Target Re-identification
Final Project Report

Chady Raach
chady.raach@student-cs.fr

Marah Gamdou
marah.gamdou@student-cs.fr

Mehdi Zemni
mehdi.zemni@student-cs.fr

Abstract

In this project, we explore several approaches to optimise target re-identification (re-ID) as a re-ranking problem. Our work consisted in trying different methods to re-rank the re-ID results. Each image is represented by a node in a graph to which we have applied supervised and unsupervised link prediction methods. The purpose of unsupervised approaches is to define a new distance where a query is close to images from the gallery that have the same ID. Supervised link prediction consists in adding new node features from the graph topology and predicting the strength of the link between a query and a gallery image, both represented by a node. Supervised and unsupervised solutions allow to re-rank the gallery images such that true matches are the closest to each query.

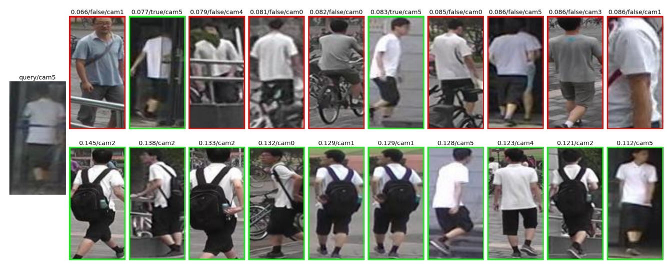

Figure 1: Person re-identification with different acquisition settings: In the left, the query is shown. In the first row, we can see the closest images ordered and returned by the re-ID approach (when the image is in a red box, it is not a true match). In the second row, all the true matches of the query that exist in the dataset are listed (the goal is to list all these true matches first).

1. Motivation

Target re-identification (Re-ID) has been widely studied as a specific target retrieval problem across non-overlapping cameras. It aims to match a specific person/object across different times, places, or cameras, which has drawn a lot of interest in the computer vision community since it is imperative to intelligent surveillance systems to correctly identify the same object when it appears in different frames and cameras. The challenge lies in how to find the same person/object from gallery images where there are background clutter, diversity of poses, occlusion, etc.

Representing the problem by a graph is pertinent since this structure is well adapted to create clusters of identified objects and quickly update the cluster with a new node when it has an existing object/person or create a new cluster when the incoming image is not similar to the existing objects/persons. Moreover, this representation is very suitable for an online retrieval especially with efficient implementation of features and distance computation. Furthermore, this structure avoids exploring all the existing images each time we want to retrieve an incoming image.

In Figure 1, we show a concrete example (that we created using Market dataset) that motivates re-ID task: from different cameras and in different acquisition conditions, intelligent video surveillance should be able to identify a human/target and easily distinguish him from similar persons in the same area. This example reflects the problems of intra-class variation and inter-class similarity, which can be solved with graph-based approaches.

2. Related Work

The target re-identification is composed of several steps; namely bounding box generation, deep metric learning, retrieval of the object of interest using the feature representations extracted using the Re-ID model learned in the deep metric learning step, and re-ranking.

In this section, we will focus on previous work that has been done on object of interest retrieval and re-ranking, as we will focus on this part in our project.

To increase retrieval performance, re-ranking techniques based on computed distances have gained much attention :

In [5], a k-nearest neighbors (k-NN) based approach was used to produce new rank lists and recompute distances. The authors of [1] proposed to jointly learn the context and content information in the ranking list to remove candidates in the top neighbors and improve performance of person re-ID. The work of [3] extended this to revise the initial ranking list with a new similarity obtained from fusion of content and contextual similarity.

Differently from common neighbors, the authors of [2] used reciprocal neighbors (common neighbors that reciprocate in a k-neighborhood) and proposed to compute a contextual dissimilarity measure (CDM).

Zhong et al [8] used a neighborhood-based method that utilizes k-reciprocal nearest neighbors with Jaccard distance in order to improve the reranking accuracy. Part of our work will be inspired by this unsupervised method which is supposed to improve the Re-ID outcome.

Note, while reciprocal list based comparisons provides the current best re-ranking scores, many methods appeared in order to reduce that complexity of recomputing the reciprocal rank lists for each image pair such as the following:

Zhang et al. [7] used an efficient re-ranking method based on two phases. The first phase is based on edge encoding using the notion of reciprocal neighbors. The second phase is based on message propagation using a GNN where neighbor features are aggregated with edge weights to propagate information. In our work we will explore this GNN-based method and test its effectiveness.

In this work [4], the authors proposed a cross-neighborhood re-ranking to increase target retrieval performance. This idea of cross-neighborhood is based on the intuition that for a pair of images, the accumulation of the distances of only the two immediate level neighbors of each image with the other image results in a promising re-ranking.

In our project, we will explore some of the mentioned methods as well as other graph-based methods.

3. Problem Definition

In general, re-ID can be regarded as a retrieval problem. Given a query object/person, we want to search in a gallery for images containing the same object/person in a cross-camera mode. After extracting features using deep metric learning step, a good practice consists of adding a re-ranking step, with the expectation that the relevant images will receive higher ranks. In this project we will focus on this step of re-ranking, where we will try to predict the objects having the same identity as the query given the extracted features and using Link prediction algorithms.

A preliminary work is done by transforming all images into a feature map using a pretrained convolutional neural network (learned in the deep metric learning part). Note that, the feature map has a fixed size depending on the network parameters and allows a homogeneous representation of the dataset even though data have different sources (camera resolution, zoom settings, etc). f_i denotes the feature map of image i, id_i the ID of the object it contains and c_i the camera that acquired the image i.

Images are split into gallery and queries. Gallery images are modeled by a graph $G = (V, E)$: V are nodes representing gallery images $g_1, \ldots, g_n$ while E are edges modeling the link between gallery images and based on the similarity between feature maps of the gallery image. Queries are the images to retrieve and are denoted by $q_1, \ldots, q_m$: the goal is to predict the links that they may have with the gallery images. We aim to output a sorted list of re-ranked gallery images where the stronger the predicted link is, the higher the rank of gallery image is. This allows to have true ID matches in the first ranks and therefore identify the query object/person in the gallery.

Re-ranking differs from initial ranking by the definition of a new distance. In the present work, we denote by $\mathbf{d}(f_i, f_j)$ the cosine distance between image i and the image j in the feature map space defined as:

$$d(f_j, f_j) = 1 - sim(f_j, f_j) \tag{1}$$

where $sim(f_i, f_j)$ is the cosine similarity between the feature map vectors f_i and f_j. Note that by definition of cosine similarity, this distance is positive, bounded by 2 and equal to zero if and only if $f_i = f_j$. Distance $\mathbf{d}$ is used to firstly build the graph then several approaches are applied to deduce a new metric that allows to re-rank the gallery images such that the closest to a certain query are the images that have the same ID.

4. Methodology

Figure 2 summarizes our methodology to solve the problem of re-identification. First, Resnet is trained to give a convenient representation of images (fastreid toolbox was used for this part [1]). Features induce a distance used to transform the gallery into graph. Then, a re-ranking strategy is proposed to correct links and robustify the graph to the noise and the diversity of the gallery. To find a query ID, the query image is projected into the feature space. The links to gallery images are predicted using the re-ranking strategy and the query is re-identified to the closest re-ranked neighbors.

4.1. Object features extractor

Identified images are reduced into features using a deep metric learning based on triplet loss and cross-entropy loss. A part of the dataset is used to train a **ResNet model** to extract features. Unseen identified objects are represented by several images in the gallery and are projected in the features space. Projections are denoted (f_i). They are used either to define and compute the graph edges or to be part of the node features.

[1] https://github.com/JDAI-CV/fast-reid

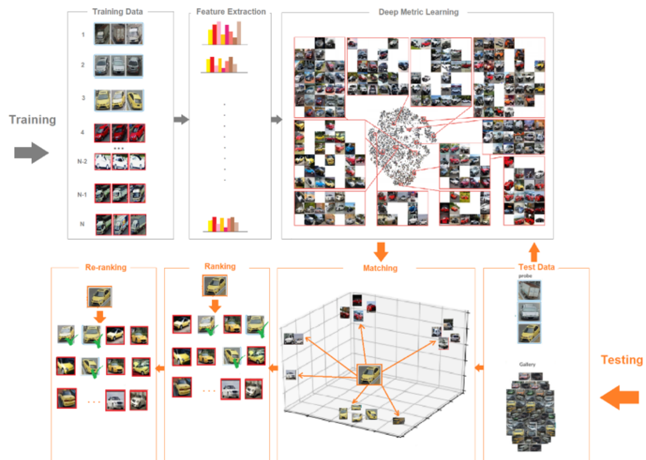

Figure 2: Re-identification Pipeline

4.2. Graphical representation of the gallery

The above-mentioned features are represented as nodes of a graph and edges are built at a first place as an **ϵ-graph** or **knn-graph** based on feature related distance deduced from the cosine similarity as defined in equation 1.

In **ϵ-graph**, a node is linked to all nodes that are ϵ close to it in terms of cosine distance. In **knn-graph**, a node is linked to the k closest nodes in terms of cosine distance. Therefore, node connection is only distance based and does not use the object ID of the nodes. The ranking uses only the information given by the feature maps and the list of sorted closest neighbors may contain mismatches at high ranking positions. In figure 3, we show an example of how

Figure 3: Example of ranking images from the gallery for a given query. In the left, the query image is displayed. In the first row, we sort gallery image by increasing cosine distance to the query image. In the second row, we display the correct matches from the farthest image to the closest.

the distance inducted by the feature map is not sufficient to get enough ID matches in the ranked list. The presence of the bike creates a bias to select images that contain a bike even though it is a different person. One can see that false matches are very close to the query while the ranking fails to predict the link with true ID matches. This example and others motivates the necessity to find a new ranking strategy that bypass the distance in the feature space.

4.3. Re-ranking

At this stage, the aim is to take advantage of the graph structure built using the cosine distance in to order to define a more accurate order relationship between nodes. The topology of the graph is significantly used at this stage to either build graphical features and predict score for edges or to extend the definition of neighborhood. We have tried different approaches to improve the baseline raking:

- k-reciprocal encoding: In this method, we encode the reciprocal neighboring relations between nodes and compute Jaccard distance by comparing the reciprocal neighbors set of each node. This distance is then used to re-rank the gallery images with respect to the query.
- GNN-based Re-Ranking : This method reduces the large time cost of complicated operations in conventional re-ranking methods. This method consists in updating features by the message propagation in GNN.

We expect that these algorithms are more adapted to the task since it robustifies the identification operation to the variability of data such as luminosity, blur, etc.

We also tested other methods in order to see whether they help to improve the performance such as :

- Link prediction-based Re-Ranking(using GNN): this method uses the cosine distance in order to build a graph that will be used in a link prediction task. We then take the output probabilities of our GNN (trained on predicting links between the nodes having the same identity) as a new similarity measure (instead of the cosine similarity).
- Graph-features based re-ranking: this method uses the graph-features (that are homogeneous to a similarity measure) computed on our graph (that is built using the cosine distance) in order to perform the re-ranking.

5. Tested re-raking approaches

5.1. k-reciprocal encoding

For a probe p, we define the k-nearest neighbors $N(p, k)$ as the set of the k closest gallery images.

$$N(p, k) = \{g_0, g_1, \dots, g_k\} \quad (2)$$

where $d(f_p, f_{g_0}) \leq \dots \leq d(f_p, f_{g_k})$. The set of reciprocal neighbors is composed of nodes that belong to $N(p, k)$ and such that p belongs to their k-neighbors set:

$$\mathcal{R}(p, k) = \{g_i | (g_i \in N(p, k)) \wedge (p \in N(g_i, k))\}. \quad (3)$$

The reciprocal neighbors may not include all positive matches because of the variation of acquisition conditions or the existence of strong correlations with other images from the gallery as illustrated by the bike example in figure 3. To address this problem, the set of k-reciprocal neighbors of the probe p is enriched by the l-reciprocal neighbors of its k-reciprocal neighbors that satisfy a certain condition expressed below:

$$\mathcal{R}^*(p,k) = \mathcal{R}(p,k) \cup \mathcal{R}(g,l)$$

$$\forall g \in \mathcal{R}(p,k) \textbf{ s.t. } |\mathcal{R}(p,k) \cap \mathcal{R}(g,l)| \geq \frac{2}{3}|\mathcal{R}(g,l)|$$

Hence, we can define a new distance between nodes known as Jaccard distance and computed by the following equation:

$$\mathbf{d}_J(p,g) = 1 - \frac{|\mathcal{R}^*(p,k) \cap \mathcal{R}^*(g,k)|}{|\mathcal{R}^*(p,k) \cup \mathcal{R}^*(g,k)|} \tag{4}$$

To keep information about the similarity distance, the final distance is jointly aggregated from the similarity distance and the Jaccard distance. This aggregation is parametrized by a hyperparameter λ such that

$$\mathbf{d}_f(p,g) = (1-\lambda)\mathbf{d}_J(p,g) + \lambda\mathbf{d}(p,g) \tag{5}$$

To efficiently compute the Jaccard distance, we encode the reciprocal neighboring relation into a vector. For a probe p, we define a vector $\mathcal{V}_p$ having the size of the gallery:

$$\mathcal{V}_{p,g_i} = \begin{cases} e^{-d(p,g_i)} & \text{if } g_i \in N(p,k) \\ 0 & \text{else} \end{cases} \tag{6}$$

Hence,

$$d_J(p,g_i) = 1 - \frac{\sum_{j=1}^{N} \min(\mathcal{V}_{p,g_i}, \mathcal{V}_{p,g_j})}{\sum_{j=1}^{N} \max(\mathcal{V}_{p,g_i}, \mathcal{V}_{p,g_j})} \tag{7}$$

Finally, we have defined a distance that takes into account similarity between image as well as contextual information of the graph represented by the reciprocal neighboring relation.

We have conducted several experiments to finetune the hyperparameters of this distance. The best configuration on Market 1501 Dataset is the following:

$$k = 20, l = 6, \lambda = 0.1 \tag{8}$$

The results of this method are presented and commented in section 6.4.

5.2. GNN-based re-ranking

The key idea behind this method is that the similarity between images can be represented as a relation graph. It should be noted that the GNN won't be optimized during a training process. It will be used for message passing using a fixed function that captures relation between nodes.

The proposed approach consists of the following two stages. In the first stage, based on the entire image group (query images and gallery images), we construct a graph and encode local information in edges. In the second stage, the proposed GNN propagates messages by aggregating neighbor features with edge weights. The final re-ranked retrieval list is calculated by comparing the similarity of refined node features.

Graph construction First, we construct a graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, v_2, ...v_n\}$ denote nodes which represent all images (queries and galleries) and connected edges represent the similarity between vertices. We compute the cosine similarity matrix on all nodes:

$$S_{i,j} = sim(x_i, x_j) \text{ for}(x_i, x_j) \in \mathcal{V} \times \mathcal{V} \tag{9}$$

To obtain the contextual information of the whole image group, we extract neighbor information from S on the entire graph to refine node features. First we define the adjacency matrix A as:

$$A_{i,j} = \begin{cases} 1 & \text{if } j \in \mathcal{N}(i,k_1) \\ 0 & \text{else} \end{cases} \tag{10}$$

Then, we define the symmetric adjacent matrix $A^* = \frac{A+A^T}{2}$:

$$A^*_{i,j} = \begin{cases} 1 & \text{if } j \in \mathcal{N}(i,k_1) \wedge i \in \mathcal{N}(j,k_1) \\ 0 & \text{if } j \notin \mathcal{N}(i,k_1) \wedge i \notin \mathcal{N}(j,k_1) \\ 0 & \text{else} \end{cases} \tag{11}$$

Finally, we define the i^{th} row of the symmetric adjacent matrix A as the new feature vector of vertex v_i. Thus, we replace the original features with features encoding neighbor similarity.

$$h_i = [A^*_{i,0}, A^*_{i,1}, ..., A^*_{i,n}] \tag{12}$$

To complete the graph construction, we need to define the edges $\mathcal{E}$. We will simply construct a knn graph by connecting top-k_2 edges in S for each vertex and weights $e_{i,j}$ are defined as the cosine score between i and j. We use different values for k_1 and k_2. The first one is used to define new node features based on the neighborhood and the the second one is used to define edges (k_2-nn graph)

Message Propagation The constructed graph will be used to aggregate features of neighbors as follows:

$$h_i^{l+1} = h_i^l + aggregate(\{f_\Theta(e_{i,j}).h_j^l\}) \tag{13}$$

where h_i^l represents the feature of v_i in the l^{th} layer, f_Θ is the function to compute the weight of propagating message and $aggregate$ represents the aggregator types: sum,

$mean$ or max. Authors use a fixed function $f_\Theta(e_{i,j}) = e_{i,j}^\alpha$.

Finally, we derive the final ranking list according to the cosine similarity of refined features.

5.3. Other Tested Methods

We tested other simpler methods than those described above. However, these methods did not improve the re-ranking performance. This may be due to the fact that they rely too much on the characteristics of the neighborhood or the structure of the graph, so that they become vulnerable to noise in the cosine distance. This is different from previous methods where special attention is paid to the reliability of the neighbors.

The first method that was explored is (supervised) link prediction using a GNN[2] architecture, where we try to predict the existence of links only between images of the same identity. This is supposed to help us with the re-ranking as it helps to connect the queries with the true matches and disconnect them from the false matches. We then used the output probabilities of the GNN as a similarity measure (as they are homogeneous to a similarity measure) to compute the ranking list for each query.

We used the cosine distance as an edge feature for the prediction and created the edge labels based on the identities of the nodes (the edges exist only between nodes that have the same identities).

For this GNN-based approach we used a knn-graph with k=10 due to memory limitations (we mentioned above how we created the graphical representation of the gallery).

The second method used the graph features issued from the knn-graphs and ϵ-graphs built using the cosine distance such as a Jaccard distance and Adamic Adar. We tried to combine these graph features directly with the cosine similarity (as they are homogeneous to a similarity measure) or to use them alone to perform the re-ranking (to test their potential to improve the re-ranking results). We think that this method did not improve the performance given that these graph features are vulnerable to the noise in the cosine-distance.

6. Evaluation

In this section we will present the experiments we performed for each mentioned algorithm.

[2] https://docs.dgl.ai/generated/dgl.nn.mxnet.conv.SAGEConv.html?highlight=sage#dgl.nn.mxnet.conv.SAGEConv

6.1. Datasets

Market 1501 Dataset [3] : The Market-1501 dataset is collected in front of a supermarket in Tsinghua University. A total of six cameras were used, including 5 high-resolution cameras, and one low-resolution camera. Field-of-view overlap exists among different cameras. Overall, this dataset contains 32,668 annotated bounding boxes of 1,501 identities. In this open system, images of each identity are captured by at most six cameras.

Figure 4: Market 1501 Dataset

VERI-Wild [4] : A large-scale vehicle ReID dataset in the wild is captured from a large CCTV surveillance system consisting of 174 cameras across one month (30*24h) under unconstrained scenarios.

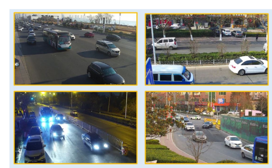

Figure 5: VERI-Wild Dataset

6.2. Evaluations Metrics

CMC : For each query $i \in Q$, all gallery images $k \in G$ are ranked based on the distance $d(i,k)$. Cumulative Match Characteristic (CMC) counts if top-x ranked gallery samples contain the query identity. The rank index of the correct match is denoted as $R_i^{k^*}$. The gallery image k^* has the same identity as the query i. CMC@x is defined as: [6]

$$CMC@x = \frac{\sum_{i=1}^{N} \mathbf{I}(R_i^{k^*} \leq x)}{N} \tag{14}$$

where $\mathbf{I}(.)$ is the measure function.

This metric is also called Rank@x. For instance, Rank@1 shows if top-1 ranked gallery samples contain the query identity.

[3] https://drive.google.com/file/d/0B8-rUzbwVRk0c054eEozWG9COHM/view?resourcekey=0-8nyl7K9_x37HlQm34MmrYQ

[4] https://github.com/PKU-IMRE/VERI-Wild

	Parameters			Evaluation metrics					
	k	l	λ	Rank@1	Rank@5	Rank@10	mAP	mINP	metric.
Baseline	-	-	-	91.25	96.37	**97.60**	77.08	54.18	84.16
Jaccard only	20	6	0	93.86	96.00	96.70	84.00	75.23	88.93
small l	20	4	0.1	93.76	**96.37**	97.09	84.55	75.07	89.15
big l	20	10	0.1	**93.86**	96.00	96.70	84.00	75.23	88.93
Best setting	20	6	0.1	93.81	96.02	96.70	**84.96**	**75.87**	**89.38**

Table 1: Experimental result of re-ranking using k-reciprocal encoding on Market 1501 Dataset

Graph	Parameters	Rank@1	Rank@5	Rank@10	mAP	mINP	Metric	Gallery Edges	Gallery Nodes
fully connected graph	-	90.14	96.13	97.50	72.92	49.87	81.53	31444056	5608
ϵ-graph	$\epsilon = 0.05$	67.22	90.93	95.22	49.16	21.45	58.19	58962	5608
ϵ-graph	$\epsilon = 0.1$	90.14	96.13	97.50	68.74	49.11	79.44	1132814	5608
ϵ-graph	$\epsilon = 0.2$	90.14	96.13	97.50	72.89	49.88	81.52	29100184	5608
KNN-graph	$K = 10$	88.10	95.22	97.30	42.41	17.59	65.26	56080	5608
KNN-graph	$K = 50$	90.14	96.13	97.50	66.45	45.77	78.30	280400	5608
KNN-graph	$K = 100$	90.14	96.13	97.50	67.64	48.51	78.89	560802	5608

Table 2: Graphical representation of the gallery

mAP : For each query $i \in Q$, an average precision (AP) calculates the area under the Precision-Recall curve AP_i Then, mAP evaluates the overall performance:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{15}$$

mINP : The negative penalty (NP) measures the penalty to find the hardest correct match

$$NP_i = \frac{R_i^{hard} - |G_i|}{R_i^{hard}} \tag{16}$$

where R_i^{hard} indicates the rank position of the hardest match, and $|G_i|$ represents the total number of correct matches for query i. For consistency with CMC and mAP, the inverse negative penalty (INP) is preferred. Overall, the mean INP (mINP) of all the queries is represented by:

$$mINP = \frac{1}{N}\sum_{i=1}^{N}(1 - NP_i) \tag{17}$$

Metric : We created this metric (that we will be reporting along with our experimental results) by combining the mAP and Rank@1.

6.3. Graphical representation of the gallery

In this section, we will examine the impact of the parameters of the ϵ-graph or the KNN graph on the re-ranking performance. As mentioned above, using a graph can help to efficiently retrieve true matches without having to explore the entire gallery every time we have an incoming query. This method is also memory efficient as it avoids saving all distances between all images in the gallery.

In the following experiments (on the graphical representation), we used a subset of Market dataset (a subset 220 identities with 5608 gallery images and 2404 queries)

We tried to see the impact of the above mentioned parameters (ϵ for the ϵ graph and K for the knn-graph) on the market dataset as described in the table 2. Notice that for an ϵ-graph where ϵ is sufficiently large (because we keep all distances smaller than ϵ), we get the same performance as a fully connected graph. The same is true for a knn-graph where K is sufficiently high.

These created graphs will be used later to compute graph-features such as Jaccard coefficient and Adamic Adar.

6.4. Re-ranking using k-reciprocal encoding and Jaccard distance

First, note that the efficient implementation of Jaccard distance computation allowed to run the re-ranking result on the whole Market 1501 dataset. We have split the dataset into gallery and queries. Gallery represents 70% of the dataset images while the remaining images are used as queries.

In Table 1, We see that Jaccard distance improves all the metrics compared to the baseline ranking but Rank@10. Combining similarity distance and Jaccard distance induced by reciprocal neighbors gives the best metrics for mAP and mINP. This means we have more true matches in the re-ranked list, even though Rank@5 and Rank@10 metrics decrease compared to the baseline. Overall, This methods allow to confidently and correctly re-ID the queries.

	Parameters				Evaluation metrics					
	# layers	k1	k2	neigh feat	Rank@1	Rank@5	Rank@10	mAP	mINP	metric.
Baseline	-	-	-	✗	90.14	**96.13**	**97.50**	72.92	49.87	81.53
model 1	-	25	-	✓	90.43	95.72	96.02	72.63	58.23	81.53
model 2	1	25	7	✓	**90.77**	96.05	96.72	82.38	64.93	86.57
model 3	2	25	7	✓	90.27	95.80	96.51	**85.44**	**65.10**	**88.35**

Table 3: Experimental result of GNN-based re-ranking on Market 1501 Dataset

6.5. GNN-based re-ranking

We run some experiments using the GNN-based re-ranking method using different configurations. Table 3 summarizes the experimental results obtained on the Market 1501 dataset. In these experiments, we used a subset of Market dataset (a subset of 220 identities with 5608 gallery images and 2404 queries) like in section 6.3.

We observe that using neighbor features instead of the original features for the nodes improved the evaluation metrics especially the mAP, mINP and metric (which is a combination of mAP and Rank@1). This means that this modification has an impact on the precision of our re-ranking model. We notice also that using the GNN to aggregate features using message propagating improved these metrics by a large margin.

6.6. Other Tested Methods

The GNN-based link prediction method that we explained previously did not improve the performance. In fact, it even degraded the performance. For example the Rank@1 dropped from 85.33 to 75.67 on a subset of Market dataset.

The Graph-features based prediction described above did not improve the performance as well. For example using Adamic Adar instead of the cosine distance gives a Rank@1 of 67.97 instead of 90.14 on a subset of Market dataset.

7. Conclusions

In this project, we tested several re-ranking algorithms for target re-identification. These methods use high-confidence retrieved samples to refine the search results using graph-based and neighborhood-based approaches. Two of our methods improved re-ranking performance on the Market dataset, which is a widely used dataset in the state of the art.

In the future, we can combine the tested methods in order to obtain a more rich feature space (distance based features and graphical features) that can be used in the GNN supervised link prediction.

References

[1] J. Garcia, N. Martinel, C. Micheloni, and A. Gardel. Person re-identification ranking optimisation by discriminant context information analysis. *In ICCV IEEE International Conference on Computer Vision*, 2015. 1

[2] H. Harzallah H. Jegou and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. *In Computer Vision and Pattern Recognition,CVPR, pages 1–8.*, 2007. 2

[3] Q. Leng, R. Hu, C. Liang, Y. Wang, and J. Chen. Person re-identification with content and context re-ranking. *Multimedia Tools and Applications*, 2015. 2

[4] M. Saquib Sarfraz and et al. A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 2

[5] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. *In Computer Vision and Pattern Recognition*, 2012. 1

[6] Zheng WANG, Xin YUAN, Toshihiko YAMASAKI, and et al. Re-identification = retrieval + verification: Back to essence and forward with a new metric. 2020. 5

[7] X. Zhang, M. Jiang, Z. Zheng, X. Tan, E. Ding, and Y. Yang. Understanding image retrieval re-ranking: a graph neural network perspective. 2020. 2

[8] Zhun Zhong and al. et. Re-ranking person re-identification with k-reciprocal encoding. 2017. 2