± 0.324	73.113 ± 0.709	75.748 ± 0.707	68.978
	R. Gain	**+4.056%**	**+2.223%**	**+1.402%**	**+1.285%**	**+2.242%**
300	SupCon [143]	57.981 ± 0.285	68.093 ± 0.345	72.738 ± 1.023	75.498 ± 0.200	68.578
	CCL (ours)	59.589 ± 0.626	69.094 ± 0.228	73.253 ± 1.023	75.691 ± 0.806	69.406
	R. Gain	**+2.773%**	**+1.470%**	**+0.708%**	**+0.256%**	**+1.302%**
Average Gain		**+5.863%**	**+3.009%**	**+1.881%**	**+1.404%**	**+3.039%**

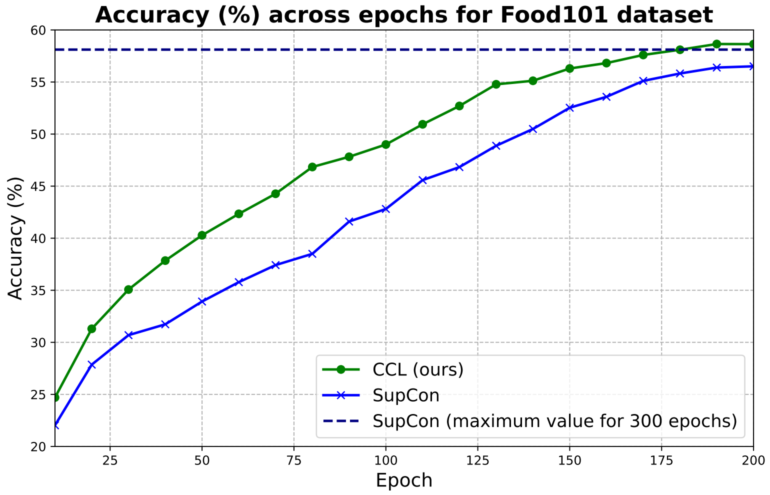
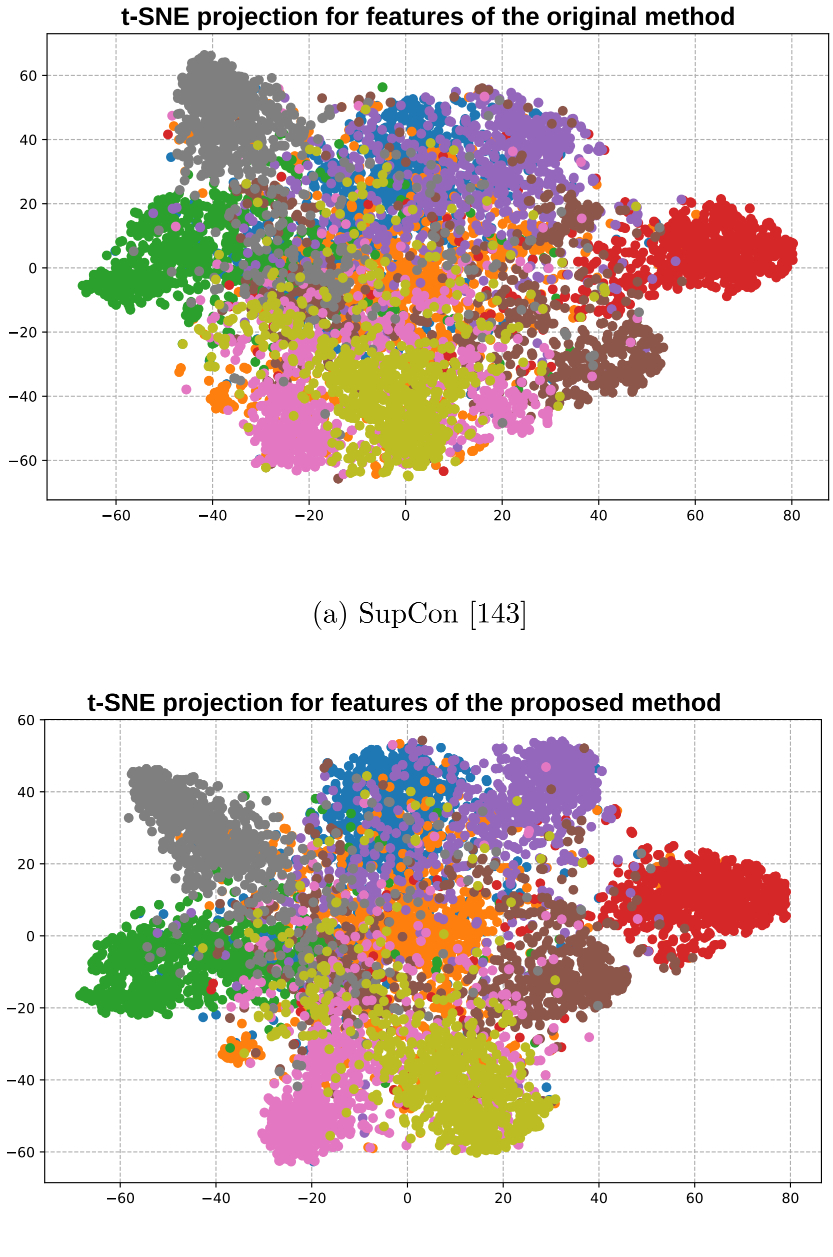


Figure 10.5 – Accuracy (%) on the test set across epochs comparing SupCon to CCL.

Finally, we present a qualitative result. A t-SNE projection was generated on the Food101 dataset considering 9 random classes. Figure 10.6 presents the results considering the features from the SupCon loss and our proposed (CCL) loss, which were extracted from the linear classification model on the test set. Each color represents a different class. Notice that our approach, shown in plot *(b)*, presents better separability of classes. In plot *(a)*, for example, the orange group is barely visible, groups yellow and pink overlap significantly. Additionally, other groups, such as the red and gray, are situated closer to the others. All these cases were improved in plot *(b)*.



(a) SupCon [143]

(b) CCL (ours)

Figure 10.6 – t-SNE visualization for 9 classes comparing the features of the original method to CCL on the Food101 dataset with 20% of training data.

11 Conclusions

This chapter concludes this dissertation by discussing the contributions and other relevant aspects. Section 11.1 reviews the main results obtained for each task: query performance prediction, image retrieval, and image classification. Additionally, it provides a comparative analysis of the proposed methods alongside other approaches from the literature. Section 11.2 discusses how the contributions address the research questions of this study. Section 11.3 lists the publications and submissions obtained, along with the international Fulbright fellowship. Section 11.4 mentions the available codes for the proposed approaches. Finally, Section 11.5 presents potential extensions and future work, describing their connections to the contributions achieved in this research.

11.1 Discussion of Results

Given the notable outcomes achieved by contextual similarity learning across all scenarios considered, this section discusses the results for each task. For query performance prediction, the results of RQPPF and DRNE are jointly compared and discussed in Section 11.1.1. Section 11.1.2 compares the approaches evaluated for image retrieval in person Re-ID and general-purpose datasets, including comparisons with the state-of-the-art. Section 11.1.3 overviews Manifold-GCN and RFE semi-supervised classification results obtained and a comparison with the state-of-the-art. Moreover, the gains achieved by CCL are briefly discussed.

11.1.1 Query Performance Prediction

A great variety of experiments was conducted to evaluate DRNE and RQPPF, showing their capacity to effectively perform QPP in various datasets. Table 11.1 presents a summary of the relative gains for both approaches in comparison to Authority [243] and Reciprocal Density [248], which are used as baselines. Notice that RQPPF provided gains in all the evaluated scenarios, while DRNE showed inferior performance in some cases, especially when compared to Reciprocal in the MPEG-7 dataset. However, for the most part, DRNE provided higher and more consistent gains when compared to Reciprocal. Specifically for the AIR descriptor, DRNE revealed superior results in all cases.

In general, the results showed that the proposed methods are better than the baselines in most cases. Additionally, the choice of the best method depends not only on the dataset but also on the descriptor. RQPPF is more flexible and also uses Authority and Reciprocal as part of its formulation, while DRNE does not. However, DRNE seems more robust to outlier descriptors, while RQPPF does not. Among potential extensions, combining DRNE and RQPPF is one of the possibilities for future work.

Table 11.1 – Relative gains of DRNE and RQPPF when compared to Authority (Auth.) and Reciprocal Density (Rec.). Average gains are reported for each dataset.

Descriptor	**Original MAP**	**Compared to Auth. [243]**		**Compared to Rec. [248]**	
		DRNE	**RQPPF**	**DRNE**	**RQPPF**
MPEG-7					
AIR [103]	89.39%	+14.81%	+3.50%	+16.84%	+12.99%
ASC [191]	85.28%	-2.50%	+3.76%	-8.29%	+1.84%
IDSC [190]	81.70%	-3.93%	+3.69%	-7.58%	+1.88%
CFD [244]	80.71%	+3.47%	+3.99%	-1.24%	+2.71%
BAS [13]	71.52%	+0.85%	+3.69%	-5.13%	+1.09%
SS [317]	37.67%	+7.08%	+6.01%	+3.32%	+3.52%
Average Gain		+3.30%	+4.11%	-0.35%	+4.01%
Brodatz					
LAS [308]	75.15%	+7.50%	+9.01%	+9.59%	+11.15%
CCOM [148]	57.57%	+3.30%	+7.33%	+8.60%	+11.18%
LBP [231]	48.40%	+0.75%	+5.10%	+18.42%	+15.36%
Average Gain		+3.85%	+7.15%	+12.20%	+12.56%
Market					
OSNET [436]	43.30%	-2.63%	+1.19%	+5.40%	+5.45%
ResNet [110]	22.82%	-0.89%	+0.13%	+7.95%	+5.29%
Average Gain		-1.76%	+0.66%	+6.68%	+5.37%
Duke					
OSNET [436]	52.69%	+0.71%	+2.35%	+1.00%	+3.37%
ResNet [110]	32.00%	-2.14%	+0.52%	-0.12%	+2.46%
Average Gain		-0.72%	+1.44%	+0.44%	+2.92%

11.1.2 Image Retrieval

Four of the seven proposed methods were evaluated in image retrieval: HRSF, JaccardMax, RFE, and Manifold-GCN. The results obtained are reviewed for both person Re-ID and general-purpose datasets, including comparisons against each other and with the state-of-the-art. A brief discussion about the gains is also presented.

- **Person Re-ID**

Considering the wide variety of descriptors employed and to provide a fair comparison, Table 11.2 presents the best results obtained for each method using only the OSNET model and its variants (i.e., OSNET, OSNET-IBN, and OSNET-AIN) on Market, DukeMTMC, and CUHK03 datasets.

For the Market and CUHK03 datasets, HRSF leads with the best results for both R1 and MAP. HRSF is the only method that performs selection, which is an advantage over the others since it can select the best combination of descriptors among the OSNET variants. For the DukeMTMC dataset, RFE and JaccardMax compete for the best results. The worst results in this table are the ones obtained by the Manifold-GCN. Besides Manifold-GCN being semi-supervised, while all the other approaches are unsupervised, this result highlights the importance of future research for this method. Since it was mainly proposed for classification, the results for retrieval are significantly behind others, probably due to the features not being properly distributed in the latent space, which requires further investigation in future work.

Table 11.3 presents the methods ranked according to their results for each measure and dataset. The average rank reveals that, while HRSF shows the best results in most cases, JaccardMax and RFE follow closely, with average ranks of 2.0 and 2.2, respectively. As previously discussed, Manifold-GCN is behind with an average rank of 3.7.

Table 11.2 – Comparison between the proposed approaches on person Re-ID considering MAP (%) and R-01 (%). The best results obtained with the OSNET descriptor and its variants are reported.

Method	**Year**	**Datasets**					
		Market1501		**DukeMTMC**		**CUHK03**	
		R1	**MAP**	**R1**	**MAP**	**R1**	**MAP**
HRSF ($\mathfrak{X}^*$, best result) [331]	2022	**75.71**	**62.94**	77.24	68.88	**39.04**	**39.69**
Correlation Graph + Jaccard Max [324]	2022	73.25	59.84	76.21	**69.27**	—	—
RFE [334]	2023	72.42	59.51	**77.69**	69.21	36.89	39.24
Manifold-GCN [333]	2023	70.30	57.48	74.22	65.83	35.19	35.99

Table 11.3 – Proposed approaches on person Re-ID ranked according to their effectiveness (R1 and MAP). The best results obtained with the OSNET descriptor and its variants were considered.

Method	**Year**	**Datasets**						**Average Rank**
		Market1501		**DukeMTMC**		**CUHK03**		
		R1	**MAP**	**R1**	**MAP**	**R1**	**MAP**	
HRSF ($\mathfrak{X}^*$, best result) [331]	2022	**1**	**1**	2	3	**1**	**1**	1.5
Correlation Graph + Jaccard Max [324]	2022	2	2	3	**1**	—	—	2.0
RFE [334]	2023	3	3	**1**	2	2	2	2.2
Manifold-GCN [333]	2023	4	4	4	4	3	3	3.7

To compare the proposed methods with the state-of-the-art in person Re-ID, which is presented in Table 11.4, the best results for each approach were considered. For HRSF, JaccardMax, and RFE the best results used OSNET and its variants. Unlike the others, the JaccardMax evaluation employed the TransReID descriptor, which provided better results for this method. This table highlights in bold the highest value for each column. The best among our methods is also highlighted. All the baseline results are the ones reported in the literature, following the same protocol as ours.

In general, it can be observed that the proposed approaches provide better results for MAP than R1 when compared to other methods. This evinces that they can significantly improve the top positions of ranked lists, but not necessarily achieve the best results when considering only the first position. In this case, Market1501 was revealed as the most challenging dataset, where the proposed methods are better or comparable to the ones up to 2020. After that, the baselines show a considerable improvement. In contrast, for the DukeMTMC dataset, the MAP of 73.96% obtained by the proposed JaccardMax (2022) is the best result achieved, surpassed only by VAL-PAT, which is a very recent approach from 2023. For the CUHK03 dataset, many of the methods have no results reported in

Table 11.4 – Proposed approaches compared to state-of-the-art on person Re-ID considering MAP (%) and R-01 (%).

Method	**Year**	**Datasets**					
		Market1501		**DukeMTMC**		**CUHK03**	
		R1	**MAP**	**R1**	**MAP**	**R1**	**MAP**
Unsupervised Methods							
ARN [181]	2018	70.3	39.4	60.2	33.4	—	—
EANet [118]	2018	66.4	40.6	45.0	26.4	51.4	31.7
TAUDL [170]	2018	63.7	41.2	61.7	43.5	44.7	31.2
ECN [431]	2019	75.1	43.0	63.3	40.4	—	—
MAR [397]	2019	67.7	40.0	87.1	48.0	—	—
UTAL [171]	2019	69.2	46.2	62.3	44.6	56.3	42.3
SSL [189]	2020	71.7	37.8	52.5	28.6	—	—
HCT [402]	2020	80.0	56.4	69.6	50.7	—	—
CAP [353]	2021	91.4	79.2	81.1	67.3	—	—
IICS [376]	2021	89.5	72.9	80.0	64.4	—	—
RLCC [415]	2021	90.8	77.7	83.2	69.2	—	—
ICE [43]	2021	93.8	82.3	83.3	69.9	—	—
MGH [368]	2021	93.2	81.7	83.7	70.2	—	—
MGCE-HCL [297]	2022	92.1	79.6	82.5	67.5	—	—
MCRN [367]	2022	92.5	80.8	83.5	69.9	—	—
O2CAP [354]	2022	92.5	82.7	83.9	71.2	—	—
DIDAL [201]	2023	94.2	84.8	—	—	—	—
VAL-PAT [23]	2023	—	—	86.1	74.9	—	—
Domain Adaptative Methods							
HHL (D,M) [430]	2018	62.2	31.4	46.9	27.2	—	—
HHL (C03) [430]	2018	56.8	29.8	42.7	23.4	—	—
ATNet (D,M) [197]	2019	55.7	25.6	45.1	24.9	—	—
CSGLP (D,M) [273]	2019	63.7	33.9	56.1	36.0	—	—
ISSDA (D,M) [306]	2019	81.3	63.1	72.8	54.1	—	—
ECN++ (D,M) [432]	2020	84.1	63.8	74.0	54.4	—	—
MMCL (D,M) [348]	2020	84.4	60.4	72.4	51.4	—	—
JVCT+ (D,M) [44]	2021	90.5	75.4	81.9	67.6	—	—
MCRN (D,M) [367]	2022	93.8	83.8	84.5	71.5	—	—
Cross-Domain Methods (single-source)							
EANet (C03) [118]	2018	59.4	33.3	39.3	22.0	—	—
EANet (D,M) [118]	2018	61.7	32.9	51.4	31.7	—	—
SPGAN (D,M) [71]	2018	43.1	17.0	33.1	16.7	—	—
DAAM (D,M) [121]	2019	42.3	17.5	29.3	14.5	—	—
AF3 (D,M) [195]	2019	67.2	36.3	56.8	37.4	—	—
AF3 (MT) [195]	2019	68.0	37.7	66.3	46.2	—	—
PAUL (MT) [380]	2019	68.5	40.1	72.0	53.2	—	—
Cross-Domain Methods (multi-source)							
CAMEL [396]	2017	54.5	26.3	—	—	31.9	—
EMTL [370]	2018	52.8	25.1	39.7	22.3	—	—
Baseline by [153]	2019	80.5	56.8	67.4	46.9	29.4	27.4
Proposed Methods (contributions)							
HRSF ($\mathcal{X}^*$, best result) [331]	2022	**75.71**	62.94	77.24	68.88	**39.04**	**39.69**
Correlation Graph + Jaccard Max [324]	2022	75.42	**63.53**	**78.59**	**73.96**	—	—
RFE [334]	2023	72.42	59.51	77.69	69.21	36.89	39.24
Manifold-GCN [333]	2023	70.30	57.48	74.22	65.83	35.19	35.99

the literature, since this dataset is not as commonly evaluated as the others. However, all methods provided a better MAP than the baselines, being only behind UTAL.

The presented comparisons raise two topics for discussion: *(i)* Why the obtained results are significantly better in DukeMTMC? Why does Market1501 appear to be considerably more difficult? *(ii)* RFE and CG [249] + JaccardMax exhibit close results when using the same descriptors for Re-ID. Is this also true in other scenarios?

The first topic is challenging to answer, especially because Market1501 and DukeMTMC datasets have very similar characteristics (e.g., dataset size, number of individuals, images per person, size of the train and evaluation sets, and number of cameras). However, one particular difference might explain it. The Market1501 dataset was annotated using an automated detector, the Deformable Part Model (DPM), which is known to be prone to noise and potential misalignment. Conversely, the DukeMTMC was manually annotated by humans providing cleaner data with well-aligned bounding boxes. Further investigation to address this aspect can be conducted as future work.

Regarding the close results of RFE and CG [249] + JaccardMax for Re-ID, these methods are compared on general-purpose datasets to evaluate if they exhibit similar behavior.

- **General-Purpose Datasets**

Tables 11.5 and 11.6 compare RFE and CG [249] + JaccardMax with the state-of-the-art in image retrieval tasks for the datasets Holidays and UKBench, respectively. In both datasets, RFE outperformed all the baselines. For Holidays, CG [249] + JaccardMax is behind RFE with 91.12% but still surpasses most of the other methods. In contrast, for UKbench, both achieved the same result of 3.97, which is very close to the maximum score (i.e., 4).

Table 11.5 – State-of-the-art (SOTA) comparison on Holidays dataset (MAP).

MAP for state-of-the-art methods						
Jégou *et al.* [127]	Tolias *et al.* [315]	Paulin *et al.* [238]	Qin *et al.* [268]	Zheng *et al.* [425]	Sun *et al.* [299]	Zheng *et al.* [423]
75.07%	82.20%	82.90%	84.40%	85.20%	85.50%	85.80%

Pedronette *et al.* [241]	Arandjelovic *et al.* [12]	Li *et al.* [178]	Razavian *et al.* [271]	Pedronette *et al.* [253]	Gordo *et al.* [104]	Valem *et al.* [329]
86.16%	87.50%	89.20%	89.60%	90.02%	90.30%	90.51%

Valem *et al.* [328]	Liu *et al.* [203]	Pedronette *et al.* [251]	Pedronette *et al.* [252]	Yu *et al.* [398]	Berman *et al.* [26]
90.51%	90.89%	90.94%	91.25%	91.40%	91.80%

Proposed Approaches	
CG + JacMax	**RFE**
91.12%	**91.97%**

Table 11.6 – State-of-the-art (SOTA) comparison on UKBench dataset (NS-Score).

N-S-Scores for state-of-the-art methods							
Qin *et al.* [267]	Zhang *et al.* [413]	Zheng *et al.* [424]	Bai *et al.* [16]	Xie *et al.* [371]	Lv *et al.* [210]	Liu *et al.* [203]	Pedronette *et al.* [241]
3.67	3.83	3.84	3.86	3.89	3.91	3.92	3.93

Bai *et al.* [20]	Liu *et al.* [159]	Valem *et al.* [328]	Bai *et al.* [17]	Valem *et al.* [329]	Valem *et al.* [327]	Chen *et al.* [50]
3.93	3.93	3.93	3.94	3.94	3.95	3.96

Proposed Approaches	
CG + JacMax	**RFE**
3.97	**3.97**

- **Discussion about Gains**

From the observed results, we can notice that the proposed approaches are comparable or better than state-of-the-art approaches in most cases. The best method in each scenario varies since each dataset and descriptor presents different aspects. An important attribute of the proposed approaches is their capacity to improve the input data by employing contextual similarity learning. Figure 11.1 presents the relative gains of RFE and JaccardMax for different datasets and descriptors. This demonstrates the capacity of contextual similarity learning to improve the results across multiple scenarios. The Holidays and UKBench datasets exhibited smaller gains because their descriptors already achieved higher results, making further enhancements more challenging compared to other datasets. Despite this, it is impressive that, despite the advancements in feature extraction, for different deep learning models from CNNs to Vision Transformers, the potential to obtain improved results was achieved across all cases.

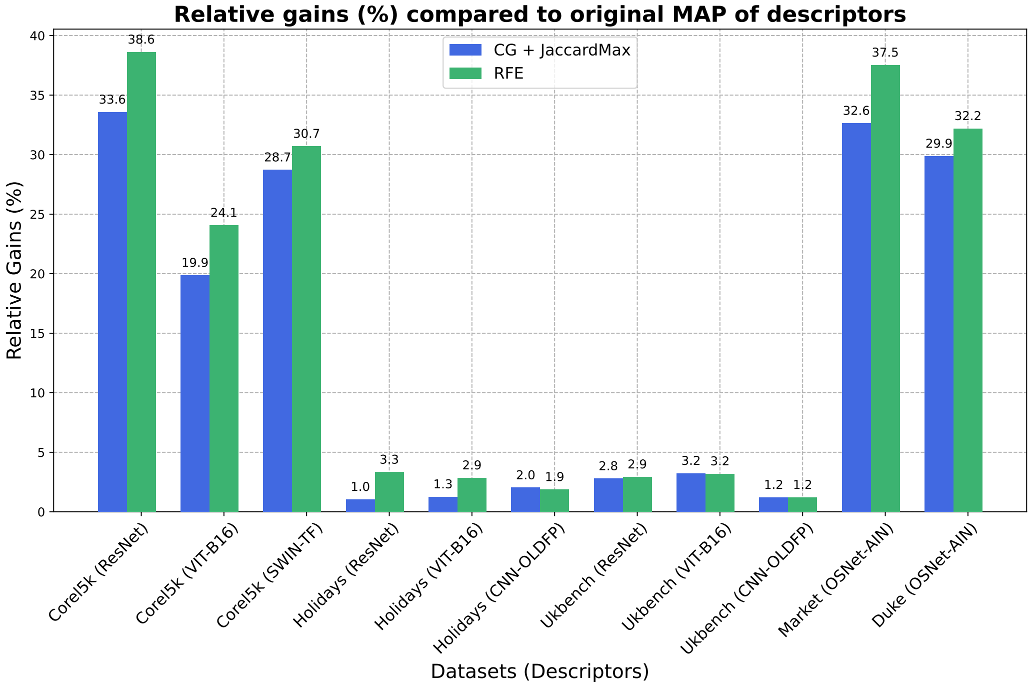


Figure 11.1 – RFE and JaccardMax relative gains (%) over MAP of descriptors.

11.1.3 Image Classification

Since both Manifold-GCN and RFE were employed for semi-supervised classification, Table 11.7 compares them to baselines, both traditional and recent, on Flowers and Corel5k datasets. The values achieved by Manifold-GCN are the highest in all the cases, and they are closely followed by RFE. These results reveal the high effectiveness of the proposed approaches that, besides the significant gains, are also comparable or superior to various methods in the literature.

Table 11.7 – Accuracy comparison (%) for baselines on Flowers and Corel5k datasets. We compared the proposed RFE and Manifold-GCN with semi-supervised classification baselines. The methods are compared with different input features. The results of our methods are highlighted with a gray background; the best results for each pair of features and dataset are marked in bold.

Method	**Input**	**Flowers**	**Corel5k**
CoMatch [169]	**Images**	82.55	*85.70*
kNN	**ResNet Features**	63.67	76.80
SVM [54]		80.54	88.73
OPF [8]		71.77	83.56
SL-Perceptron		75.44	83.56
ML-Perceptron		78.88	87.10
PseudoLabel+SGD [162]		82.69	89.76
LS+kNN [433]		73.49	83.98
LS+SVM [433, 54]		73.53	83.26
LS+OPF [433, 8]		72.66	82.32
LS+SL-Perceptron [433]		72.34	82.38
LS+ML-Perceptron [433]		73.03	82.53
GNN-LDS [90]		54.98	62.69
GNN-KNN-LDS [90]		79.32	88.94
WSEF [264]		85.12	91.68
RFE		84.95	91.54
Manifold-GCN		**85.88**	**93.08**
kNN	**SENet Features**	48.71	58.78
SVM [54]		73.30	85.89
OPF [8]		64.00	81.33
SL-Perceptron		71.84	82.28
ML-Perceptron		72.62	86.90
PseudoLabel+SGD [162]		76.87	89.85
LS+kNN [433]		58.05	72.16
LS+SVM [433, 54]		59.84	72.79
LS+OPF [433, 8]		59.25	72.20
LS+SL-Perceptron [433]		59.27	72.19
LS+ML-Perceptron [433]		59.39	72.24
GNN-LDS [90]		52.24	65.80
GNN-KNN-LDS [90]		73.69	89.95
WSEF [264]		76.16	89.74
RFE		77.56	92.20
Manifold-GCN		**78.82**	**92.79**

The CCL was also proposed and evaluated for classification but in supervised scenarios. Since it is the only method proposed in this category and it uses different datasets in the protocol, a direct comparison of it with other proposed methods is not feasible. Therefore, a discussion about its gains is presented. The experimental evaluation in Chapter 10 showed that the results are consistently better than those of SupCon, which CCL is based on, and SimCLR, another method commonly used as a baseline in this task. Figure 11.2 presents a plot that evinces the capacity of CCL to provide gains when compared to SupCon for three datasets and with higher values as the training set size decreases. The integration of contextual information within the contrastive loss significantly improved the results, as initially hypothesized, with gains up to 10.759%.

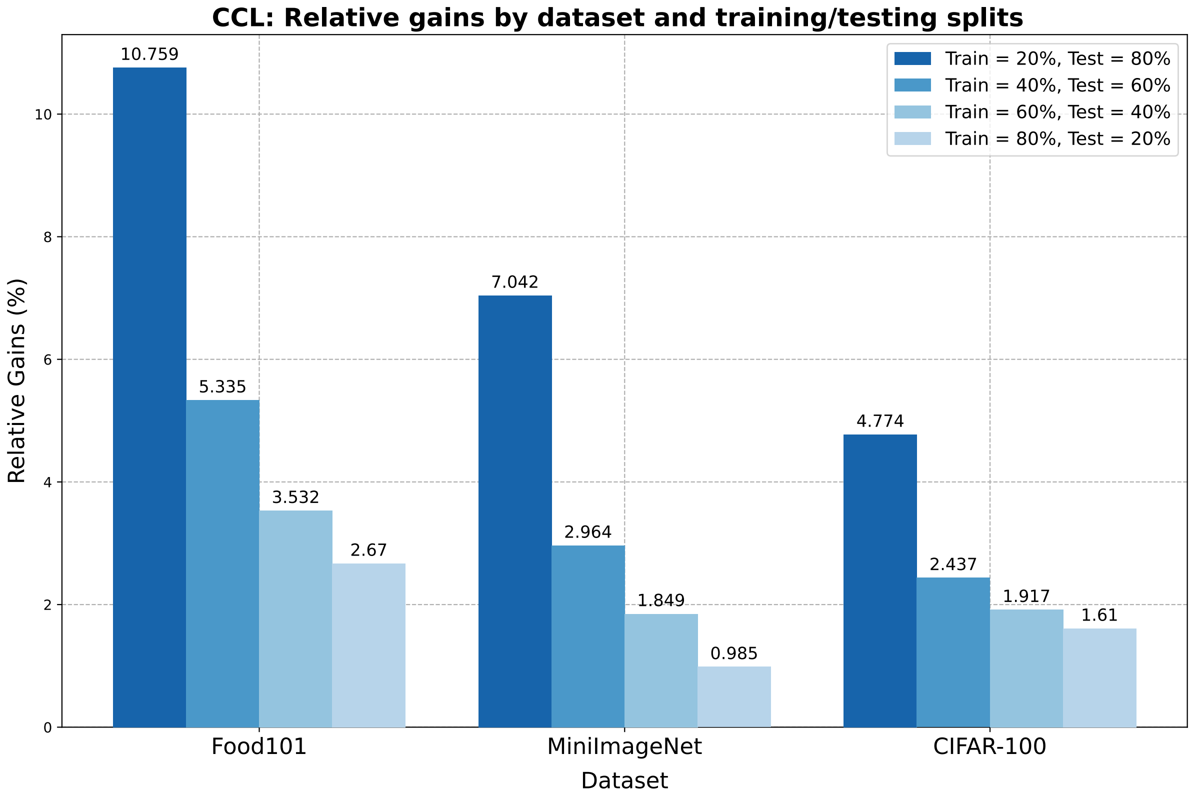


Figure 11.2 – Relative gains (%) obtained by CCL in comparison to SupCon for different train/test splits considering 100 training epochs.

11.2 Contributions and Research Questions

This section discusses the interconnections between research questions and proposed contributions. For the convenience of the reader, Table 11.8 presents a summary of all research questions, using bullet points to indicate the proposed methods associated with each.

1. *How can contextual similarity information be used for selection and fusion in unsupervised person Re-ID?*
   Discussion: HRSF performs selection and fusion of ranked lists from different descriptors in person Re-ID. It aims to select and combine data that is complementary and effective. The selection is based on measurements of correlation and query performance prediction. The experimental results revealed the effectiveness of the proposed approach.

2. *How can data be modeled using contextual similarity information for query performance prediction?*
   Discussion: Two approaches were proposed in this direction, DRNE and RQPPF. They model data using ranked lists, that encode similarity information among items. The DRNE creates *"contextual images"*, grayscale images where the intensity of pixels is defined based on the positions of items in the ranked lists. The RQPPF implements *"contextual rank-based features"*, computed considering reciprocal neighborhood, effectiveness estimation measures (i.e., Authority or Reciprocal Density), and positions in the ranked lists.

3. *How can contextual similarity information be used to generate synthetic data?*
   Discussion: This work exploited the idea of generating synthetic ranked lists. The elements are randomly generated according to the probabilities in the matrix. Various aspects of the neighborhood and how the elements are distributed across ranked lists can be encoded in this matrix. For example, the values in the diagonal can increase or decrease the effectiveness of the ranked lists being generated.

4. *How can contextual similarity learning be employed on synthetically generated data?*
   Discussion: Both DRNE and RQPPF address this question. The DRNE creates *"contextual images"* which are submitted to a denoising network. The proposed model is a variant of the Denoiser CNN, which returns a score for each contextual image. It interprets the incorrectness of a ranked list as noise, which is learned by the network during the training on synthetic data. In contrast, RQPPF models *"contextual rank-based features"*. This approach is flexible and enables the application of various regression models to estimate effectiveness measures using these meta-features.

5. *How can more complex structures, which encode contextual information more effectively, be applied to unsupervised similarity learning?*

   Discussion: One example of a structure that can be used is the hypergraph, which is well-suited for capturing more complex relationships in the data. Hypergraphs are utilized by both HRSF and RFE. Hypergraphs are useful because they allow edges to connect any number of vertices and can provide valuable insights. In these approaches, each image is represented by a node. For RFE, it considers the idea that similar objects present similar ranked lists and, therefore, similar hyperedges. Once the hyperedges are represented by an incidence matrix, the product of the hyperedges can be exploited to compute a more effective similarity measure between nodes. HRSF proposes the HQPP for selection, which is based on the conjecture that similar objects are expected to reference each other in the same hyperedge. Therefore, hyperedges that concentrate a high number of ranking references on a few nodes are expected to be more effective.

6. *How can contextual information from similarity learning approaches be encoded to generate embeddings that are useful for tasks beyond retrieval, such as classification?*

   Discussion: Generating embeddings is one of the key innovations of RFE. Connected components (CCs) with high confidence are defined based on hypergraph structures. These CCs are computed based on the most reliable edges identified through the hyperedge weights. The CCs encode class information and result in objects in the same CC to have their similarities increased. More effective embeddings are computed for each dataset element considering their similarity to the identified CCs. These generated embeddings were evaluated for classification. In contrast, the Manifold-GCN can also export embeddings by using the output from the GCN layer before softmax. This allowed Manifold-GCN to be applied for retrieval in person Re-ID.

7. *How can contextual similarity information be incorporated into the input graph utilized by Graph Convolutional Networks (GCNs) and improve their classification results?*

   Discussion: This question is addressed by Manifold-GCN. It uses manifold learning re-ranking approaches that compute improved ranked lists that are used to build a new graph (i.e., kNN or reciprocal kNN). The experimental evaluation revealed that using this graph as the GCN input improved the classification results in all evaluated scenarios (i.e., 5 different GCNs, 3 datasets, and 4 feature extractors).

8. *Can rank-based information be utilized to measure the correlation between images more effectively?*

   Discussion: As a variation of the Jaccard Index, the Jaccard Max was proposed. Most rank-based measures are highly dependent on the neighborhood size parameter

(k). This is mitigated by identifying the depth that presents the maximum Jaccard index until a depth k. The main conjecture behind this approach is that a high overlap between ranked lists, at any depth, should be considered a strong indication of similarity.

9. *Can a correlation measure be proposed and applied to enhance image retrieval with manifold learning?*

   Discussion: The Jaccard Max was used in combination with the Correlation Graph for manifold learning. The correlation measure was utilized to weigh the graph edges. During the iterative thresholding process, for higher threshold values, only the elements with a higher correlation are connected, and vice versa. When compared to other correlation measures, Jaccard Max produced superior results.

10. *How can contextual similarity information be incorporated into metric learning, including its direct integration into losses such as contrastive loss?*

    Discussion: CCL replaces pairwise image comparison by introducing a new contextual similarity measure using neighboring elements. The CCL yields a more semantically meaningful image embedding ensuring better separability of classes in the latent space. Experimental evaluation of three datasets has shown that CCL yields superior results in classification accuracy, particularly for fewer training epochs and limited training data.

Table 11.8 – Research questions addressed by each of the proposed approaches.

	Proposed Methods						
Summarized Research Question	**DRNE**	**RQPPF**	**JaccardMax**	**HRSF**	**RFE**	**Manifold-GCN**	**CCL**
1. Unsupervised selection and fusion in Re-ID?				•			
2. Model data with contextual information for QPP?	•	•					
3. Create synthetic data with contextual similarity?	•	•					
4. Contextual similarity learning using synthetic data?	•	•					
5. How to apply more complex structures in unsupervised similarity learning?				•	•		
6. How can embeddings be generated for different tasks?					•	•	
7. Improve GCN input graph for better classification?						•	
8. More effective rank-based correlation measure?			•				
9. Correlation measure with manifold learning?			•				
10. Similarity information into contrastive loss?							•

11.3 Publications and International Fellowship

During the period of this doctorate, a total of 27 publications and submissions were produced. This section lists all publications, submissions, and works under review submitted to both conferences and international scientific journals. The first-authored publications, associated with Chapters 5 through 10 of this text, are:

1. **Lucas Pascotti Valem**, Daniel Carlos Guimarães Pedronette, and Mohand Said Allili. Contrastive Loss based on Contextual Similarity for Image Classification. In *19th International Symposium on Visual Computing (ISVC)*, 2024.
   Status: Submitted [325].

2. **Lucas Pascotti Valem**, Vanessa Helena Pereira Ferrero, and Daniel Carlos Guimarães Pedronette. Self-supervised regression for query performance prediction on image retrieval. In *2023 IEEE Sixth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 95–98, Los Alamitos, CA, USA, sep 2023.
   Status: Published [336].

3. **Lucas Pascotti Valem**, Daniel Carlos Guimarães Pedronette, and Longin Jan Latecki. Rank flow embedding for unsupervised and semi-supervised manifold learning. *IEEE Transactions on Image Processing*, 32:2811–2826, 2023.
   Status: Published [334].

4. **Lucas Pascotti Valem**, Daniel Carlos Guimarães Pedronette, and Longin Jan Latecki. Graph convolutional networks based on manifold learning for semi-supervised image classification. *Computer Vision and Image Understanding*, 227:103618, 2023.
   Status: Published [333].

5. **Lucas Pascotti Valem** and Daniel Carlos Guimarães Pedronette. Person re-id through unsupervised hypergraph rank selection and fusion. *Image Vision Computing*, 123(C), 2022.
   Status: Published [331].

6. **Lucas Pascotti Valem**, Vinicius Atsushi Sato Kawai, Vanessa Helena Pereira-Ferrero, and Daniel Carlos Guimarães Pedronette. A novel rank correlation measure for manifold learning on image retrieval and person re-id. In 2022 *IEEE International Conference on Image Processing (ICIP)*, pages 1371–1375, 2022.
   Status: Published [324].

7. **Lucas Pascotti Valem** and Daniel Carlos Guimarães Pedronette. A denoising convolutional neural network for self-supervised rank effectiveness estimation on image retrieval. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, ICMR '21, page 294–302, New York, NY, USA, 2021.
   Status: Published [330].

Resulting from works in collaboration, directly or indirectly related to contextual similarity learning and person Re-ID, there are 20 works published or accepted, listed from the most recent to the earliest:

1. V. H. P. Ferrero, T. G. Lewis, **L. P. Valem**, L. G. P. Ferrero, D. C. G. Pedronette, L. J. Latecki. Unsupervised Affinity Learning based on Manifold Analysis for Image Retrieval: A Survey. *Computer Science Review*, 2024.
Status: Published [256].

2. Gustavo Rosseto Leticio, Vinicius Sato Kawai, **Lucas Pascotti Valem**, Daniel Carlos Guimarães Pedronette, Ricardo da S. Torres. Manifold information through neighbor embedding projection for image retrieval. *Pattern Recognition Letters*, 2024.
Status: Published [167].

3. Vinicius Sato Kawai, **Lucas Pascotti Valem**, Alexandro Baldassin, Edson Borin, Daniel Carlos Guimarães Pedronette, Longin Jan Latecki. Rank-based Hashing for Effective and Efficient Nearest Neighbor Search for Image Retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2024.
Status: Published [138].

4. V. H. P. Ferrero, **L. P. Valem**, G. R. Leticio, D. C. G. Pedronette. Feature Fusion and Augmentation based on Manifold Ranking for Image Classification. *International Journal of Semantic Computing (IJSC)*, 2024.
Status: Accepted, to appear [258].

5. João Gabriel Camacho Presotto, **Lucas Pascotti Valem**, Nikolas Gomes de Sá, Daniel Carlos Guimarães Pedronette, and João Paulo Papa. Weakly supervised learning through rank-based contextual measures. *Neurocomputing*, 2024.
Status: Published [265].

6. Gustavo Leticio, **Lucas Pascotti Valem**, Leonardo Tadeu Lopes, and Daniel Carlos Guimarães Pedronette. PyUdlf: A python framework for unsupervised distance learning tasks. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM ’23, page 9680–9684, New York, NY, USA, 2023. Association for Computing Machinery.
Status: Published [166].

7. V. Pereira-Ferrero, **L. P. Valem**, G. Leticio, and D. Pedronette. Feature fusion and augmentation based on manifold ranking for image classification. In *2023 IEEE Sixth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 75–82, Los Alamitos, CA, USA, sep 2023. IEEE Computer Society. **Status: Published [257].**

8. Vanessa Helena Pereira-Ferrero, **Lucas Pascotti Valem**, and Daniel Carlos Guimarães Pedronette. Feature augmentation based on manifold ranking and LSTM for image

classification. *Expert Systems with Applications*, 213:118995, 2023.
Status: Published [259].

9. Lucas Barbosa de Almeida, **Lucas Pascotti Valem**, and Daniel Carlos Guimarães Pedronette. Graph convolutional networks and manifold ranking for multimodal video retrieval. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2811–2815, 2022.
Status: Published [63].

10. João Gabriel Camacho Presotto, Samuel Felipe dos Santos, **Lucas Pascotti Valem**, Fabio Augusto Faria, João Paulo Papa, Jurandy Almeida, and Daniel Carlos Guimarães Pedronette. Weakly supervised learning based on hypergraph manifold ranking. *Journal of Visual Communication and Image Representation*, 89:103666, 2022.
Status: Published [263].

11. Filipe Alves de Fernando, Daniel Carlos Guimarães Pedronette, Gustavo José de Sousa, **Lucas Pascotti Valem**, and Ivan Rizzo Guilherme. Rade+: A semantic rank-based graph embedding algorithm. *International Journal of Information Management Data Insights*, 2(1):100078, 2022.
Status: Published [65].

12. Claudio Filipi Gonçalves dos Santos, Diego de Souza Oliveira, Leandro A. Passos, Rafael Gonçalves Pires, Daniel Felipe Silva Santos, **Lucas Pascotti Valem**, Thierry P. Moreira, Marcos Cleison S. Santana, Mateus Roder, João Paulo Papa, and Danilo Colombo. Gait recognition based on deep learning: A survey. *ACM Comput. Surv.*, 55(2), jan 2022.
Status: Published [76].

13. Daniel Carlos Guimarães Pedronette, **Lucas Pascotti Valem**, and Longin Jan Latecki. Efficient rank-based diffusion process with assured convergence. *Journal of Imaging*, 7(3), 2021.
Status: Published [252].

14. Daniel Carlos Guimarães Pedronette, **Lucas Pascotti Valem**, and Ricardo da S. Torres. A bfs-tree of ranking references for unsupervised manifold learning. *Pattern Recognition*, 111:107666, 2021.
Status: Published [253].

15. Nikolas Gomes de Sá, **Lucas Pascotti Valem**, and Daniel Carlos Guimarães Pedronette. A multi-level rank correlation measure for image retrieval. In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021)* - Volume 5: VISAPP, pages 370–378. INSTICC, SciTePress, 2021.
Status: Published [68].

16. Lucas Barbosa de Almeida, Vanessa Helena Pereira-Ferrero, **Lucas Pascotti Valem**, Jurandy Almeida, and Daniel Carlos Guimarães Pedronette. Representation learning

for image retrieval through 3D CNN and manifold ranking. In *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 417–424, 2021.
Status: Published [62].

17. João Gabriel Camacho Presotto, **Lucas Pascotti Valem**, Nikolas Gomes de Sá, Daniel Carlos Guimarães Pedronette, and João Paulo Papa. Weakly supervised learning through rank-based contextual measures. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5752–5759, 2021.
Status: Published [264].

18. Filipe Alves de Fernando, Daniel Carlos Guimarães Pedronette, Gustavo José de Sousa, **Lucas Pascotti Valem**, and Ivan Rizzo Guilherme. Rade: A rank-based graph embedding approach. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020)* - Volume 5: VISAPP, pages 142–152. INSTICC, SciTePress, 2020.
Status: Published [64].

19. Leonardo Tadeu Lopes, **Lucas Pascotti Valem**, Daniel Carlos Guimarães Pedronette, Ivan Rizzo Guilherme, João Paulo Papa, Marcos Cleison Silva Santana, and Danilo Colombo. Manifold learning-based clustering approach applied to anomaly detection in surveillance videos. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* - Volume 5: VISAPP, pages 404–412. INSTICC, SciTePress, 2020.
Status: Published [204].

20. Flávia Pisani, **Lucas Pascotti Valem**, Daniel Carlos Guimarães Pedronette, Ricardo da S. Torres, Edson Borin, and Mauricio Breternitz Jr. A unified model for accelerating unsupervised iterative re-ranking algorithms. *Concurrency and Computation: Practice and Experience*, 32(14):e5702, 2020.
Status: Published [261].

Figure 11.3 presents a diagram with all the works that originated from collaborations and their respective publications. It also shows how they connect to the concepts and are directly or indirectly related to contextual similarity learning or person Re-ID.

- **International Fellowship**

During the course of the Ph.D. program, the student received a *"Fulbright Doctoral Dissertation Research Abroad Award"*, conducted under the supervision of:

- Professor Longin Jan Latecki from Temple University (Philadelphia, Pennsylvania, United States of America), which contributed to the research projects involving the RFE [334] and Manifold-GCN [333] approaches.

The fellowship covered a 9-month stay in Philadelphia from September 2023 to May 2024 to conduct the research at Temple University.

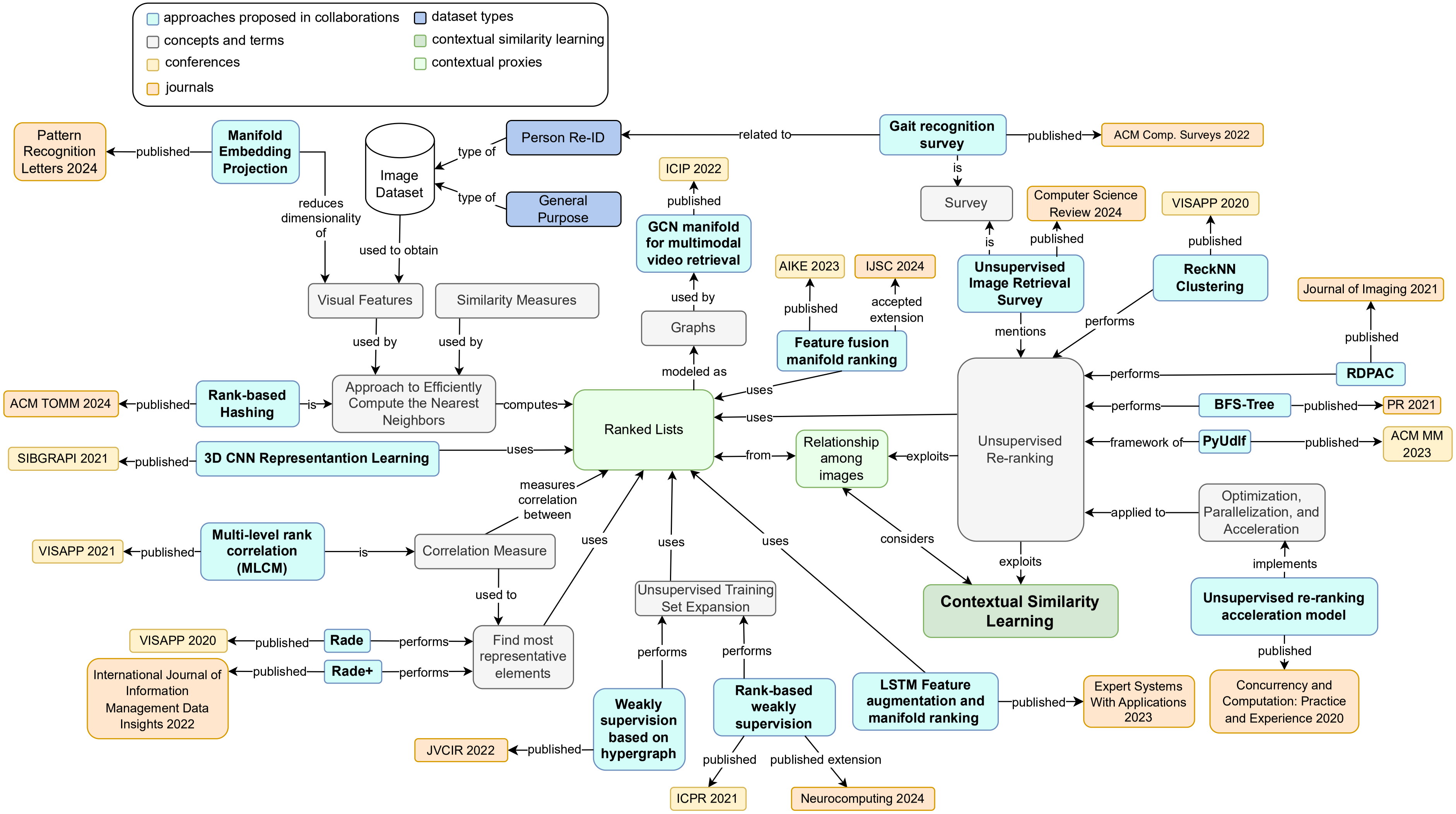

approaches proposed in collaborations
concepts and terms
conferences
journals
dataset types
contextual similarity learning
contextual proxies
Pattern Recognition Letters 2024
published
Manifold Embedding Projection
reduces dimensionality of
Image Dataset
type of
type of
Person Re-ID
General Purpose
related to
Gait recognition survey
published
ACM Comp. Surveys 2022
is
Survey
ICIP 2022
published
GCN manifold for multimodal video retrieval
used by
Graphs
modeled as
used to obtain
Visual Features
Similarity Measures
used by
used by
ACM TOMM 2024
published
Rank-based Hashing
is
Approach to Efficiently Compute the Nearest Neighbors
computes
Ranked Lists
SIBGRAPI 2021
published
3D CNN Representantion Learning
uses
AIKE 2023
IJSC 2024
published
accepted extension
Feature fusion manifold ranking
uses
uses
from
Relationship among images
exploits
Computer Science Review 2024
VISAPP 2020
is
published
published
Unsupervised Image Retrieval Survey
ReckNN Clustering
mentions
performs
Journal of Imaging 2021
published
RDPAC
performs
Unsupervised Re-ranking
performs
BFS-Tree
published
PR 2021
framework of
PyUdlf
published
ACM MM 2023
applied to
Optimization, Parallelization, and Acceleration
implements
Unsupervised re-ranking acceleration model
published
Concurrency and Computation: Practice and Experience 2020
exploits
Contextual Similarity Learning
considers
measures correlation between
VISAPP 2021
published
Multi-level rank correlation (MLCM)
is
Correlation Measure
used to
uses
uses
uses
Unsupervised Training Set Expansion
VISAPP 2020
published
Rade
performs
International Journal of Information Management Data Insights 2022
published
Rade+
performs
Find most representative elements
performs
performs
Weakly supervision based on hypergraph
JVCIR 2022
published
Rank-based weakly supervision
published
published extension
ICPR 2021
Neurocomputing 2024
LSTM Feature augmentation and manifold ranking
published
Expert Systems With Applications 2023

Figure 11.3 – Published and submitted collaborations and their connection to terms and concepts related to this dissertation.

11.4 Code Availability

As of the writing of this text, the RFE [334] and JaccardMax [324] implementations are available in the open-source Unsupervised Distance Learning Framework [326] (UDLF) [16]. The UDLF is a C/C++ framework of unsupervised distance learning methods for image and multimedia retrieval tasks, it currently implements eleven different approaches and is continuously maintained. As a result of a collaboration, a Python wrapper was proposed for UDLF, the pyUDLF [166], which is also accessible from Github [17].

It is intended that the code for the other proposed approaches will be made available to facilitate further research and development. This will also enable the scientific community to replicate the results.

11.5 Future Work

Given the diversity of challenges and contributions discussed in this work, various future research directions are possible:

1. **Evaluate for multi-query Re-ID**: All the results were presented for single-query Re-ID, where the query consists of only a single image for each search. For future work, the proposed methods can be adapted for scenarios where multiple queries are provided as input.

2. **Investigate the impact of Re-ID detectors**: Experiments can be conducted to evaluate and investigate the impact of multiple Re-ID detectors on the effectiveness of the proposed methods. This can also help to explain the differences in results on Market and Duke datasets, for example.

3. **Other types of multimedia content**: In this work, all the evaluations were performed using image data. However, most of the proposed methods are flexible and work using features and ranked lists. Therefore, the features could be extracted from other types of multimedia content (e.g., sound, video, text, and others) and the proposed methods could also be evaluated in these scenarios.

4. **Improve synthetic data generation**: Both DRNE and RQPPF are trained on synthetic data. Investigations of approaches that could improve the generation of this data could be conducted.

5. **Employ other denoising networks for DRNE**: The proposed DRNE revealed that ranked lists can be represented as images and a denoising network can be

[16] `github.com/UDLF/UDLF`

[17] `github.com/UDLF/pyUDLF`

utilized to perform query performance prediction. This method was validated using the DnCNN, and investigations with other denoising networks could be conducted.

6. **Other measures to compute RQPPF meta-features**: Aiming at encoding information from the reciprocal neighborhood to build the meta-features, the proposed RQPPF mainly utilized Authority and Reciprocal Density effectiveness estimation scores. Other measures could be employed, including the HQPP used by the proposed HRSF approach. Even the DRNE could be combined with RQPPF.

7. **Expand HRSF selection with other QPP approaches**: For selecting the best ranked lists from different descriptors, HRSF uses HQPP. For future work, other measures can be considered, including the ones proposed, the DRNE and RQPPF.

8. **Adapt and use Jaccard Max for selection in HRSF**: It is highly desirable to combine ranked lists that show high complementarity. For this purpose, HRSF employs a correlation measure for selection. The Jaccard Max could be adapted to compare ranked lists of the same image and be used as part of the HRSF workflow.

9. **Research strategies to export embeddings from GCNs**: This work exports embeddings from the GCNs by considering the output of the last layer before the softmax operation. However, the experimental results revealed that the embeddings using this strategy are not adequate. This was verified when comparing the semi-supervised Re-ID results of Manifold-GCN with the other unsupervised approaches. An investigation can be conducted to research other strategies to obtain embeddings from GCNs.

10. **Combine RFE and Manifold-GCN**: The Manifold-GCN utilizes manifold learning techniques to construct the input graphs for the GCNs. However, the input features are those generated by the descriptors. One potential approach is to, instead of using these features, use the embeddings generated by the RFE as the input for the GCNs in Manifold-GCN.

11. **Utilize RFE embeddings for other tasks**: This work evaluated the RFE embeddings for image classification. However, these embeddings could be used in many other tasks (e.g., clustering, retrieval).

12. **Generate pseudo-labels with RFE**: Besides the embeddings, the RFE also outputs the connected components (CC) found by the algorithm. Images that belong to the same CC usually belong to the same class. An investigation could be conducted to use these CCs to generate pseudo-labels.

13. **Adapt CCL for semi-supervised scenarios**: As the training progresses, unlabeled data that are classified with a higher probability of belonging to a certain class can

be incorporated into the training process. This idea could be used to turn CCL applicable to semi-supervised scenarios.

14. **Integrate the neighborhood information in other contrastive losses**: The CCL incorporates neighborhood information into a supervised contrastive loss. However, an investigation could be conducted to adapt and incorporate this idea in other types of contrastive losses.

15. **Utilize CCL embeddings for retrieval**: The embeddings produced by Contrastive Cluster Learning (CCL) could be applied to a wider range of tasks beyond image classification, including image retrieval. This would expand the potential applications of CCL to tasks such as Re-ID. To enhance its effectiveness in these areas, further research is needed to analyze and improve the separability margins of classes within the latent space.

Table 11.9 lists the possible future work alongside proposed approaches, using bullet points to indicate when they are related.

Table 11.9 – Future work related to each of the proposed approaches.

	Proposed Methods						
Future Work	**DRNE**	**RQPPF**	**JaccardMax**	**HRSF**	**RFE**	**Manifold-GCN**	**CCL**
1. Multi-query Re-ID	●	●	●	●	●	●	
2. Impact of Re-ID detectors			●	●	●	●	
3. Other types of multimedia content	●	●	●	●	●	●	●
4. Improve synthetic data	●	●					
5. Other denoising networks	●						
6. Improve meta-features		●					
7. HRSF with other QPP approaches	●	●		●			
8. JaccardMax with HRSF			●	●			
9. Strategies to export embeddings from GCNs						●	
10. Combine RFE and Manifold-GCN					●	●	
11. RFE embeddings for other tasks					●		
12. Generate pseudo-labels with RFE					●		
13. CCL for semi-supervision							●
14. Neighborhood into other contrastive losses							●
15. CCL embeddings for retrieval							●

Bibliography

[1] Agarwal, M. and Mostafa, J. (2011). Content-based image retrieval for alzheimer's disease detection. In *2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 13–18.

[2] Albawi, S.; Mohammed, T. A.; and Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6.

[3] Ali, N.; Zafar, B.; Iqbal, M. K.; Sajid, M.; Younis, M. Y.; Dar, S. H.; Mahmood, M. T.; and Lee, I. H. (2019). Modeling global geometric spatial information for rotation invariant classification of satellite images. *PLoS One*, 14(7):e0219833.

[4] Alnissany, A. and Dayoub, Y. (2023). Modified centroid triplet loss for person re-identification. *Journal of Big Data*, 10(1):74.

[5] Alqasemi, F. A.; Alabbasi, H. Q.; Sabeha, F. G.; Alawadhi, A.; Kahlid, S.; and Zahary, A. (2019). Feature selection approach using knn supervised learning for content-based image retrieval. In *2019 First International Conference of Intelligent Computing and Engineering (ICOICE)*, pages 1–5.

[6] Alves, C. and Traina, A. J. M. (2022). Variational autoencoders for medical image retrieval. In *2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–6.

[7] Alvin, Y. H. Y. and Chakraborty, D. (2023). Approximate Maximum Rank Aggregation: Beyond the Worst-Case. In Bouyer, P. and Srinivasan, S., editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2023)*, volume 284 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:21, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[8] Amorim, W. P.; Falcão, A. X.; and d. Carvalho, M. H. (2014). Semi-supervised pattern classification using optimum-path forest. In *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 111–118.

[9] An, L.; Chen, X.; Yang, S.; and Li, X. (2017). Person re-identification by multi-hypergraph fusion. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2763–2774.

[10] Anand, A.; Leonhardt, J.; Rudra, K.; and Anand, A. (2022). Supervised contrastive learning approach for contextual ranking. In *Proceedings of the 2022 ACM SIGIR International Conference on Theory of Information Retrieval*, ICTIR '22, page 61–71, New York, NY, USA. Association for Computing Machinery.

[11] Antelmi, A.; Cordasco, G.; Polato, M.; Scarano, V.; Spagnuolo, C.; and Yang, D. (2023). A survey on hypergraph representation learning. *ACM Comput. Surv.*, 56(1).

[12] Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; and Sivic, J. (2016). Netvlad: Cnn architecture for weakly supervised place recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5297–5307.

[13] Arica, N. and Vural, F. T. Y. (2003). BAS: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recognition Letters*, 24(9-10):1627–1639.

[14] Awad, M. and Khanna, R. (2015). *Machine Learning*, pages 1–18. Apress, Berkeley, CA.

[15] Baeza-Yates, R. and Ribeiro-Neto, B. (2013). *Recuperação de Informação: Conceitos e Tecnologia das Máquinas de Busca*. Editora Bookman.

[16] Bai, S. and Bai, X. (2016). Sparse contextual activation for efficient visual re-ranking. *IEEE Trans. on Image Processing (TIP)*, 25(3):1056–1069.

[17] Bai, S.; Bai, X.; Tian, Q.; and Latecki, L. J. (2017). Regularized diffusion process for visual retrieval. In *Conf. on Artificial Intelligence (AAAI)*, pages 3967–3973.

[18] Bai, S.; Bai, X.; Tian, Q.; and Latecki, L. J. (2019). Regularized diffusion process on bidirectional context for object retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(5):1213–1226.

[19] Bai, S.; Zhang, F.; and Torr, P. H. (2021a). Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637.

[20] Bai, S.; Zhou, Z.; Wang, J.; Bai, X.; Latecki, L. J.; and Tian, Q. (2017). Ensemble diffusion for retrieval. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 774–783.

[21] Bai, X.; Bai, S.; and Wang, X. (2015). Beyond diffusion process: Neighbor set similarity for fast re-ranking. *Information Sciences*, 325:342 – 354.

[22] Bai, Z.; Wang, Z.; Wang, J.; Hu, D.; and Ding, E. (2021b). Unsupervised multi-source domain adaptation for person re-identification. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12909–12918.

[23] Bao, L.-N.; Wei, L.; Qiu, X.; gang Zhou, W.; Li, H.; and Tian, Q. (2023). Learning transferable pedestrian representation from multimodal information supervision. *ArXiv*, abs/2304.05554.

[24] Barz, B. and Denzler, J. (2021). Content-based image retrieval and the semantic gap in the deep learning era. In Del Bimbo, A.; Cucchiara, R.; Sclaroff, S.; Farinella, G. M.; Mei, T.; Bertini, M.; Escalante, H. J.; and Vezzani, R., editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 245–260, Cham. Springer International Publishing.

[25] Bedagkar-Gala, A. and Shah, S. K. (2014). A survey of approaches and trends in person re-identification. *Image and Vision Computing*, 32(4):270 – 286.

[26] Berman, M.; Jégou, H.; Andrea, V.; Kokkinos, I.; and Douze, M. (2019). MultiGrain: a unified image embedding for classes and instances. *arXiv e-prints*.

[27] Berthelot, D.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Sohn, K.; Zhang, H.; and Raffel, C. (2020). Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*.

[28] Berthelot, D.; Carlini, N.; Goodfellow, I. J.; Papernot, N.; Oliver, A.; and Raffel, C. (2019). Mixmatch: A holistic approach to semi-supervised learning. *CoRR*, abs/1905.02249.

[29] Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; and Upcroft, B. (2016). Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468.

[30] Bianchi, F. M.; Grattarola, D.; Livi, L.; and Alippi, C. (2021). Graph neural networks with convolutional arma filters. *IEEE TPAMI*, pages 1–1.

[31] Black, E. and Fredrikson, M. (2021). Leave-one-out unfairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 285–295, New York, NY, USA. Association for Computing Machinery.

[32] Bolme, D. S.; Beveridge, J. R.; Draper, B. A.; and Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2544–2550.

[33] Bossard, L.; Guillaumin, M.; and Van Gool, L. (2014). Food-101 – mining discriminative components with random forests. In Fleet, D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 446–461, Cham. Springer International Publishing.

[34] Bretto, A. (2013). *Hypergraph Theory: An Introduction.* Springer International Publishing.

[35] Brodatz, P. (1966). *Textures: A Photographic Album for Artists and Designers.* Dover.

[36] Cai, D.; Zhang, C.; and He, X. (2010). Unsupervised feature selection for multi-cluster data. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, page 333–342, New York, NY, USA. Association for Computing Machinery.

[37] Camps, O.; Gou, M.; Hebble, T.; Karanam, S.; Lehmann, O.; Li, Y.; Radke, R. J.; Wu, Z.; and Xiong, F. (2017). From the lab to the real world: Re-identification in an airport camera network. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):540–553.

[38] Chakraborty, D.; Das, S.; Khan, A.; and Subramanian, A. (2022). Fair rank aggregation. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2022).*

[39] Chang, X.; Hospedales, T. M.; and Xiang, T. (2018). Multi-level factorisation net for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[40] Chatzichristofis, S. A. and Boutalis, Y. S. (2008a). Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *Proceedings of the 6th international conference on Computer vision systems*, ICVS'08, pages 312–322.

[41] Chatzichristofis, S. A. and Boutalis, Y. S. (2008b). Fcth: Fuzzy color and texture histogram - a low level feature for accurate image retrieval. In *WIAMIS*, pages 191–196.

[42] Chaudhuri, U.; Banerjee, B.; and Bhattacharya, A. (2019). Siamese graph convolutional network for content based remote sensing image retrieval. *Computer Vision and Image Understanding*, 184:22–30.

[43] Chen, H.; Lagadec, B.; and Bremond, F. (2021a). Ice: Inter-instance contrastive encoding for unsupervised person re-identification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14940–14949.

[44] Chen, H.; Wang, Y.; Lagadec, B.; Dantcheva, A.; and Bremond, F. (2021b). Joint generative and contrastive learning for unsupervised person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2004–2013.

[45] Chen, S.-B.; Tian, X.-Z.; Ding, C. H. Q.; Luo, B.; Liu, Y.; Huang, H.; and Li, Q. (2020a). Graph convolutional network based on manifold similarity learning. *Cognitive Computation*, 12(6):1144.

[46] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

[47] Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. (2020b). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

[48] Chen, W.; Liu, Y.; Wang, W.; Bakker, E. M.; Georgiou, T.; Fieguth, P. W.; Liu, L.; and Lew, M. S. (2021c). Deep image retrieval: A survey. *CoRR*, abs/2101.11282.

[49] Chen, X. and He, K. (2021). Exploring simple siamese representation learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15745–15753.

[50] Chen, X. and Li, Y. (2020). Deep feature learning with manifold embedding for robust image retrieval. *Algorithms*, 13(12).

[51] Chen, Y.; Li, J.; Xiao, H.; Jin, X.; Yan, S.; and Feng, J. (2017). Dual path networks. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4467–4475. Curran Associates, Inc.

[52] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807.

[53] Cieplinski, L. (2001). Mpeg-7 color descriptors and their applications. In Skarbek, W., editor, *Computer Analysis of Images and Patterns*, pages 11–20, Berlin, Heidelberg. Springer Berlin Heidelberg.

[54] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.*, 20(3):273–297.

[55] Cristani, M. and Murino, V. (2018). Chapter 10 - person re-identification. In Chellappa, R. and Theodoridis, S., editors, *Academic Press Library in Signal Processing, Volume 6*, pages 365 – 394. Academic Press.

[56] Cronen-Townsend, S.; Zhou, Y.; and Croft, W. B. (2002). Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 299–306.

[57] Dabov, K.; Foi, A.; Katkovnik, V.; and Egiazarian, K. (2007). Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095.

[58] Dai, J.; Zhang, P.; Lu, H.; and Wang, H. (2018). Video person re-identification by temporal residual learning. *IEEE Transactions on Image Processing*, PP.

[59] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1.

[60] Datta, R.; Joshi, D.; Li, J.; and Wang, J. Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):5:1–5:60.

[61] Datta, S.; Ganguly, D.; Greene, D.; and Mitra, M. (2022). Deep-qpp: A pairwise interaction-based deep learning model for supervised query performance prediction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 201–209, New York, NY, USA. Association for Computing Machinery.

[62] De Almeida, L. B.; Pereira-Ferrero, V. H.; Valem, L. P.; Almeida, J.; and Pedronette, D. C. G. (2021). Representation learning for image retrieval through 3d cnn and manifold ranking. In *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 417–424.

[63] De Almeida, L. B.; Valem, L. P.; and Pedronette, D. C. G. (2022). Graph convolutional networks and manifold ranking for multimodal video retrieval. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2811–2815.

[64] De Fernando, F. A.; Pedronette, D. C. G.; de Sousa, G. J.; Valem, L. P.; and Guilherme, I. R. (2020). Rade: A rank-based graph embedding approach. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020) - Volume 5: VISAPP*, pages 142–152. INSTICC, SciTePress.

[65] De Fernando, F. A.; Pedronette, D. C. G.; de Sousa, G. J.; Valem, L. P.; and Guilherme, I. R. (2022). Rade+: A semantic rank-based graph embedding algorithm. *International Journal of Information Management Data Insights*, 2(1):100078.

[66] De Oliveira, A. A.; Oakley, E.; Torres, R. d. S.; and Rocha, A. (2019). Relevance prediction in similarity-search systems using extreme value theory. *J. Vis. Commun. Image Represent.*, 60:236–249.

[67] De Sá, N. G.; Valem, L. P.; and Pedronette, D. C. G. (2021a). A multi-level rank correlation measure for image retrieval. In *VISIGRAPP (5: VISAPP)*, pages 370–378.

[68] De Sá, N. G.; Valem, L. P.; and Pedronette, D. C. G. (2021b). A multi-level rank correlation measure for image retrieval. In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 5: VISAPP*, pages 370–378. INSTICC, SciTePress.

[69] Delvinioti, A.; Jégou, H.; Amsaleg, L.; and Houle, M. E. (2014). Image retrieval with reciprocal and shared nearest neighbors. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 2, pages 321–328.

[70] Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[71] Deng, W.; Zheng, L.; Ye, Q.; Kang, G.; Yang, Y.; and Jiao, J. (2018). Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *CVPR*.

[72] Dewil, V.; Barral, A.; Facciolo, G.; and Arias, P. (2022). Self-supervision versus synthetic datasets: which is the lesser evil in the context of video denoising? In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4896–4906, Los Alamitos, CA, USA. IEEE Computer Society.

[73] Dollár, P.; Appel, R.; Belongie, S.; and Perona, P. (2014). Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545.

[74] Donoser, M. and Bischof, H. (2013). Diffusion processes for retrieval revisited. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1320–1327.

[75] Dorogush, A. V.; Ershov, V.; and Gulin, A. (2018). Catboost: gradient boosting with categorical features support.

[76] Dos Santos, C. F. G.; de Souza Oliveira, D.; Passos, L. A.; Pires, R. G.; Santos, D. F. S.; Valem, L. P.; Moreira, T. P.; Santana, M. C. S.; Roder, M.; Papa, J. P.; and Colombo, D. (2022). Gait recognition based on deep learning: A survey. *ACM Comput. Surv.*, 55(2).

[77] Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

[78] Dou, Z.; Cui, H.; Zhang, L.; and Wang, B. (2020). Learning global and local consistent representations for unsupervised image retrieval via deep graph diffusion networks. *arXiv preprint arXiv:2001.01284*.

[79] Drucker, H.; Burges, C. J. C.; Kaufman, L.; Smola, A.; and Vapnik, V. (1996). Support vector regression machines. In Mozer, M.; Jordan, M.; and Petsche, T., editors, *Advances in Neural Information Processing Systems*, volume 9. MIT Press.

[80] Dubey, S. R. (2022). A decade survey of content based image retrieval using deep learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(5):2687–2704.

[81] Duncanson, K. A.; Thwaites, S.; Booth, D.; Hanly, G.; Robertson, W. S. P.; Abbasnejad, E.; and Thewlis, D. (2023). Deep metric learning for scalable gait-based person re-identification using force platform data. *Sensors*, 23(7).

[82] Dwibedi, D.; Aytar, Y.; Tompson, J.; Sermanet, P.; and Zisserman, A. (2021). With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9568–9577.

[83] Dwork, C.; Kumar, R.; Naor, M.; and Sivakumar, D. (2001). Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, page 613–622, New York, NY, USA. Association for Computing Machinery.

[84] Elsayed, G.; Krishnan, D.; Mobahi, H.; Regan, K.; and Bengio, S. (2018). Large margin deep networks for classification. In *Advances in neural information processing systems*, pages 842–852.

[85] Engelen, J. E. V. and Hoos, H. H. (2019). A survey on semi-supervised learning. *Machine Learning*, 109:373–440.

[86] Escolano, F.; Hancock, E. R.; Lozano, M. A.; and Curado, M. (2017). The mutual information between graphs. *Pattern Recognition Letters*, 87:12 – 19.

[87] Fagin, R.; Kumar, R.; and Sivakumar, D. (2003). Comparing top k lists. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, page 28–36, USA. Society for Industrial and Applied Mathematics.

[88] Figueira, D.; Bazzani, L.; Minh, H. Q.; Cristani, M.; Bernardino, A.; and Murino, V. (2013). Semi-supervised multi-feature learning for person re-identification. In *2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 111–116.

[89] Fogel, I. and Sagi, D. (1989). Gabor filters as texture discriminator. *Biological Cybernetics*, 61(2):103–113.

[90] Franceschi, L.; Niepert, M.; Pontil, M.; and He, X. (2019). Learning discrete structures for graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning.*

[91] Fredriksson, T.; Mattos, D. I.; Bosch, J.; and Olsson, H. H. (2020). Data labeling: An empirical investigation into industrial challenges and mitigation strategies. In Morisio, M.; Torchiano, M.; and Jedlitschka, A., editors, *Product-Focused Software Process Improvement*, pages 202–216, Cham. Springer International Publishing.

[92] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232.

[93] Fu, Z.; Li, Y.; Mao, Z.; Wang, Q.; and Zhang, Y. (2021). Deep metric learning with self-supervised ranking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(2):1370–1378.

[94] García, J.; Martinel, N.; Gardel, A.; Bravo, I.; Foresti, G. L.; and Micheloni, C. (2016). Modeling feature distances by orientation driven classifiers for person re-identification. *Journal of Visual Communication and Image Representation*, 38:115 – 129.

[95] García, J.; Martinel, N.; Gardel, A.; Bravo, I.; Foresti, G. L.; and Micheloni, C. (2017). Discriminant context information analysis for post-ranking person re-identification. *IEEE Transactions on Image Processing*, 26(4):1650–1665.

[96] García, J.; Martinel, N.; Micheloni, C.; and Gardel, A. (2015). Person re-identification ranking optimisation by discriminant context information analysis. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1305–1313.

[97] Ge, W.; Pan, C.; Wu, A.; Zheng, H.; and Zheng, W.-S. (2021). *Cross-Camera Feature Prediction for Intra-Camera Supervised Person Re-Identification across Distant Scenes*, page 3644–3653. Association for Computing Machinery, New York, NY, USA.

[98] Geusebroek, J.-M.; Burghouts, G. J.; and Smeulders, A. W. M. (2005). The amsterdam library of object images. *International Journal of Computer Vision*, 61(1):103–112.

[99] Gionis, A.; Indyk, P.; and Motwani, R. (1999). Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, page 518–529.

[100] Giovanni, F. D.; Rowbottom, J.; Chamberlain, B. P.; Markovich, T.; and Bronstein, M. M. (2022). Graph neural networks as gradient flows: understanding graph convolutions via energy. *arXiv preprint arXiv:2206.10991.*

[101] Gong, C.; Wang, D.; and Liu, Q. (2021). Alphamatch: Improving consistency for semi-supervised learning with alpha-divergence. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13678–13687.

[102] Gopalakrishnan, S.; Sridharan, S.; Nayak, S. R.; Nayak, J.; and Venkataraman, S. (2022). Central hubs prediction for bio networks by directed hypergraph - ga with validation to covid-19 ppi. *Pattern Recognition Letters*, 153:246–253.

[103] Gopalan, R.; Turaga, P.; and Chellappa, R. (2010). Articulation-invariant representation of non-planar shapes. In *ECCV*, volume 3, pages 286–299.

[104] Gordo, A.; Almazán, J.; Revaud, J.; and Larlus, D. (2017). End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124.

[105] Grabner, H.; Grabner, M.; and Bischof, H. (2006). Real-time tracking via on-line boosting. In Chantler, M. J.; Fisher, R. B.; and Trucco, E., editors, *BMVC*, pages 47–56. British Machine Vision Association.

[106] Gray, D. and Tao, H. (2008). Viewpoint invariant pedestrian recognition with an ensemble of localized features. In Forsyth, D.; Torr, P.; and Zisserman, A., editors, *Computer Vision – ECCV 2008*, pages 262–275, Berlin, Heidelberg. Springer Berlin Heidelberg.

[107] Gui, J.; Chen, T.; Zhang, J.; Cao, Q.; Sun, Z.; Luo, H.; and Tao, D. (2023). A survey on self-supervised learning: Algorithms, applications, and future trends.

[108] Guo, R.; Li, C.; Li, Y.; and Lin, J. (2018). Density-adaptive kernel based re-ranking for person re-identification. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 982–987.

[109] Hammoudeh, Z. and Lowd, D. (2024). Training data influence analysis and estimation: a survey. *Machine Learning*, 113(5):2351–2403.

[110] He, K.; Zhang, X.; Ren, S.; and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[111] He, S.; Luo, H.; Wang, P.; Wang, F.; Li, H.; and Jiang, W. (2021). Transreid: Transformer-based object re-identification. In *ICCV*, pages 15013–15022.

[112] He, X.; Cai, D.; and Niyogi, P. (2005). Laplacian score for feature selection. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS'05, pages 507–514, Cambridge, MA, USA. MIT Press.

[113] Henriques, J. F.; Caseiro, R.; Martins, P.; and Batista, J. (2015). High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596.

[114] Hermans, A.; Beyer, L.; and Leibe, B. (2017). In Defense of the Triplet Loss for Person Re-Identification. *arXiv preprint arXiv:1703.07737*.

[115] Hoi, S. C.; Liu, W.; and Chang, S.-F. (2010). Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Transactions on Multimedia Computing and Communication Applications*, 6(3):18:1–18:26.

[116] Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5:1457–1469.

[117] Hu, J.; Shen, L.; and Sun, G. (2018). Squeeze-and-excitation networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[118] Huang, H.; Yang, W.; Chen, X.; Zhao, X.; Huang, K.; Lin, J.; Huang, G.; and Du, D. (2018). Eanet: Enhancing alignment for cross-domain person re-identification. *arXiv preprint arXiv:1812.11369*.

[119] Huang, J.; Kumar, S. R.; Mitra, M.; Zhu, W.-J.; and Zabih, R. (1997). Image indexing using color correlograms. In *CVPR*, pages 762–768.

[120] Huang, Y.; Liu, Q.; Zhang, S.; and Metaxas, D. N. (2010). Image retrieval via probabilistic hypergraph ranking. In *IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR'10)*, pages 3376–3383.

[121] Huang, Y.; Peng, P.; Jin, Y.; Xing, J.; Lang, C.; and Feng, S. (2019). Domain adaptive attention model for unsupervised cross-domain person re-identification. *CoRR*, abs/1905.10529.

[122] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org.

[123] Iscen, A.; Avrithis, Y.; Tolias, G.; Furon, T.; and Chum, O. (2018a). Fast spectral ranking for similarity search. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7632–7641.

[124] Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. (2018b). Mining on manifolds: Metric learning without labels. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7642–7651.

[125] Iscen, A.; Tolias, G.; Avrithis, Y.; Furon, T.; and Chum, O. (2017). Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *CVPR*.

[126] Izenman, A. J. (2008). *Linear Discriminant Analysis*, pages 237–280. Springer New York, New York, NY.

[127] Jegou, H.; Douze, M.; and Schmid, C. (2008). Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, ECCV '08, pages 304–317.

[128] Jegou, H.; Schmid, C.; Harzallah, H.; and Verbeek, J. (2010). Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):2–11.

[129] Jensen, J. L. W. V. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193.

[130] Jia, Q. and Tian, X. (2015). Query difficulty estimation via relevance prediction for image retrieval. *Signal Processing*, 110:232–243.

[131] Jia, Q.; Tian, X.; and Mei, T. (2014). Query difficulty estimation via pseudo relevance feedback for image search. In *2014 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

[132] Jia, X.; Zhong, X.; Ye, M.; Liu, W.; and Huang, W. (2022). Complementary data augmentation for cloth-changing person re-identification. *IEEE Transactions on Image Processing*, 31:4227–4239.

[133] Jiang, J.; Wang, B.; and Tu, Z. (2011). Unsupervised metric learning by self-smoothing operator. In *2011 International Conference on Computer Vision*, pages 794–801.

[134] Jin, X.; Lan, C.; Zeng, W.; Chen, Z.; and Zhang, L. (2020). Style normalization and restitution for generalizable person re-identification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3140–3149.

[135] Ju, W.; Yi, S.; Wang, Y.; Xiao, Z.; Mao, Z.; Li, H.; Gu, Y.; Qin, Y.; Yin, N.; Wang, S.; Liu, X.; Luo, X.; Yu, P. S.; and Zhang, M. (2024). A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges. *arXiv preprint arXiv:2403.04468*.

[136] Juan, D.; Lu, C.; Li, Z.; Peng, F.; Timofeev, A.; Chen, Y.; Gao, Y.; Duerig, T.; Tomkins, A.; and Ravi, S. (2019). Graph-rise: Graph-regularized image semantic embedding. *CoRR*, abs/1902.10814.

[137] Karanam, S.; Gou, M.; Wu, Z.; Rates-Borras, A.; Camps, O.; and Radke, R. J. (2019). A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3):523–536.

[138] Kawai, V. S.; Valem, L. P.; Baldassin, A.; Borin, E.; Pedronette, D. C. G.; and Latecki, L. J. (2024). Rank-based hashing for effective and efficient nearest neighbor search for image retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications.*

[139] Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3149–3157, Red Hook, NY, USA. Curran Associates Inc.

[140] Khasanova, A. M. and Pasechnik, M. O. (2021). Social media analysis with machine learning. In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pages 32–35.

[141] Khemani, B.; Patil, S.; Kotecha, K.; and Tanwar, S. (2024). A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, 11(1):18.

[142] Khosla, A.; Jayadevaprakash, N.; Yao, B.; and Fei-Fei, L. (2011). Novel dataset for fine-grained image categorization. In *Workshop on Fine-Grained Visual Categorization, CVPR.*

[143] Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. (2020). Supervised contrastive learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

[144] Khraimeche, Y.; Bilodeau, G.-A.; Steele, D.; and Mahadik, H. (2020). Unsupervised disentanglement gan for domain adaptive person re-identification. *ArXiv*, abs/2007.15560.

[145] Kim, B.; Choo, J.; Kwon, Y.; Joe, S.; Min, S.; and Gwon, Y. (2021). Selfmatch: Combining contrastive self-supervision and consistency for semi-supervised learning. *CoRR*, abs/2101.06480.

[146] Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations,*

ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net.

[147] Klicpera, J.; Bojchevski, A.; and Günnemann, S. (2019). Combining neural networks with personalized pagerank for classification on graphs. In *International Conference on Learning Representations*.

[148] Kovalev, V. and Volmer, S. (1998). Color co-occurence descriptors for querying-by-example. In *International Conference on Multimedia Modeling (ICMM)*, page 32.

[149] Kreiss, S.; Bertoni, L.; and Alahi, A. (2019). Pifpaf: Composite fields for human pose estimation. *CoRR*, abs/1903.06593.

[150] Krizhevsky, A.; Nair, V.; and Hinton, G. E. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto. Technical Report TR-2009.

[151] Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

[152] Krull, A.; Buchholz, T.; and Jug, F. (2019). Noise2void - learning denoising from single noisy images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2124–2132.

[153] Kumar, D.; Siva, P.; Marchwica, P.; and Wong, A. (2019). Fairest of them all: Establishing a strong baseline for cross-domain person reid. *CoRR*, abs/1907.12016.

[154] Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. Wiley.

[155] Kurland, O. (2014). The cluster hypothesis in information retrieval. In de Rijke, M.; Kenter, T.; de Vries, A. P.; Zhai, C.; de Jong, F.; Radinsky, K.; and Hofmann, K., editors, *Advances in Information Retrieval*, pages 823–826, Cham. Springer International Publishing.

[156] Köstinger, M.; Hirzer, M.; Wohlhart, P.; Roth, P. M.; and Bischof, H. (2012). Large scale metric learning from equivalence constraints. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2288–2295.

[157] Laine, S. and Aila, T. (2017). Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*.

[158] Laine, S.; Karras, T.; Lehtinen, J.; and Aila, T. (2019). High-quality self-supervised deep image denoising. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.;

Fox, E.; and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 6970–6980. Curran Associates, Inc.

[159] Lao, G.; Liu, S.; Tan, C.; Wang, Y.; Li, G.; Xu, L.; Feng, L.; and Wang, F. (2021). Three degree binary graph and shortest edge clustering for re-ranking in multi-feature image retrieval. *Journal of Visual Communication and Image Representation*, 80:103282.

[160] Lary, D. J.; Alavi, A. H.; Gandomi, A. H.; and Walker, A. L. (2016). Machine learning in geosciences and remote sensing. *Geoscience Frontiers*, 7(1):3–10. Special Issue: Progress of Machine Learning in Geosciences.

[161] Latecki, L. J.; Lakamper, R.; and Eckhardt, U. (2000). Shape descriptors for non-rigid shapes with a single closed contour. In *CVPR*, pages 424–429.

[162] Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3.

[163] Lee, K.; Jang, I.-S.; Kim, K.-J.; and Kim, P.-K. (2022). Reet: Region-enhanced transformer for person re-identification. In *2022 18th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8.

[164] Lehtinen, J.; Munkberg, J.; Hasselgren, J.; Laine, S.; Karras, T.; Aittala, M.; and Aila, T. (2018). Noise2Noise: Learning image restoration without clean data. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2965–2974. PMLR.

[165] Leng, Q.; Hu, R.; Liang, C.; Wang, Y.; and Chen, J. (2014). Person re-identification with content and context re-ranking. *Multimedia Tools and Applications*, 74.

[166] Leticio, G.; Valem, L. P.; Lopes, L. T.; and Pedronette, D. C. G. (2023). pyudlf: A python framework for unsupervised distance learning tasks. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM ’23, page 9680–9684, New York, NY, USA. Association for Computing Machinery.

[167] Leticio, G. R.; Kawai, V. S.; Valem, L. P.; Pedronette, D. C. G.; and da S. Torres, R. (2024). Manifold information through neighbor embedding projection for image retrieval. *Pattern Recognition Letters*, 183:17–25.

[168] Li, D.; Li, D.; Zhang, Z.; Wang, L.; and Tan, T. (2019a). Unsupervised cross-domain person re-identification: A new framework. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1222–1226.

[169] Li, J.; Xiong, C.; and Hoi, S. C. H. (2021a). Comatch: Semi-supervised learning with contrastive graph regularization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9455–9464.

[170] Li, M.; Zhu, X.; and Gong, S. (2018a). Unsupervised person re-identification by deep learning tracklet association. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 737–753.

[171] Li, M.; Zhu, X.; and Gong, S. (2019b). Unsupervised tracklet person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

[172] Li, Q.; Han, Z.; and Wu, X.-m. (2018b). Deeper insights into graph convolutional networks for semi-supervised learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

[173] Li, S.; Gao, C.; Yu, H.; and Zhang, J. (2016). Person re-identification via person dpm based partition. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3856–3861.

[174] Li, W.; Wu, Y.; Mukunoki, M.; and Minoh, M. (2012). Common-near-neighbor analysis for person re-identification. In *2012 19th IEEE International Conference on Image Processing*, pages 1621–1624.

[175] Li, W.; Zhao, R.; and Wang, X. (2012a). Human reidentification with transferred metric learning. In *ACCV*.

[176] Li, W.; Zhao, R.; Xiao, T.; and Wang, X. (2014). Deepreid: Deep filter pairing neural network for person re-identification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 152–159.

[177] Li, W.; Zhu, X.; and Gong, S. (2018c). Harmonious attention network for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[178] Li, X.; Larson, M.; and Hanjalic, A. (2015). Pairwise geometric matching for large-scale object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2015)*, pages 5153–5161.

[179] Li, Y.; Geng, B.; Yang, L.; Xu, C.; and Bian, W. (2012b). Query difficulty estimation for image retrieval. *Neurocomputing*, 95:48–53.

[180] Li, Y.; Wu, Z.; Karanam, S.; and Radke, R. J. (2014). Real-world re-identification in an airport camera network. In *Proceedings of the International Conference on Distributed Smart Cameras*, ICDSC '14, New York, NY, USA. Association for Computing Machinery.

[181] Li, Y.-J.; Yang, F.-E.; Liu, Y.-C.; Yeh, Y.-Y.; Du, X.; and Frank Wang, Y.-C. (2018d). Adaptation and re-identification network: An unsupervised deep transfer learning approach to person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[182] Li, Z.; Yang, Y.; Liu, J.; Zhou, X.; and Lu, H. (2021b). Unsupervised feature selection using nonnegative spectral analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1):1026–1032.

[183] Liao, C.; Tsiligkaridis, T.; and Kulis, B. (2023). Supervised metric learning to rank for retrieval via contextual similarity optimization. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

[184] Liao, S.; Hu, Y.; Xiangyu Zhu; and Li, S. Z. (2015). Person re-identification by local maximal occurrence representation and metric learning. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2197–2206.

[185] Liao, S. and Shao, L. (2022a). Graph sampling based deep metric learning for generalizable person re-identification. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7349–7358.

[186] Liao, S. and Shao, L. (2022b). Graph sampling based deep metric learning for generalizable person re-identification. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7349–7358.

[187] Liao, S.; Zhao, G.; Kellokumpu, V.; Pietikäinen, M.; and Li, S. Z. (2010). Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1301–1306.

[188] Lin, Y.; Wu, Y.; Yan, C.; Xu, M.; and Yang, Y. (2020a). Unsupervised person re-identification via cross-camera similarity exploration. *IEEE Transactions on Image Processing*.

[189] Lin, Y.; Xie, L.; Wu, Y.; Yan, C.; and Tian, Q. (2020b). Unsupervised person re-identification via softened similarity learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3387–3396.

[190] Ling, H. and Jacobs, D. W. (2007). Shape classification using the inner-distance. *IEEE TPAMI*, 29(2):286–299.

[191] Ling, H.; Yang, X.; and Latecki, L. J. (2010). Balancing deformability and discriminability for shape matching. In *ECCV*, volume 3, pages 411–424.

[192] Lisanti, G.; Masi, I.; Bagdanov, A. D.; and Bimbo, A. D. (2015). Person re-identification by iterative re-weighted sparse ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1629–1642.

[193] Liu, C.; Yu, G.; Volkovs, M.; Chang, C.; Rai, H.; Ma, J.; and Gorti, S. K. (2019a). Guided similarity separation for image retrieval. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[194] Liu, G.-H. and Yang, J.-Y. (2013). Content-based image retrieval using color difference histogram. *Pattern Recognition*, 46(1):188 – 198.

[195] Liu, H.; Cheng, J.; Wang, S.; and Wang, W. (2019b). Attention: A big surprise for cross-domain person re-identification. *CoRR*, abs/1905.12830.

[196] Liu, H.; Feng, J.; Qi, M.; Jiang, J.; and Yan, S. (2016a). End-to-end comparative attention networks for person re-identification. *CoRR*, abs/1606.04404.

[197] Liu, J.; Zha, Z.-J.; Chen, D.; Hong, R.; and Wang, M. (2019c). Adaptive transfer network for cross-domain person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[198] Liu, S. and Deng, W. (2015). Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 730–734.

[199] Liu, W.; Wen, Y.; Yu, Z.; and Yang, M. (2016b). Large-margin softmax loss for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 2.

[200] Liu, X.; Song, M.; Tao, D.; Zhou, X.; Chen, C.; and Bu, J. (2014). Semi-supervised coupled dictionary learning for person re-identification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3550–3557.

[201] Liu, Y.; Ge, H.; Wang, Z.; Hou, Y.; and Zhao, M. (2024). Discriminative identity-feature exploring and differential aware learning for unsupervised person re-identification. *IEEE Transactions on Multimedia*, 26:623–636.

[202] Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030.

[203] Liu, Z.; Wang, S.; Zheng, L.; and Tian, Q. (2017). Robust imagegraph: Rank-level feature fusion for image search. *IEEE Transactions on Image Processing*, 26(7):3128–3141.

[204] Lopes, L. T.; Valem, L. P.; Pedronette, D. C. G.; Guilherme, I. R.; Papa., J. P.; Santana, M. C. S.; and Colombo, D. (2020). Manifold learning-based clustering approach applied to anomaly detection in surveillance videos. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, pages 404–412. INSTICC, SciTePress.

[205] Lowe, D. (1999). Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157.

[206] Loy, C. C.; Xiang, T.; and Gong, S. (2009). Multi-camera activity correlation analysis. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1988–1995.

[207] Lukežič, A.; Vojíř, T.; Čehovin Zajc, L.; Matas, J.; and Kristan, M. (2018). Discriminative correlation filter with channel and spatial reliability. *International Journal of Computer Vision*, 126.

[208] Luo, X.; Ju, W.; Gu, Y.; Mao, Z.; Liu, L.; Yuan, Y.; and Zhang, M. (2023). Self-supervised graph-level representation learning with adversarial contrastive learning. *ACM Trans. Knowl. Discov. Data*, 18(2).

[209] Lux, M. (2011). Content based image retrieval with lire. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 735–738, New York, NY, USA. ACM.

[210] Lv, Y.; Zhou, W.; Tian, Q.; Sun, S.; and Li, H. (2018). Retrieval oriented deep feature learning with complementary supervision mining. *IEEE Transactions on Image Processing*, 27(10):4945–4957.

[211] Ma, A. J. and Li, P. (2015). Query based adaptive re-ranking for person re-identification. In Cremers, D.; Reid, I.; Saito, H.; and Yang, M.-H., editors, *Computer Vision – ACCV 2014*, pages 397–412, Cham. Springer International Publishing.

[212] Ma, B.; Su, Y.; and Jurie, F. (2012). Local descriptors encoded by fisher vectors for person re-identification. In Fusiello, A.; Murino, V.; and Cucchiara, R., editors, *Computer Vision – ECCV 2012. Workshops and Demonstrations*, pages 413–422, Berlin, Heidelberg. Springer Berlin Heidelberg.

[213] Ma, B.; Su, Y.; and Jurie, F. (2014). Covariance descriptor based on bio-inspired features for person re-identification and face verification. *Image and Vision Computing*, 32(6):379 – 390.

[214] Maaten, L. V. D. and Hinton, G. E. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

[215] Manjunath, B. S.; Ohm, J. R.; Vasudevan, V. V.; and Yamada, A. (2001). Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715.

[216] Martinel, N.; Micheloni, C.; and Foresti, G. L. (2015). Kernelized saliency-based person re-identification through multiple metric learning. *IEEE Transactions on Image Processing*, 24(12):5645–5658.

[217] Matsukawa, T.; Okabe, T.; Suzuki, E.; and Sato, Y. (2016). Hierarchical gaussian descriptor for person re-identification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1363–1372.

[218] Meishvili, G.; Djelouah, A.; Hattori, S.; and Schroers, C. (2022). Contrastive learning for controllable blind video restoration. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press.

[219] Meng, J.; Wu, S.; and Zheng, W.-S. (2019). Weakly supervised person re-identification. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 760–769.

[220] Miyato, T.; ichi Maeda, S.; Koyama, M.; and Ishii, S. (2019). Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:1979–1993.

[221] Mo, S.; Sun, Z.; and Li, C. (2023). Multi-level contrastive learning for self-supervised vision transformers. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2777–2786.

[222] Mopuri, K. R. and Babu, R. V. (2015). Object level deep feature pooling for compact image representation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[223] Moran, N.; Schmidt, D.; Zhong, Y.; and Coady, P. (2020). Noisier2noise: Learning to denoise from unpaired noisy data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12061–12069.

[224] Mou, Y.; He, K.; Wang, P.; Wu, Y.; Wang, J.; Wu, W.; and Xu, W. (2022). Watch the neighbors: A unified k-nearest neighbor contrastive learning framework for ood intent discovery.

[225] Nguyen, V.; Ngo, T. D.; Nguyen, K. M. T. T.; Duong, D. A.; Nguyen, K.; and Le, D. (2013). Re-ranking for person re-identification. In *2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR)*, pages 304–308.

[226] Ni, H.; Li, Y.; Gao, L.; Shen, H.; and Song, J. (2023). Part-aware transformer for generalizable person re-identification. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11246–11255, Los Alamitos, CA, USA. IEEE Computer Society.

[227] Nie, L.; Jiao, F.; Wang, W.; Wang, Y.; and Tian, Q. (2021). Conversational image search. *IEEE Transactions on Image Processing*, 30:7732–7743.

[228] Nie, L.; Li, Y.; Feng, F.; Song, X.; Wang, M.; and Wang, Y. (2020). Large-scale question tagging via joint question-topic embedding learning. *ACM Trans. Inf. Syst.*, 38(2).

[229] Nilsback, M.-E. and Zisserman, A. (2006). A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454.

[230] Nistér, D. and Stewénius, H. (2006). Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, volume 2, pages 2161–2168.

[231] Ojala, T.; Pietikäinen, M.; and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987.

[232] Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175.

[233] Oliveira, A. and Rocha, A. (2020). Contextual features and sequence labeling techniques for relevance prediction in retrieval. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 305–310.

[234] Omohundro, S. M. (1989). Five balltree construction algorithms. Technical report, International Computer Science Institute.

[235] Ouyang, J.; Wu, H.; Wang, M.; Zhou, W.; and Li, H. (2021). Contextual similarity aggregation with self-attention for visual re-ranking. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3135–3148. Curran Associates, Inc.

[236] Papa, J. P.; Falcão, A.; and Suzuki, C. (2009). Supervised pattern classification based on optimum-path forest. *Int. J. Imaging Syst. Technol.*, 19(2):120–131.

[237] Park, G.; Baek, Y.; and Lee, H.-K. (2005). Re-ranking algorithm using post-retrieval clustering for content-based image retrieval. *Information Processing and Management*, 41(2):177–194.

[238] Paulin, M.; Mairal, J.; Douze, M.; Harchaoui, Z.; Perronnin, F.; and Schmid, C. (2017). Convolutional patch representations for image retrieval: An unsupervised approach. *Int. Journal of Computer Vision*, 121(1):149–168.

[239] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[240] Pedronette, D. C. G.; Almeida, J.; and Torres, R. d. S. (2014a). A scalable re-ranking method for content-based image retrieval. *Information Sciences*, 265:91–104.

[241] Pedronette, D. C. G.; Gonçalves, F. M. F.; and Guilherme, I. R. (2018). Unsupervised manifold learning through reciprocal kNN graph and Connected Components for image retrieval tasks. *Pattern Recognition*, 75:161 – 174.

[242] Pedronette, D. C. G. and Latecki, L. J. (2021). Rank-based self-training for graph convolutional networks. *Information Processing & Management*, 58(2):102443.

[243] Pedronette, D. C. G.; Penatti, O. A.; and Torres, R. d. S. (2014b). Unsupervised manifold learning using reciprocal knn graphs in image re-ranking and rank aggregation tasks. *Image and Vision Computing*, 32(2):120 – 130.

[244] Pedronette, D. C. G. and Torres, R. d. S. (2010). Shape retrieval using contour features and distance optmization. In *VISAPP*, volume 1, pages 197 – 202.

[245] Pedronette, D. C. G. and Torres, R. d. S. (2011). Exploiting contextual spaces for image re-ranking and rank aggregation. In *Proceedings of the 1st International Conference on Multimedia Retrieval, (ICMR)*, page 13. ACM.

[246] Pedronette, D. C. G. and Torres, R. d. S. (2012). Exploiting contextual information for image re-ranking and rank aggregation. *International Journal of Multimedia Information Retrieval*, 1(2):115–128.

[247] Pedronette, D. C. G. and Torres, R. d. S. (2013). Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition*, 46(8):2350–2360.

[248] Pedronette, D. C. G. and Torres, R. d. S. (2015). Unsupervised effectiveness estimation for image retrieval using reciprocal rank information. In *2015 28th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 321–328.

[249] Pedronette, D. C. G. and Torres, R. d. S. (2016). A correlation graph approach for unsupervised manifold learning in image retrieval tasks. *Neurocomputing*, 208:66–79. SI: BridgingSemantic.

[250] Pedronette, D. C. G.; Torres, R. d. S.; and Calumby, R. T. (2014c). Using contextual spaces for image re-ranking and rank aggregation. *Multimedia Tools and Applications*, 69(3):689–716.

[251] Pedronette, D. C. G.; Valem, L. P.; Almeida, J.; and Torres, R. d. S. (2019). Multimedia retrieval through unsupervised hypergraph-based manifold ranking. *IEEE Transactions on Image Processing*, 28(12):5824–5838.

[252] Pedronette, D. C. G.; Valem, L. P.; and Latecki, L. J. (2021a). Efficient rank-based diffusion process with assured convergence. *Journal of Imaging*, 7(3).

[253] Pedronette, D. C. G.; Valem, L. P.; and Torres, R. d. S. (2021b). A bfs-tree of ranking references for unsupervised manifold learning. *Pattern Recognition*, 111:107666.

[254] Penatti, O. A.; Valle, E.; and Torres, R. d. S. (2012). Comparative study of global color and texture descriptors for web image retrieval. *Journal of Visual Communication and Image Representation*, 23(2):359–380.

[255] Peng, L.; Qiang, B.; and Wu, J. (2022). A survey: Image classification models based on convolutional neural networks. In *2022 14th International Conference on Computer Research and Development (ICCRD)*, pages 291–298.

[256] Pereira-Ferrero, V.; Lewis, T.; Valem, L.; Ferrero, L.; Pedronette, D.; and Latecki, L. (2024a). Unsupervised affinity learning based on manifold analysis for image retrieval: A survey. *Computer Science Review*, 53:100657.

[257] Pereira-Ferrero, V.; Valem, L.; Leticio, G.; and Pedronette, D. (2023a). Feature fusion and augmentation based on manifold ranking for image classification. In *2023 IEEE Sixth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 75–82, Los Alamitos, CA, USA. IEEE Computer Society.

[258] Pereira-Ferrero, V. H.; Valem, L. P.; Leticio, G. R.; and Pedronette, D. C. G. (2024b). Feature fusion and augmentation based on manifold ranking for image classification. *International Journal of Semantic Computing (IJSC)*. Accepted, to appear.

[259] Pereira-Ferrero, V. H.; Valem, L. P.; and Pedronette, D. C. G. (2023b). Feature augmentation based on manifold ranking and lstm for image classification. *Expert Systems with Applications*, 213:118995.

[260] Piras, L. and Giacinto, G. (2017). Information fusion in content based image retrieval: A comprehensive overview. *Information Fusion*, 37:50–60.

[261] Pisani, F.; Pascotti Valem, L.; Guimarães Pedronette, D. C.; da S. Torres, R.; Borin, E.; and Breternitz Jr., M. (2020). A unified model for accelerating unsupervised iterative

re-ranking algorithms. *Concurrency and Computation: Practice and Experience*, 32(14):e5702.

[262] Poesina, E.; Ionescu, R. T.; and Mothe, J. (2023). iQPP: A benchmark for image query performance prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 2953–2963, New York, NY, USA. Association for Computing Machinery.

[263] Presotto, J. G. C.; Dos Santos, S. F.; Valem, L. P.; Faria, F. A.; Papa, J. P.; Almeida, J.; and Pedronette, D. C. G. (2022). Weakly supervised learning based on hypergraph manifold ranking. *Journal of Visual Communication and Image Representation*, 89:103666.

[264] Presotto, J. G. C.; Valem, L. P.; De Sá, N. G.; Pedronette, D. C. G.; and Papa, J. P. (2021). Weakly supervised learning through rank-based contextual measures. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5752–5759.

[265] Presotto, J. G. C.; Valem, L. P.; De Sá, N. G.; Pedronette, D. C. G.; and Papa, J. P. (2024). Weakly supervised classification through manifold learning and rank-based contextual measures. *Neurocomputing*, 589(C).

[266] Presotto, J. G. C.; Valem, L. P.; and Pedronette, D. C. G. (2019). Unsupervised effectiveness estimation through intersection of ranking references. In *Computer Analysis of Images and Patterns - CAIP 2019*, volume 11679, pages 231–244.

[267] Qin, D.; Gammeter, S.; Bossard, L.; Quack, T.; and van Gool, L. (2011). Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR 2011*, pages 777–784.

[268] Qin, D.; Wengert, C.; and Gool, L. V. (2013). Query adaptive similarity for large scale object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2013)*, pages 1610–1617.

[269] Quan, Y.; Chen, M.; Pang, T.; and Ji, H. (2020). Self2self with dropout: Learning self-supervised denoising from single image. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1887–1895.

[270] Raghu, M. and Schmidt, E. (2020). A survey of deep learning for scientific discovery. *arXiv preprint arXiv:2003.11755*.

[271] Razavian, A. S.; Sullivan, J.; Carlsson, S.; and Maki, A. (2016). Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258.

[272] Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788.

[273] Ren, C.; Liang, B.; and Lei, Z. (2019). Domain adaptive person re-identification via camera style generation and label propagation. *CoRR*, abs/1905.05382.

[274] Rodrigues, J. and Carbonera, J. (2024). Graph convolutional networks for image classification: Comparing approaches for building graphs from images. In *Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 1: ICEIS*, pages 437–446. INSTICC, SciTePress.

[275] Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520.

[276] Schmid, C. (2001). Constructing models for content-based image retrieval. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 2, pages II–II.

[277] Schölkopf, B.; Platt, J.; and Hofmann, T. (2007). Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems (NIPS'07)*, pages 1601–1608.

[278] Shao, P. and Tao, J. (2024). Multi-level graph contrastive learning. *Neurocomputing*, 570:127101.

[279] Shao, S.; Chen, K.; Karpur, A.; Cui, Q.; Araujo, A.; and Cao, B. (2023). Global features are all you need for image retrieval and reranking. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11002–11012.

[280] Sharma, A. and Kumar, D. (2022). Classification with 2-d convolutional neural networks for breast cancer diagnosis. *Scientific Reports*, 12(1):21857.

[281] Sharma, A.; Vans, E.; Shigemizu, D.; Boroevich, K. A.; and Tsunoda, T. (2019). Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Scientific Reports*, 9(1):11399.

[282] Shen, X.; Xiao, Y.; Hu, S. X.; Sbai, O.; and Aubry, M. (2021). Re-ranking for image retrieval and transductive few-shot classification. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25932–25943.

[283] Shimomura, L.; Oyamada, R.; Vieira, M.; and Kaster, D. (2021). A survey on graph-based methods for similarity searches in metric spaces. *Information Systems*, 95:101507.

[284] Shriram K. Vasudevan, P.L.K. Priyadarsini, S. V. (2012). *Content Based Image Retrieval (CBIR): A deeper insight*. LAP LAMBERT Academic Publishing.

[285] Shtok, A.; Kurland, O.; and Carmel, D. (2016). Query performance prediction using reference lists. *ACM Trans. Inf. Syst.*, 34(4).

[286] Shu, Y.; Gu, X.; Yang, G.-Z.; and Lo, B. P. L. (2022). Revisiting self-supervised contrastive learning for facial expression recognition. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press.

[287] Siam, M. K. H.; Bhattacharjee, M.; Mahmud, S.; Sarkar, M. S.; and Rana, M. M. (2024). The impact of machine learning on society: An analysis of current trends and future implications. *arXiv preprint arXiv:2404.10204*.

[288] Singh, N. K.; Khare, M.; and Jethva, H. B. (2022). A comprehensive survey on person re-identification approaches: various aspects. *Multimedia Tools and Applications*, 81(11):15747–15791.

[289] Sohn, K.; Berthelot, D.; Li, C.-L.; Zhang, Z.; Carlini, N.; Cubuk, E. D.; Kurakin, A.; Zhang, H.; and Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

[290] Soltanayev, S. and Chun, S. Y. (2018). Training deep learning based denoisers without ground truth data. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 3257–3267. Curran Associates, Inc.

[291] Somers, V.; Vleeschouwer, C. D.; and Alahi, A. (2023). Body part-based representation learning for occluded person re-identification. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1613–1623.

[292] Song, K.; Han, J.; Cheng, G.; Lu, J.; and Nie, F. (2022). Adaptive neighborhood metric learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4591–4604.

[293] Song, Z.; Yang, X.; Xu, Z.; and King, I. (2023). Graph-based semi-supervised learning: A comprehensive review. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8174–8194.

[294] Srivastava, D.; Singh, S. S.; Rajitha, B.; Verma, M.; Kaur, M.; and Lee, H.-N. (2023). Content-based image retrieval: A survey on local and global features selection, extraction, representation, and evaluation parameters. *IEEE Access*, 11:95410–95431.

[295] Stehling, R. O.; Nascimento, M. A.; and Falcão, A. X. (2002). A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM*, pages 102–109.

[296] Sukhbaatar, S.; Bruna, J.; Paluri, M.; Bourdev, L.; and Fergus, R. (2014). Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*.

[297] Sun, H.; Li, M.; and Li, C.-G. (2022). Hybrid contrastive learning with cluster ensemble for unsupervised person re-identification. In Wallraven, C.; Liu, Q.; and Nagahara, H., editors, *Pattern Recognition*, pages 532–546, Cham. Springer International Publishing.

[298] Sun, L.; Ji, S.; and Ye, J. (2008). Hypergraph spectral learning for multi-label classification. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’08, pages 668–676.

[299] Sun, S.; Li, Y.; Zhou, W.; Tian, Q.; and Li, H. (2017). Local residual similarity for image re-ranking. *Information Sciences*, 417(Sup. C):143 – 153.

[300] Sun, S.; Zhou, W.; Tian, Q.; Yang, M.; and Li, H. (2018a). Assessing image retrieval quality at the first glance. *IEEE Transactions on Image Processing*, 27(12):6124–6134.

[301] Sun, Y.; Zheng, L.; Yang, Y.; Tian, Q.; and Wang, S. (2018b). Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In Ferrari, V.; Hebert, M.; Sminchisescu, C.; and Weiss, Y., editors, *Computer Vision – ECCV 2018*, pages 501–518, Cham. Springer International Publishing.

[302] Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12.

[303] Sánchez, J.; Perronnin, F.; Mensink, T.; and Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105.

[304] Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.

[305] Tan, F.; Yuan, J.; and Ordonez, V. (2021). Instance-level image retrieval using reranking transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12085–12095.

[306] Tang, H.; Zhao, Y.; and Lu, H. (2019). Unsupervised person re-identification with iterative self-supervised domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.*

[307] Tang, W. (2023). Review of image classification algorithms based on graph convolutional networks. *EAI Endorsed Transactions on AI and Robotics*, 2(1).

[308] Tao, B. and Dickinson, B. W. (2000). Texture recognition and image retrieval using gradient indexing. *JVCIR*, 11(3):327–342.

[309] Tarvainen, A. and Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1195–1204, Red Hook, NY, USA. Curran Associates Inc.

[310] Tay, C.; Roy, S.; and Yap, K. (2019). Aanet: Attribute attention network for person re-identifications. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7127–7136.

[311] Thammasorn, P.; Hippe, D. S.; Chaovalitwongse, W. A.; Spraker, M.; Wootton, L.; Nyflot, M.; Combs, S.; Peeken, J.; and Ford, E. (2019). Neighborhood watch: Representation learning with local-margin triplet loss and sampling strategy for k-nearest-neighbor image classification. *CoRR*, abs/1911.07940.

[312] Thomee, B. and Lew, M. S. (2012). Interactive search in image retrieval: a survey. *International Journal of Multimedia Information Retrieval*, 1(2):71–86.

[313] Tian, X.; Jia, Q.; and Mei, T. (2015). Query difficulty estimation for image search with query reconstruction error. *IEEE Transactions on Multimedia*, 17(1):79–91.

[314] Tian, X.; Lu, Y.; and Yang, L. (2012). Query difficulty prediction for web image search. *IEEE Transactions on Multimedia*, 14(4):951–962.

[315] Tolias, G.; Avrithis, Y.; and Jégou, H. (2013). To aggregate or not to aggregate: Selective match kernels for image search. In *IEEE International Conference on Computer Vision (ICCV'2013)*, pages 1401–1408.

[316] Torres, R. d. S. and Falcão, A. X. (2006). Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, 13:161–185.

[317] Torres, R. d. S. and Falcão, A. X. (2007). Contour Salience Descriptors for Effective Image Retrieval and Analysis. *Image and Vision Computing*, 25(1):3–13.

[318] Torres, R. d. S.; Falcão, A. X.; Gonçalves, M. A.; Papa, J. P.; Zhang, B.; Fan, W.; and Fox, E. A. (2009). A genetic programming framework for content-based image

retrieval. *Pattern Recognition*, 42(2):283–292. Learning Semantics from Multimedia Content.

[319] Toshev, A. and Szegedy, C. (2013). Deeppose: Human pose estimation via deep neural networks. *CoRR*, abs/1312.4659.

[320] Tripathi, S. and King, C. R. (2024). Contrastive learning: Big data foundations and applications. In *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)*, CODS-COMAD '24, page 493–497, New York, NY, USA. Association for Computing Machinery.

[321] Uelwer, T.; Robine, J.; Wagner, S. S.; Höftmann, M.; Upschulte, E.; Konietzny, S.; Behrendt, M.; and Harmeling, S. (2023). A survey on self-supervised representation learning. *arXiv preprint arXiv:2308.11455*.

[322] Urbanowicz, R. J.; Meeker, M.; La Cava, W.; Olson, R. S.; and Moore, J. H. (2018). Relief-based feature selection: Introduction and review. *Journal of Biomedical Informatics*, 85:189–203.

[323] Valem, L. P.; De Oliveira, C. R.; Pedronette, D. C. G.; and Almeida, J. (2018a). Unsupervised similarity learning through rank correlation and knn sets. *ACM Trans. Multim. Comput. Commun. Appl.*, 14(4):80:1–80:23.

[324] Valem, L. P.; Kawai, V. A. S.; Pereira-Ferrero, V. H.; and Pedronette, D. C. G. (2022). A novel rank correlation measure for manifold learning on image retrieval and person re-id. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 1371–1375.

[325] Valem, L. P.; Pedronette, D.; and Allili, M. S. (2024). Contrastive loss based on contextual similarity for image classification. In *19th International Symposium on Visual Computing (ISVC)*. Submitted.

[326] Valem, L. P. and Pedronette, D. C. G. (2017). An unsupervised distance learning framework for multimedia retrieval. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, ICMR '17, pages 107–111, New York, NY, USA. ACM.

[327] Valem, L. P. and Pedronette, D. C. G. (2019). An unsupervised genetic algorithm framework for rank selection and fusion on image retrieval. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, ICMR '19, page 58–62, New York, NY, USA. Association for Computing Machinery.

[328] Valem, L. P. and Pedronette, D. C. G. (2020a). Graph-based selective rank fusion for unsupervised image retrieval. *Pattern Recognition Letters*, 135:82–89.

[329] Valem, L. P. and Pedronette, D. C. G. (2020b). Unsupervised selective rank fusion for image retrieval tasks. *Neurocomputing*, 377:182 – 199.

[330] Valem, L. P. and Pedronette, D. C. G. (2021). A denoising convolutional neural network for self-supervised rank effectiveness estimation on image retrieval. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, ICMR '21, page 294–302, New York, NY, USA. Association for Computing Machinery.

[331] Valem, L. P. and Pedronette, D. C. G. (2022). Person re-id through unsupervised hypergraph rank selection and fusion. *Image Vision Comput.*, 123(C).

[332] Valem, L. P.; Pedronette, D. C. G.; and Almeida, J. (2018b). Unsupervised similarity learning through cartesian product of ranking references. *Pattern Recognition Letters*, 114:41 – 52.

[333] Valem, L. P.; Pedronette, D. C. G.; and Latecki, L. J. (2023a). Graph convolutional networks based on manifold learning for semi-supervised image classification. *Computer Vision and Image Understanding*, 227:103618.

[334] Valem, L. P.; Pedronette, D. C. G.; and Latecki, L. J. (2023b). Rank flow embedding for unsupervised and semi-supervised manifold learning. *IEEE Transactions on Image Processing*, 32:2811–2826.

[335] Valem, L. P.; Pedronette, D. C. G.; Torres, R. d. S.; Borin, E.; and Almeida, J. (2015). Effective, efficient, and scalable unsupervised distance learning in image retrieval tasks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ICMR '15, pages 51–58, New York, NY, USA. ACM.

[336] Valem, L. P.; Pereira-Ferrero, V.; and Pedronette, D. (2023c). Self-supervised regression for query performance prediction on image retrieval. In *2023 IEEE Sixth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pages 95–98, Los Alamitos, CA, USA. IEEE Computer Society.

[337] Van de Sande, K. E. A.; Gevers, T.; and Snoek, C. G. M. (2010). Evaluating color descriptors for object and scene recognition. *PAMI*, 32(9):1582–1596.

[338] Vanyan, A. and Khachatrian, H. (2021). Deep semi-supervised image classification algorithms: a survey. *JUCS - Journal of Universal Computer Science*, 27(12):1390–1407.

[339] Varior, R. R.; Haloi, M.; and Wang, G. (2016a). Gated siamese convolutional neural network architecture for human re-identification. *ArXiv*, abs/1607.08378.

[340] Varior, R. R.; Shuai, B.; Lu, J.; Xu, D.; and Wang, G. (2016b). A siamese long short-term memory architecture for human re-identification. In *ECCV*.

[341] Varoquaux, G. and Cheplygina, V. (2022). Machine learning for medical imaging: methodological failures and recommendations for the future. *npj Digital Medicine*, 5(1):48.

[342] Vassou, S. A.; Anagnostopoulos, N.; Amanatiadis, A.; Christodoulou, K.; and Chatzichristofis, S. A. (2017). Como: A compact composite moment-based descriptor for image retrieval. In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, CBMI '17, pages 30:1–30:5, New York, NY, USA. ACM.

[343] Vasudevan, V.; Bassenne, M.; Islam, M. T.; and Xing, L. (2023). Image classification using graph neural network and multiscale wavelet superpixels. *Pattern Recogn. Lett.*, 166(C):89–96.

[344] Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. (2018). Graph attention networks. In *International Conference on Learning Representations*.

[345] Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; and Others (2016). Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638.

[346] Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. (2011). The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

[347] Wang, B. and Tu, Z. (2012). Affinity learning via self-diffusion for image segmentation and clustering. In *2012 IEEE CVPR*, pages 2312–2319.

[348] Wang, D. and Zhang, S. (2020). Unsupervised person re-identification via multi-label classification. In *CVPR*, pages 10978–10987.

[349] Wang, G.; Wang, G.; Zhang, X.; Lai, J.; Yu, Z.; and Lin, L. (2020). Weakly supervised person re-id: Differentiable graphical learning and a new benchmark.

[350] Wang, H.; Zhu, X.; Xiang, T.; and Gong, S. (2016). Towards unsupervised open-set person re-identification. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 769–773.

[351] Wang, J.; Li, Y.; Bai, X.; Zhang, Y.; Wang, C.; and Tang, N. (2011a). Learning context-sensitive similarity by shortest path propagation. *Pattern Recognition*, 44(10-11):2367–2374.

[352] Wang, J. and Zhu, Z. (2010). Image retrieval system based on multi-feature fusion and relevance feedback. In *2010 International Conference on Machine Learning and Cybernetics*, volume 4, pages 2053–2058.

[353] Wang, M.; Lai, B.; Huang, J.; Gong, X.; and Hua, X.-S. (2021). Camera-aware proxies for unsupervised person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).*

[354] Wang, M.; Li, J.; Lai, B.; Gong, X.; and Hua, X.-S. (2022). Offline-online associated camera-aware proxies for unsupervised person re-identification. *IEEE Transactions on Image Processing*, 31:6548–6561.

[355] Wang, M. and Song, T. (2013). Remote sensing image retrieval by scene semantic matching. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5):2874–2886.

[356] Wang, X.; Yang, M.; Cour, T.; Zhu, S.; Yu, K.; and Han, T. (2011b). Contextual weighting for vocabulary tree based image retrieval. In *ICCV'2011*, pages 209–216.

[357] Webber, C.; Ioana, C.; and Marwan, N. (2016). *Recurrence Plots and Their Quantifications: Expanding Horizons: Proceedings of the 6th International Symposium on Recurrence Plots, Grenoble, France, 17-19 June 2015*. Springer.

[358] Webber, W.; Moffat, A.; and Zobel, J. (2010). A similarity measure for indefinite rankings. *ACM Transactions on Information Systems*, 28(4):20:1–20:38.

[359] Wei, L.; Zhang, S.; Gao, W.; and Tian, Q. (2018). Person transfer gan to bridge domain gap for person re-identification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 79–88.

[360] Weinberger, K. Q.; Blitzer, J.; and Saul, L. (2006). Distance metric learning for large margin nearest neighbor classification. In Weiss, Y.; Schölkopf, B.; and Platt, J., editors, *Advances in Neural Information Processing Systems*, volume 18, pages 1473–1480. MIT Press.

[361] Wojke, N.; Bewley, A.; and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3645–3649. IEEE.

[362] Wu, C.; Wang, C.; Xu, J.; Liu, Z.; Zheng, K.; Wang, X.; Song, Y.; and Gai, K. (2023a). Graph contrastive learning with generative adversarial network. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 2721–2730, New York, NY, USA. Association for Computing Machinery.

[363] Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. (2019). Simplifying graph convolutional networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6861–6871. PMLR.

[364] Wu, H.; Wang, M.; Zhou, W.; Lu, Z.; and Li, H. (2023b). Asymmetric feature fusion for image retrieval. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11082–11092.

[365] Wu, L.; Shen, C.; and van den Hengel, A. (2016). Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification. *CoRR*, abs/1606.01595.

[366] Wu, Y.; Bourahla, O. E. F.; Li, X.; Wu, F.; Tian, Q.; and Zhou, X. (2020). Adaptive graph representation learning for video person re-identification. *IEEE Transactions on Image Processing*, 29:8821–8830.

[367] Wu, Y.; Huang, T.; Yao, H.; Zhang, C.; Shao, Y.; Han, C.; Gao, C.; and Sang, N. (2022). Multi-centroid representation network for domain adaptive person re-id. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3):2750–2758.

[368] Wu, Y.; Wu, X.; Li, X.; and Tian, J. (2021). Mgh: Metadata guided hypergraph modeling for unsupervised person re-identification. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 1571–1580, New York, NY, USA. Association for Computing Machinery.

[369] Xia, F.; Sun, K.; Yu, S.; Aziz, A.; Wan, L.; Pan, S.; and Liu, H. (2021). Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, 2(02):109–127.

[370] Xian, Y. and Hu, H. (2018). Enhanced multi-dataset transfer learning method for unsupervised person re-identification using co-training strategy. *IET Computer Vision*, 12(8):1219–1227.

[371] Xie, L.; Hong, R.; Zhang, B.; and Tian, Q. (2015). Image classification and retrieval are one. In *ACM ICMR'2015*, pages 3–10.

[372] Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. (2017). Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995.

[373] Xin, X.; Wang, J.; Xie, R.; Zhou, S.; Huang, W.; and Zheng, N. (2019). Semi-supervised person re-identification using multi-view clustering. *Pattern Recognition*, 88:285–297.

[374] Xing, X.; Zhang, Y.; and Han, M. (2010). Query difficulty prediction for contextual image retrieval. In *European Conference on IR Research, ECIR*, volume 5993, pages 581–585.

[375] Xiong, F.; Gou, M.; Camps, O.; and Sznaier, M. (2014). Person re-identification using kernel-based metric learning methods. In Fleet, D.; Pajdla, T.; Schiele, B.; and

Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 1–16, Cham. Springer International Publishing.

[376] Xuan, S. and Zhang, S. (2021). Intra-inter camera similarity for unsupervised person re-identification. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11921–11930.

[377] Yang, B.; Shan, Y.; Peng, R.; Li, J.; Chen, S.; and Li, L. (2022). A feature extraction method for person re-identification based on a two-branch cnn. *Multimedia Tools and Applications*, 81(27):39169–39184.

[378] Yang, F.; Hinami, R.; Matsui, Y.; Ly, S.; and Satoh, S. (2019a). Efficient image retrieval via decoupling diffusion into online and offline processing. In *Conference on Artificial Intelligence, AAAI 2019*, pages 9087–9094. AAAI Press.

[379] Yang, J.; Zheng, W.-S.; Yang, Q.; Chen, Y.-C.; and Tian, Q. (2020). Spatial-temporal graph convolutional network for video-based person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3286–3296.

[380] Yang, Q.; Yu, H.-X.; Wu, A.; and Zheng, W.-S. (2019b). Patch-based discriminative feature learning for unsupervised person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[381] Yang, X.; Bai, X.; Latecki, L. J.; and Tu, Z. (2008). Improving shape retrieval by learning graph transduction. In *ECCV*, volume 4, pages 788–801.

[382] Yang, X.; Koknar-Tezel, S.; and Latecki, L. J. (2009a). Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *CVPR*, pages 357–364.

[383] Yang, X.; Koknar-Tezel, S.; and Latecki, L. J. (2009b). Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 357–364.

[384] Yang, X. and Latecki, L. J. (2011). Affinity learning on a tensor product graph with applications to shape and image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2011)*, pages 2369–2376.

[385] Yang, X.; Prasad, L.; and Latecki, L. J. (2013). Affinity learning with diffusion on tensor product graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):28–38.

[386] Yang, X.; Zhou, P.; and Wang, M. (2019). Person reidentification via structural deep metric learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(10):2987–2998.

[387] Yang, Y.; Yang, J.; Yan, J.; Liao, S.; Yi, D.; and Li, S. Z. (2014). Salient color names for person re-identification. In Fleet, D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 536–551, Cham. Springer International Publishing.

[388] Ye, M.; Chen, J.; Leng, Q.; Liang, C.; Wang, Z.; and Sun, K. (2015). Coupled-view based ranking optimization for person re-identification. In He, X.; Luo, S.; Tao, D.; Xu, C.; Yang, J.; and Hasan, M. A., editors, *MultiMedia Modeling*, pages 105–117, Cham. Springer International Publishing.

[389] Ye, M.; Liang, C.; Yu, Y.; Wang, Z.; Leng, Q.; Xiao, C.; Chen, J.; and Hu, R. (2016). Person reidentification via ranking aggregation of similarity pulling and dissimilarity pushing. *IEEE Transactions on Multimedia*, 18(12):2553–2566.

[390] Ye, M.; Shen, J.; Lin, G.; Xiang, T.; Shao, L.; and Hoi, S. C. H. (2020). Deep learning for person re-identification: A survey and outlook.

[391] Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z. (2014). Deep metric learning for person re-identification. In *2014 22nd International Conference on Pattern Recognition*, pages 34–39.

[392] You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. (2020). Graph contrastive learning with augmentations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

[393] Younesi, A.; Ansari, M.; Fazli, M.; Ejlali, A.; Shafique, M.; and Henkel, J. (2024). A comprehensive survey of convolutions in deep learning: Applications, challenges, and future trends. *arXiv preprint arXiv:2402.15490.*

[394] Young, H. (1974). An axiomatization of borda's rule. *Journal of Economic Theory*, 9(1):43–52.

[395] Yu, H.; Wu, A.; and Zheng, W. (2020). Unsupervised person re-identification by deep asymmetric metric embedding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):956–973.

[396] Yu, H.-X.; Wu, A.; and Zheng, W.-S. (2017a). Cross-view asymmetric metric learning for unsupervised person re-identification. In *The IEEE International Conference on Computer Vision (ICCV).*

[397] Yu, H.-X.; Zheng, W.-S.; Wu, A.; Guo, X.; Gong, S.; and Lai, J.-H. (2019). Unsupervised person re-identification by soft multilabel learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[398] Yu, W.; Yang, K.; Yao, H.; Sun, X.; and Xu, P. (2017b). Exploiting the complementary strengths of multi-layer cnn features for image retrieval. *Neurocomputing*, 237:235–241.

[399] Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Tay, F. E.; Feng, J.; and Yan, S. (2021). Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*.

[400] Yuan, Y.; Chen, W.; Yang, Y.; and Wang, Z. (2019). In defense of the triplet loss again: Learning robust person re-identification with fast approximated triplet loss and label distillation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1454–1463.

[401] Zagoris, K.; Chatzichristofis, S.; Papamarkos, N.; and Boutalis, Y. (2010). Automatic image annotation and retrieval using the joint composite descriptor. In *14th Panhellenic Conference on Informatics (PCI)*, pages 143–147.

[402] Zeng, K.; Ning, M.; Wang, Y.; and Guo, Y. (2020). Hierarchical clustering with hard-batch triplet loss for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[403] Zeng, Q. and Yu, H. (2019). Group affinity guided deep hypergraph model for person re-identification. *Electronics Letters*, 55(4):186–188.

[404] Zhai, Y.; Lu, S.; Ye, Q.; Shan, X.; Chen, J.; Ji, R.; and Tian, Y. (2020). Ad-cluster: Augmented discriminative clustering for domain adaptive person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[405] Zhang, G.; Pan, J.; Zhang, Z.; Zhang, H.; Xing, C.; Sun, B.; and Li, M. (2021a). Hybrid graph convolutional network for semi-supervised retinal image classification. *IEEE Access*, 9:35778–35789.

[406] Zhang, H.; Chen, X.; Jing, H.; Zheng, Y.; Wu, Y.; and Jin, C. (2023). Etr: An efficient transformer for re-ranking in visual place recognition. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5654–5663.

[407] Zhang, H.; Li, Y.; Zhuang, Z.; Xie, L.; and Tian, Q. (2021b). 3d-gat: 3d-guided adversarial transform network for person re-identification in unseen domains. *Pattern Recognition*, 112:107799.

[408] Zhang, K.; Li, Y.; Zuo, W.; Zhang, L.; Van Gool, L.; and Timofte, R. (2020). Plug-and-play image restoration with deep denoiser prior. *arXiv preprint*.

[409] Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. (2017). Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155.

[410] Zhang, L.; Song, H.; Aletras, N.; and Lu, H. (2018a). Graph node-feature convolution for representation learning. *arXiv preprint arXiv:1812.00086*.

[411] Zhang, L.; Xiang, T.; and Gong, S. (2016). Learning a discriminative null space for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[412] Zhang, S.; Tong, H.; Xu, J.; and Maciejewski, R. (2019). Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):11.

[413] Zhang, S.; Yang, M.; Cour, T.; Yu, K.; and Metaxas, D. (2015). Query specific rank fusion for image retrieval. *IEEE TPAMI*, 37(4):803–815.

[414] Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 116, New York, NY, USA. Association for Computing Machinery.

[415] Zhang, X.; Ge, Y.; Qiao, Y.; and Li, H. (2021c). Refining pseudo labels with clustering consensus over generations for unsupervised object re-identification. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3435–3444.

[416] Zhang, Y.; Qian, Q.; Wang, H.; Liu, C.; Chen, W.; and Wang, F. (2024). Graph convolution based efficient re-ranking for visual retrieval. *IEEE Transactions on Multimedia*, 26:1089–1101.

[417] Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; and Fu, Y. (2018b). Residual dense network for image super-resolution. In *CVPR*.

[418] Zhang, Z. and Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pages 8778–8788.

[419] Zhao, R.; Ouyang, W.; and Wang, X. (2013). Unsupervised salience learning for person re-identification. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3586–3593.

[420] Zhao, Y.; Wang, L.; Zhou, L.; Shi, Y.; and Gao, Y. (2018). Modelling diffusion process by deep neural networks for image retrieval. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, page 161. BMVA Press.

[421] Zhao, Z. and Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 1151–1157, New York, NY, USA. ACM.

[422] Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; and Tian, Q. (2015a). Scalable person re-identification: A benchmark. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1116–1124.

[423] Zheng, L.; Wang, S.; Liu, Z.; and Tian, Q. (2014a). Packing and padding: Coupled multi-index for accurate image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2014)*, pages 1947–1954.

[424] Zheng, L.; Wang, S.; Tian, L.; Fei He; Liu, Z.; and Tian, Q. (2015b). Query-adaptive late fusion for image search and person re-identification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1741–1750.

[425] Zheng, L.; Wang, S.; and Tian, Q. (2014b). Coupled binary embedding for large-scale image retrieval. *IEEE Transactions on Image Processing (TIP)*, 23(8):3368–3380.

[426] Zheng, L.; Yang, Y.; and Hauptmann, A. G. (2016). Person re-identification: Past, present and future. *CoRR*, abs/1610.02984.

[427] Zheng, L.; Yang, Y.; and Tian, Q. (2018). Sift meets cnn: A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1224–1244.

[428] Zheng, Z.; Zheng, L.; and Yang, Y. (2017). Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE International Conference on Computer Vision*, page 3754–3762.

[429] Zhong, Z.; Zheng, L.; Cao, D.; and Li, S. (2017). Re-ranking person re-identification with k-reciprocal encoding. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3652–3661.

[430] Zhong, Z.; Zheng, L.; Li, S.; and Yang, Y. (2018). Generalizing a person retrieval model hetero- and homogeneously. In *The European Conference on Computer Vision (ECCV)*.

[431] Zhong, Z.; Zheng, L.; Luo, Z.; Li, S.; and Yang, Y. (2019). Invariance matters: Exemplar memory for domain adaptive person re-identification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[432] Zhong, Z.; Zheng, L.; Luo, Z.; Li, S.; and Yang, Y. (2020). Learning to adapt invariance in memory for person re-identification. *TPAMI*, pages 1–1.

[433] Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. (2004). Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press.

[434] Zhou, D.; Weston, J.; Gretton, A.; Bousquet, O.; and Schölkopf, B. (2003). Ranking on data manifolds. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, page 169–176, Cambridge, MA, USA. MIT Press.

[435] Zhou, K. and Xiang, T. (2019). Torchreid: A library for deep learning person re-identification in pytorch. *arXiv preprint arXiv:1910.10093*.

[436] Zhou, K.; Yang, Y.; Cavallaro, A.; and Xiang, T. (2019a). Learning generalisable omni-scale representations for person re-identification. *arXiv preprint arXiv:1910.06827*.

[437] Zhou, K.; Yang, Y.; Cavallaro, A.; and Xiang, T. (2019b). Omni-scale feature learning for person re-identification. In *The IEEE International Conference on Computer Vision (ICCV)*.

[438] Zhou, W.; Li, H.; and Tian, Q. (2017). Recent advance in content-based image retrieval: A literature survey. *CoRR*, abs/1706.06064.

[439] Zhou, Y. and Croft, W. B. (2006). Ranking robustness: A novel framework to predict query performance. In *ACM Int. Conference on Information and Knowledge Management (CIKM'06)*, pages 567–574.

[440] Zhou, Y. and Croft, W. B. (2007). Query performance prediction in web search environments. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*, pages 543–550.

[441] Zhou, Y.; Liu, P.; and Qiu, X. (2022). KNN-contrastive learning for out-of-domain intent classification. In Muresan, S.; Nakov, P.; and Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5129–5141, Dublin, Ireland. Association for Computational Linguistics.

[442] Zhu, S.; Yang, L.; Chen, C.; Shah, M.; Shen, X.; and Wang, H. (2023). R2former: Unified retrieval and reranking transformer for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19370–19380.

[443] Zhu, X.; Zhu, X.; Li, M.; Morerio, P.; Murino, V.; and Gong, S. (2021). Intra-camera supervised person re-identification. *International Journal of Computer Vision*, 129(5):1580–1595.

[444] Zhu, X.; Zhu, X.; Li, M.; Murino, V.; and Gong, S. (2019). Intra-camera supervised person re-identification: A new benchmark. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 1079–1087.

[445] Zoph, B.; Vasudevan, V.; Shlens, J.; and Le, Q. V. (2017). Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012.

[446] Zou, G.; Fu, G.; Peng, X.; Liu, Y.; Gao, M.; and Liu, Z. (2021). Person re-identification based on metric learning: a survey. *Multimedia Tools and Applications*, 80(17):26855–26888.