Project/Thesis No.: CSER-23-52

A STUDY ON MULTI-VIEW PERSON RE-IDENTIFICATION USING DEEP LEARNING TECHNIQUES

By

Md. Zahim Hassan

Roll: 1707007

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

March, 2023

A Study on Multi-view Person Re-identification Using Deep Learning Techniques

By

Md. Zahim Hassan
Roll: 1707007

A thesis submitted in partial fulfillment of the requirements for the degree of
"Bachelor of Science in Computer Science & Engineering"

Supervisor:

Dr. Sk. Md. Masudul Ahsan
Professor
Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh.

_____
Signature

Department of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna 9203, Bangladesh
March, 2023

Acknowledgment

All thanks belong to the Almighty Allah, whose kindness and blessings enabled me to fairly complete this thesis work. After that, I humbly acknowledge the valuable suggestions, advice, guidance and sincere co-operation of Dr. Sk. Md. Masudul Ahsan, Professor, Department of Computer Science and Engineering, Khulna University of Engineering & Technology, under whose supervision this work was carried out. His intellectual advices, encouragement and guidance make me feel confident and scientific research needs much effort in learning & applying and need to have a broad view at problems from different perspective. I would like to convey my heartiest gratitude to all the faculty members, official and staffs of the Department of Computer Science and Engineering as they have always extended their co-operation to complete this work. Last but not least, I wish to thank my friends and family members for their constant supports.

Author

Abstract

With the raising concern of the security issues in the modern world, Person Re-identification has the potential to become a major asset in resolving various security and management related issues. Person re-identification network is widely used in person search method where these networks find correct person match from a gallery of images. Person re-identification has gained more attention among the researchers in recent years because of it's use in security and other purposes. Person re-identification is mainly divided into two parts: identifying a person from the image and matching the identified person with another identified person to get the similarity. Existing methodologies in this area focuses on dealing with various challenges of Person Re-identification such as occlusion, lighting, camera viewpoint, clothing changes, etc. from different perspectives. In this thesis, we have used the Siamese model architecture in Person Re-identification. Specifically, a pre-trained model has been used for identifying person and Siamese model architecture has been used for matching. Different feature extractor models for the Siamese architecture have been implemented for learning to generate relevant feature vectors. A new lightweight model architecture for feature extraction has been developed and proposed. In addition, an existing model has been modified to tackle the re-identification problem. Additionally, both model's performances have been analyzed by varying different hyper-parameters and selected the best combination of hyper-parameters. Further, pre-trained models have also been used in the architecture and their performances have been compared with the proposed model. In the whole process, A subset of publicly available MARS dataset is used while ensuring that it resembles all the qualities of the original dataset. Besides, a local testing dataset has been created to analyze the performances in real scenarios. The results show that the proposed model is faster in evaluating images whereas the modified model performs relatively better in evaluation metrics. A major limitation of the model is that it works effectively when person in both compared images are roughly of the same size. More use of different scaled images in training set can reduce the limitation. Further, the use of a simple person detection model can speed up the whole process.

Contents

List of Tables

List of Figures

CHAPTER I

Introduction

1.1 Introduction

As the technology upgrades, security has become the most concerning aspects in the digital era. In recent times, video surveillance has gained much focus for security purposes. One of the key parts of video surveillance is Person Re-identification. There are various approaches of Person Re-identification but they require physical contact. The goal of image-based person re-identification is to predict "match" between two image patches of same person of different pose and "no match" between different person's image patches. In computer vision community, person re-identification(re-id) has emerged as a important sector for research in recent years.

It is estimated by Cisco that the demand for internet video surveillance will increase in future by large factor. Almost every large corporate building, areas, industries, facilities have many security cameras all around it to observe and prevent any of the unwanted events like trespassing, breaking, stealing, unauthorized access and etc. [1]. Automatic Person Re-identification (Re-ID) can help in these situations. Person Re-ID can provide law enforcement agencies with a significant number of benefits in maintaining law and order. It is particularly useful in criminal finding, crowd control, finding lost person, authorizing permissions, prevent stealing, prevention of access in restricted areas, smart home control and etc.

1.2 Problem Statement

In General, a Person Re-ID system involves two major parts. At first all the persons are marked in an image. This part is known as detection/identification part. This part takes an image frame from the source as input and provides with image patches containing a single person as output. After that, these image patches (queries) are matched with previously selected target image/image patch. It is called re-identification part.

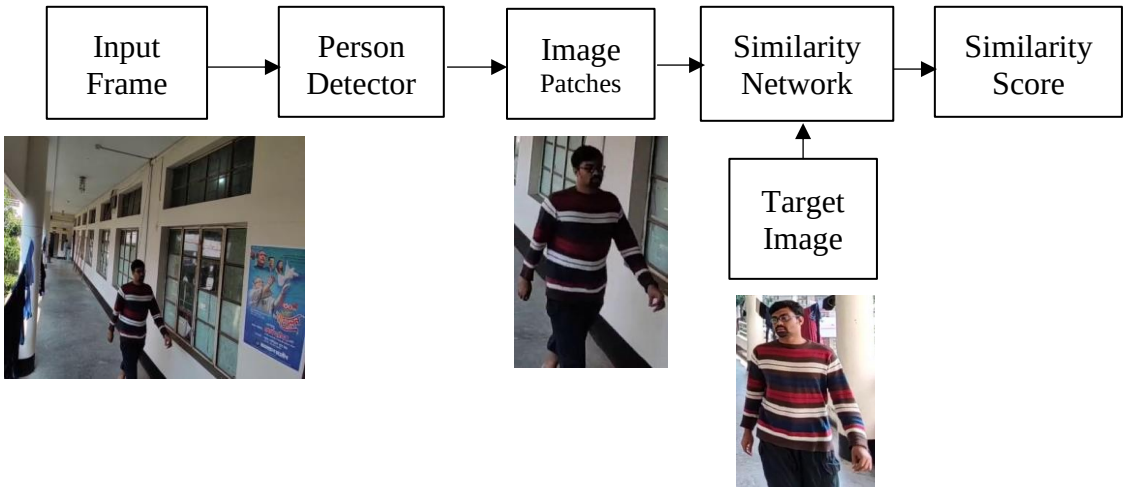

Figure 1.1: Overall Person Re-ID method flow diagram

The output of this part is a similarity score between the target and query images. The overall Person Re-ID flow diagram is presented in Figure 1.1. Here, the frame is provided as an input to the person detector model which provides with the image patch. Then, the target image and are pushed to the similarity model which outputs a similarity score. There are many challenges in this task as well. Viewpoint changes [2], low images quality [3], illumination changes [4], posture variations [5], irregular background [6], occlusions [7], heterogenous modalities [8], clothing changes [9], large and heavy models are some key challenges.

1.3 Objectives

The main objective of this thesis work is to develop a person re-id system by designing a lightweight feature extraction architecture and in the process learn about the various computer vision algorithms, image feature extraction techniques, machine learning, deep neural networks and the implementation of these knowledge to solve real world problems. Key contributions of this thesis work are:

- Design of a lightweight feature extractor model to extract relevant features from the image to recognize a person.
- Finding suitable hyper-parameters for the best use of the lightweight model.
- Modification of an existing feature extractor model for person specific feature extraction.

- Building a test dataset with 194 image pairs of 3 different persons in different environment, lighting and pose.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows:

Chapter 2 presents short summaries on the related research works on Person Re-ID system. The chapter is divided in two sections. There are discussions about image-based methods and video-based methods. Finally, the chapter is concluded with a discussion.

There are some theoretical considerations that need to be reviewed. **Chapter 3** organizes these topics. Convolution neural network and Faster R-CNN model is discussed in this chapter.

Chapter 4 contains the different methodologies that were implemented along with the course of this thesis work. The chapter organizes the details of two different feature extractor architectures. The main process of Person Re-ID is discussed in this chapter.

Chapter 5 organizes the experimental results of the discussed methodologies. The performance of the different models are compared and analyzed with different datasets. Models are analyzed in terms of both qualitative and quantitate measurements.

Finally, **Chapter 6** narrates about the limitations and future works of this thesis. The thesis is concluded with a conclusion.

CHAPTER II

Literature Review

2.1 Introduction

With the introduction of the deep learning models, computer vision research began to observe new speeds. Person Re-ID is also greatly benefited from the deep learning model and is now implemented with many different approaches. The majority of early research efforts are focused on learning distance metrics [10], [11], [12] or handcrafted feature building with body structures [13], [14], [15]. With the availability of large datasets and the advancement in deep learning techniques, there's been introduction of several new techniques that have given Person Re-ID the much-needed momentum. On the basis of the type of data used, Person Re-ID is mainly categorized in two divisions: Image based approaches and Video based approaches.

2.2 Image Based Approaches

Wang et al. [16] focused on using low computational resources for Person Re-ID. In addition, they have handled multiple resolution images. They have used ResNet50 [17] as their base network for feature extraction. From each four stages of the model, they have collected embeddings that represents different resolution of input image. At the end they have fused all these embeddings with a weighted sum to get the final embedding. They have used triplet loss as their learning metric.

Yongxin Ge et al. [18] tried to use the information from global-body and body part features. At first, they used triplet-loss based CNN model for collecting global full body and body part features from images. After that, to exploit more discriminant information for re-identification, they have proposed to use multi-metric loss function as the distance metric learning. At last, gradient descent method has been applied to train the network. Their multi-metric loss function penalizes negative pairs more than positive ones.

Bai et al. [19] suggested a post processing procedure in-form of manifold preserving algorithm. They intended to use the underlying local geometric structural information of data manifold. They aim to learn the smooth similarity using the training set where the data manifold is modeled as the affinity graph. Each time a target is observed a new affinity graph is learned. From there, matching probabilities between the target and the gallery images are obtained.

Chung et al. [20] proposed a network that can learn the spatial and temporal features form the images separately with two separate Siamese networks. One architecture learns spatial features and another learns temporal features, combining these networks again generates the final loss to be minimized. Their approach is motivated by the fact that both spatial and temporal features contain useful discriminative information.

Liu et al. [21] tried to re-identify a person form the top-down images. They presented a sequential network for feature map generation. They have used the network as the key part of the Siamese architecture for Person Re-ID task. However, they only considered images of 5 persons and all images background were the same.

Xuesong Chen et al. in [22], proposed to mine diverse salient features that often remain unnoticed by the conventional methods. To do that, they proposed Cascade Suppression Strategy. With their strategy, they were able to mine salient features from the images and then suppressed previously salient features to mine new salient features. Their Salience Guided Cascaded Suppression Network (SCSN) contains feature aggregation module and salience feature extraction unit. They have used ResNet50 as the base network.

Different from general sequential method for person re-identification, Dong et al. in [23] used the query image patch to generate lower matching proposals. Their Instance Guided Proposal Network (IGPN) uses ResNet50 as the backbone along with Siamese Region Proposal Network and a local relation block. Given a query and a set of scenes, it first generated many proposals and then removes those with low similarity scores. Then the remaining are passed to the Re-ID network.

Wojke et al. [24] proposed to use cosine similarity metric loss instead of triplet loss or magnet loss with Siamese network in metric learning method. Their analysis on different datasets demonstrated that cosine loss performs better that the others.

Angelique Loesch et al. [25] suggested an architecture that jointly performed detection and feature extraction on the same run. The architecture is based on Single Shot Detector [26]. After that, the Re-ID branch is added with triplet loss. ResNet50 is used as the feature extractor. To make their architecture more robust, they have trained with aggregating labeled pedestrian datasets.

Apart from conventional perspective, in [27], Hao et al. tried to address person re-id task from a different perspective than usual. They emphasized on color image to gray scale image matching. Person re-id is performed by learning modality invariant features. They fed the cross-modality images to the feature extractor. It confuses the modality feedback through a confusion learning mechanism. They have incorporated camera-aware and identity-aware marginal center aggregation strategy to enhance the discriminability.

Tianyu He et al. [28] worked with partial input Person Re-ID. Their implemented supervised network learns to differentiate between input image patches with the help of Part-Part Cycle Constraint and Part-Part Triplet Constraint. Their network learns from partial input patches and matches with the patches from the gallery images. To achieve this, the Part-Part Correspondence Learning framework consists of a Gated Layout Rectifier (GLRec) and Corresponding Region Locator (CRLoc) modules.

2.3 Video Based Approaches

Aich et al. [29] introduced Spatio-Temporal Representation Factorization (STRF) unit that can be used along with the 3D convolutional neural network architecture for re-identification. The STRF module contains four factorization units. Each unit contains Feature Factorization Module and Factorized Attention Mask block. The functionality of the STRF module is to generate richer feature representation by taking the input features from the convolutional layers.

Xinqian et al. [30] proposed Temporal Knowledge Propagation (TKP) method. They focused on the scenario when there is only one image and the gallery consist of many videos. In that case, image representation network is forced to fit the output of the video representation network in a shared feature space. Given input video clips, the image representation network finds visual information and the video representation network deals with temporal relations between frames. TKP is done through minimizing TKP loss,

classification and triplet loss. They have used ResNet50 as the base network in both image and video representation networks.

Tianyu et al. [31] came up with Dense Interaction Learning (DenseIL), a hybrid framework to tackle the difficulties in video-based Person Re-ID. The framework comprises of CNN encoder and Dense Interaction (DI) decoder. Given a set of input video clips, the CNN encoder extracts spatial features block by block. These features are horizontally divided into feature patches. Dense Interaction decoder catches the long range inter-frame and intra-frame dependencies following the philosophy of the self-attention and feed-forward layer. Finally, Spatial Temporal Positional Embedding (STEP-Emb) is carried out to get the feature stack for classification.

Chen et al. [32] proposed to incorporate two kinds of attention learning along with existing Re-ID networks to get the person and related body part aware feature maps. Holistic Attention Branch (HAB) built to get the person body feature maps in a cluttered background image. Particle Attention Branch (PAB) helps to distinguish the learned feature maps to different groups each of which has the ability to predict predefined human body key points. The adaptability against occlusions and position variations is boosted by combining both of them.

2.4 Discussion

The reviewed systems mostly used different deep learning techniques in terms of both images based and video-based architecture. Most of the image-based models [16], [22], [23], [25] used ResNet50 as the baseline model for feature extraction from the images. It is due to the ability of the residual blocks to alleviate the problem of vanishing gradient for a large model. Siamese network is most common among the image-based Person Re-ID models to predict the similarity score between target and query image patches. Another key thing to notice that most of the image-based models have used triplet loss as their learning metric. In video-based models, Attention mechanism is most widely adopted. The fusion of spatio-temporal information for generating most relevant person discriminating features is the common practice in video-based Re-ID. An important consideration is that most of the works have focused on incorporating separate modules along with the existing models to generate rich features instead of building a new architecture.

These reviewed systems mainly focused on solving a single problem which indicates that they suffer from other challenges of Person Re-ID. A major drawback of person re-id models is that these methodologies require huge amount of training data to perform well. Although now-a-days there are some benchmark datasets that offer huge amount of data but they are mostly region specific. Incorporating a fair amount of data in a architecture and training with them can cost an ample amount of time. Moreover, these architectures are mostly very much complex and requires a good amount of training time and testing time. Another key downside of the reviewed system is that they are not tested in subcontinental person dataset. In practical application, these issues can decrease the overall performance of the method.

CHAPTER III

Theoretical Considerations

3.1 Introduction

A branch of artificial intelligence and machine learning called neural networks and deep learning aims to replicate the composition and operation of the human brain. They are made up of several linked artificial neurons or nodes that analyze and send information, enabling complicated decision-making and pattern recognition. Deep learning is a term used to describe neural networks with numerous layers, which enable more data abstraction and representation. These models have achieved great success in a number of applications, including speech and image recognition, computational linguistics, and sports.

3.2 Neural Network and Deep Learning

Neural network and deep learning have added numerous possibilities in the field of computational intelligence. Various techniques and algorithms have been developed in this field through the time. Many algorithms have also been developed on the basis of deep learning principles to solve computer vision problems. A neural network is a type of machine learning model that is inspired by the structure and function of the human brain. It is composed of interconnected neurons, that processes and transmit information.

A neural network typically consists of an input layer, one or more hidden layers, and an output layer. The input layer receives the input data, and each subsequent layer transforms the data by applying a mathematical operation followed by a non-linear activation function. The output of the final layer is the prediction made by the neural network. The key feature of neural networks is that they can learn the underlying patterns in the data by adjusting the weights in each layer during the training process. This is typically done using a variant of stochastic gradient descent, which updates the weights based on the error of the network's predictions on a set of training data.

Deep learning is a subset of machine learning that involves training large neural networks with many layers on large amounts of data. It is called "deep" because it utilizes a large number of layers in the neural network architecture. Deep learning models are able to learn complex representations of the data by gradually transforming the input through multiple layers, each of which learns to extract a different level of abstraction. This allows deep learning models to achieve state-of-the-art results on a wide range of tasks such as image classification, computer vision, and many more.

3.2.1 Convolutional Neural Network

A convolutional neural network (CNN) is a type of neural network that is particularly well-suited for image and video analysis. CNNs use a technique called convolution, which involves passing a small "filter" over the pixels of an image, and using the results of this operation to adjust the network's weights. This process is repeated many times, with the network "learning" to recognize increasingly complex patterns in the image. The end result is a network that is able to accurately classify images and detect objects within them.

A. Convolution

Convolution is a mathematical operation that is commonly used in image processing. The convolution operation serves as a way to extract the information from the image and reduce the dimensionality of the data by detecting the important features of the images, and these features can then be used for image classification or object detection tasks.

Convolution is used to extract features from an image by applying a small "filter" to the image's pixels. The filter, also known as a kernel, is a small matrix of weights that is moved across the image in a sliding window fashion. At each position, the values of the filter are multiplied element-wise with the corresponding pixels of the image, and the results are then summed to produce a single output value. This output value is called a "feature map" and it represents the presence of certain features in that particular location of the image. The process of convolution is repeated multiple times with different filters, each time extracting different features from the image. These feature maps are then passed through non-linear activation functions, and multiple layers of convolution and pooling can be used to further extract complex features.

B. Stride, Padding, and Batch Size as Hyperparameter

I. Stride

Stride is a hyperparameter that is used in the convolution operation of a convolutional neural network (CNN). It determines the step size at which the filter is moved across the input image. However, using a larger stride can reduce the size of the output feature map and decrease the computational cost of the convolution operation. This is useful if the goal is to reduce the spatial dimensions of the feature map, or if there is a need to increase the receptive field of the network. When the stride is greater than 1, the filter skips over some of the input pixels, which can lead to loss of information. Therefore, it is important to balance the trade-off between computational efficiency and information preservation when selecting the stride size.

II. Padding

Padding is a technique used in the convolution operation to preserve the spatial dimensions of the input image and prevent information loss at the edges. The choice of padding depends on the requirements of the problem and the architecture of the network.

"Same" padding provides feature map of the same spatial dimension while "valid" padding reduces the output feature map dimension. "Same" padding is useful when the spatial dimensions of the input and output feature maps are required to be the same, whereas "valid" padding is useful when the goal is to reduce the spatial dimensions of the feature maps.

III. Batch Size

The batch size is an important hyperparameter that affects the performance of a neural network during training. A larger batch size helps in parallel processing in cost of more memory. A smaller batch size allows the network to be trained on a more diverse set of samples, which can lead to better generalization performance in cost of more training time. Batch size also reduces the possibility of the model to be overfitted.

C. Pooling

Pooling technique is used in convolutional neural networks (CNNs) to reduce the spatial dimensions of the feature maps and control overfitting. It is typically applied after one or more convolutional layers. Pooling has the advantage of reducing the computational complexity of the network by reducing the number of parameters and computation required.

It also helps to make the features learned by the network more robust to small translations and deformations of the input image.

I. Max Pooling

A filter of fixed size (e.g., 2x2) is applied to a region of the feature map, and outputs the maximum value of the region. It takes the maximum value of each group of pixels, and discards the rest. This also increases the robustness of the network as it only keeps the maximum value of the feature, which is the most dominant feature of the region.

D. Dropout

Dropout is a regularization technique used to prevent overfitting in neural networks. In a CNN, dropout is applied to the fully connected layers in order to randomly drop a certain percentage of neurons during training. This helps to reduce the dependency on any one feature, and improves the generalization of the model. Dropout is typically applied during training, and is turned off during testing.

E. Dense

A dense layer, also known as a fully connected layer, is a layer in a neural network that is connected to all neurons in the previous layer. Each neuron in the dense layer receives input from every neuron in the previous layer and applies a dot product operation with its own set of weights. The output of the dot product operation is then passed through an activation function. This process allows the dense layer to learn complex, non-linear relationships between the input and output.

F. Activation Function in Neural Network

Activation function plays a crucial role in neural networks by introducing non-linearity to the model [33]. It is applied element-wise to the output of a dense layer or a convolutional layer, and transforms the input into an output that can be used as input for the next layer. The activation function allows the neural network to learn a wide range of complex relationships between input and output. It also helps to normalize the output and make it more robust to noise. There are several activation functions named ReLU, Softmax, LeakyReLU, PreLU, ELU, ThresholdReLU, etc. In the following, some popular activation functions are briefly discussed.

I. Sigmoid Activation Function

The sigmoid activation function is a mathematical function that maps the input to an output between 0 and 1. The sigmoid function has an "S" shaped curve, which allows it to smoothly map any input to a value between 0 and 1. The output of the sigmoid function can be interpreted as the probability of the input belonging to a particular class. It is widely used in neural networks for binary classification problems, where the goal is to predict the probability of an input belonging to one of two classes. In the equation 3.1, *S(x)* is the output and *x* is the input variable.

$$S(x) = \frac{e^x}{e^x+1} \tag{3.1}$$

II. Softmax Activation Function

The softmax activation function is a generalization of the sigmoid activation function for multi-class classification problems. It is used to convert a set of input values into a probability distribution over the possible output classes. The sum of all the outputs is 1, which ensures that the outputs are probabilities. It is commonly used in the output layer of a neural network, along with cross-entropy loss, to train a multi-class classification model. Equation 3.2 defines Softmax activation function. Here *i* is the input variable and *j* is the total number of inputs.

$$S(x) = \frac{e^i}{\Sigma_j e^j} \tag{3.2}$$

III. ReLU (Rectified Linear Unit) Activation Function

The rectified linear unit (ReLU) [34] activation function is a widely used activation function in neural networks. It helps to alleviate the vanishing gradient problem, which is a common issue in deep neural networks. The gradient of the ReLU function is either zero or one, which means that it does not suffer from the vanishing gradient problem that sigmoid and tanh can suffer from. Equation 3.3 defines ReLU activation function.

$$y = \max(0, x) \tag{3.3}$$

IV. Tanh Activation Function

The hyperbolic tangent (tanh) activation function is a mathematical function that maps the input to an output between -1 and 1. The tanh function is similar to the sigmoid function, but it has a slightly different property. One of the main differences is that the output of the

tanh function is zero-centered. It is defined by equation 3.4. Here, *S(x)* is the output and *x* is the input variable.

$$S(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.4}$$

G. Optimizer in Neural Network

An optimizer is an algorithm that is used to adjust the parameters of a neural network in order to minimize the loss function. The loss function measures the difference between the predicted output of the network and the actual output, and the optimizer's goal is to find the set of parameters that minimize this difference. Choosing the right optimizer for a neural network depends on the specific problem, the architecture of the network, and the quality of the data. It is common to try multiple optimizers and compare their performance for specific problem. There are many optimizers in neural network such as SGD, RMSprop, Adam [35], Adadelta [36], Adagrad [37], Adamax, Nadam [38], Ftrl, etc.

I. Adam Optimizer

Adam (Adaptive Moment Estimation) is a popular optimizer that is widely used in neural networks. Adam uses moving averages of the parameters to provide a running estimate of the second raw moments of the gradients. It also uses a bias-correction term that helps to stabilize the learning process. Adam algorithm uses two parameters, beta1 and beta2, which control the decay rates of the moving averages of the gradient and the squared gradient, respectively. It has the ability to adapt the learning rate for each parameter, which allows the model to converge more quickly and avoid getting stuck in local minima.

H. Loss Function in Neural Network

In neural networks, the loss function is used to measure the difference between the predicted output of the network and the actual output. The loss function aims to minimize the error function. There are a variety of loss functions in neural network. Among them binary cross-entropy, categorical cross-entropy, sparse categorical cross-entropy and other loss functions are most common. Equation 3.5 defines the binary cross-entropy loss function.

$$\beta_{loss} = \frac{1}{n} \cdot \sum_{i=1}^{n} [y_i \cdot log\ p_i + (1 - y_i) \cdot log(1 - p_i)] \tag{3.5}$$

Here, n = total output size, y_i = actual output value and p_i = predicted probability.

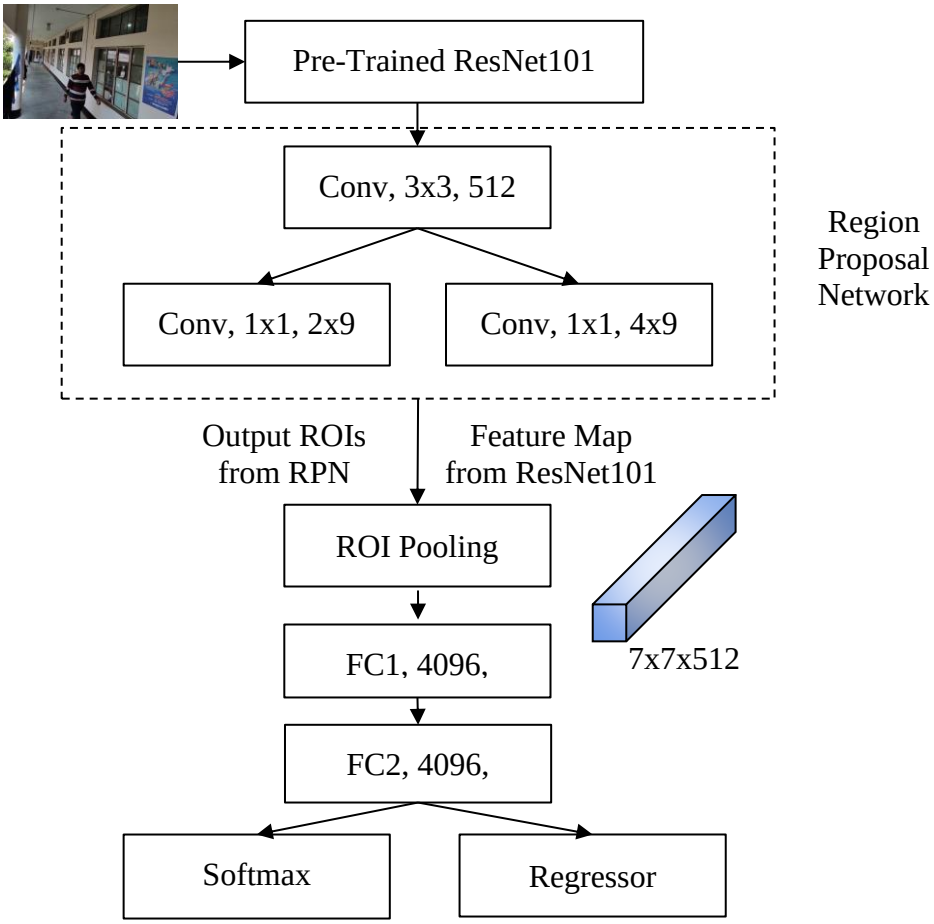

Figure 3.1: Block diagram of Faster R-CNN model

3.2.2 Faster R-CNN Model

Faster R-CNN [39] comes from the family of R-CNNs. It is an improvement of the Fast R-CNN Model. Previously many researchers have suggested to use Faster R-CNN in person detection applications [40], [41], [42]. Figure 3.1 illustrates the block diagram of Faster R-CNN model for person detection. The model comprises of three main components. These are feature map generator network, region proposal network, and region of interest pooling & output layer. Main components are briefly discussed below.

I. Feature Map Generator Network

In this thesis, Faster R-CNN ResNet101 V1 640x640 variant is used which is pre-trained on imagenet [43] dataset for person detection. This model uses ResNet101 as the feature map generator network. The model takes an image with minimum dimension of 600px.

ResNet101 is a deep residual neural network (ResNet) model that is trained on the ImageNet dataset [44]. It has 101 layers. The main idea behind ResNet is the use of residual connections, which allow the model to more easily learn the identity function and avoid vanishing gradients. The ResNet101 model is known for its strong performance on image

Table 3.1: ResNet101 Model Architecture

Layer Name	**Layer Parameters**
conv1	7x7, 64, stride 2
conv2_x	3x3, max pool, stride 2
	$\begin{bmatrix} 1\text{x}1, 64 \\ 3\text{x}3, 64 \\ 1\text{x}1, 256 \end{bmatrix} \times 3$
conv3_x	$\begin{bmatrix} 1\text{x}1, 128 \\ 3\text{x}3, 128 \\ 1\text{x}1, 512 \end{bmatrix} \times 4$
conv4_x	$\begin{bmatrix} 1\text{x}1, 256 \\ 3\text{x}3, 256 \\ 1\text{x}1, 1024 \end{bmatrix} \times 23$
conv5_x	$\begin{bmatrix} 1\text{x}1, 512 \\ 3\text{x}3, 512 \\ 1\text{x}1, 2048 \end{bmatrix} \times 3$
pooling	average pooling 2d
dense	2048-d

classification tasks. As a result, it has been used as a backbone network in many object-detection models. Table 3.1 shows the layers details of ResNet101 model. There is total five types of convolution blocks. Each block contains three convolution layers. There is a skip connection between two successive blocks.

II. Region Proposal Network (RPN)

This network is the main improvement of this model from its ancestors. The region proposals are generated with a network that could be trained and customized based on the detection task. Thus, it produces better region proposals compared to other generic methods. The network takes feature map of the input image as input and produces bounding box co-ordinates & objectness score of the bounding box area as output.

There are anchor boxes of different sizes that slides over the feature map. Those anchor boxes that includes the object and has Intersection over Union (IoU) score greater than 0.5 are selected for further use in later stages. As anchor, three scale with box area 128x128, 256x256, 512x512 with 3 different aspect ratios of 1:1, 1:2, 2:1 is used in sliding mechanism.

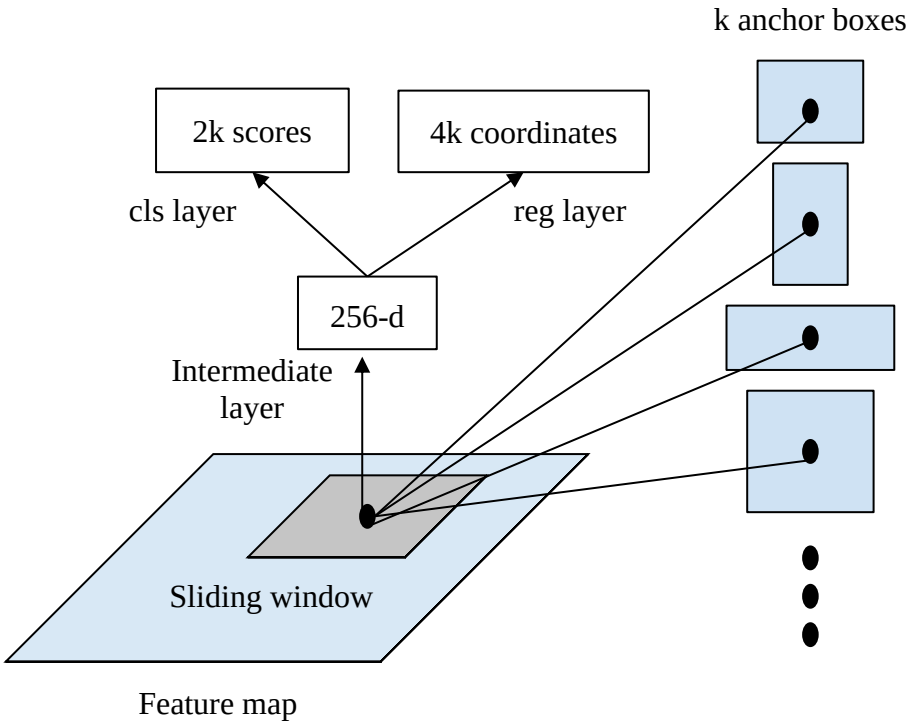

Figure 3.2: Region Proposal Network Mechanism

If the bounding box has IoU more than 0.5 then that is classified as foreground otherwise as background for each anchor box area. Figure 3.2 presents the whole discussed scenario for the RPN.

III. Region of Interest (ROI) Pooling and Output Layer

ROI pooling layer produces fixed size feature map from different size region proposals using max pool. It takes region of interest from the RPN and the feature map from the pre-trained backbone model. Then it produces output vector of size 7x7x512 using 7x7 size pooling kernel with 512 channels.

Then the output of the ROI pooling layer is flattened and passed through two fully connected dense layers. Finally, a regression layer and a softmax classifier is used to predict the confidence score and the class label.

3.3 Conclusion

Neural networks and Deep learning have revolutionized the field of computer vision, enabling computers to perform tasks such as object recognition and image classification with human-like accuracy. These models have been widely adopted in industry, with applications ranging from security systems, self-driving cars, to photo organization software. As the field continues to advance, we can expect even more sophisticated and powerful models to emerge, driving further breakthroughs in the ability of computers to perceive and understand the visual world. Overall, the impact of neural networks and deep learning in computer vision has been significant and will likely continue to shape the way we interact with technology in the future.

CHAPTER IV

Proposed Methodology

4.1 Introduction

Person Re-identification is a computer vision task that aims to match images or videos of a person across multiple cameras or viewpoints. This can be useful in a variety of settings, such as surveillance systems or retail environments. The goal is to be able to track a person as they move through the cameras' field of view, even if their appearance changes due to viewpoint changes, lighting conditions, or clothing. There are several techniques used in Person Re-identification, including feature extraction and matching, deep learning, and metric learning. These techniques are often used in combination to achieve the best performance.

At first, a query image is fed to a person detector model. The functionality of the model is to generate areas where person exists in the image and provide the image patches of persons from the query image. From there on, the image patches are passed to a branch of similarity prediction network where the network generates discriminative features from the patches. Another branch of the similarity network takes the target image and generates the features of the target image. Comparing the features between the query patches and the target image, a similarity score is given to each pair. The detailed block diagram is presented in Figure 4.1.

4.2 Proposed Methods

The proposed methodology has several steps and the steps are briefly explained in following discussions. There are two main steps in the proposed methodology: Person Detection and Person Matching.

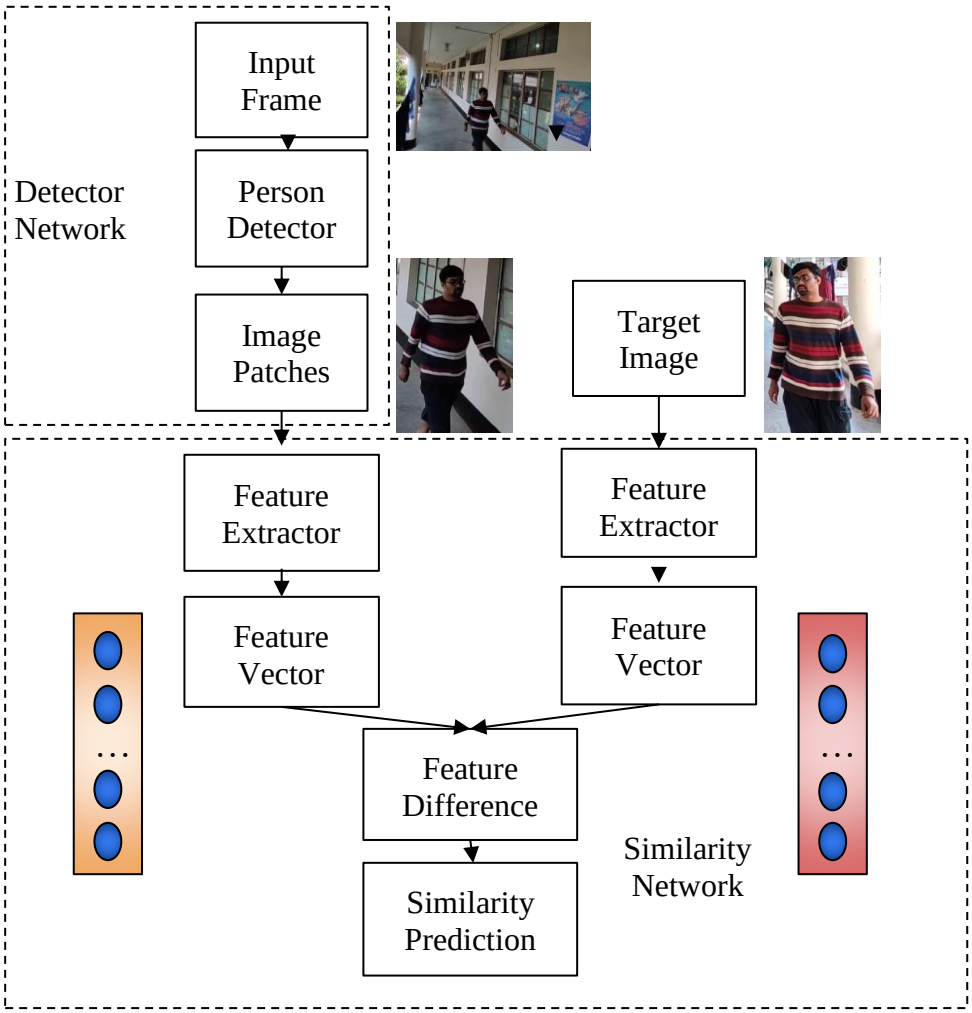

Figure 4.1: Detailed block diagram of Person Re-ID method

4.2.1 Person Detection

Person detection is the process of identifying and locating people in digital images or video streams. This can be done using various techniques such as computer vision, machine learning, and deep learning. There are also many pre-trained models available that can be used to detect people in images and videos. Faster R-CNN model is one of the renowned person detector models. In this thesis work, Faster R-CNN model is used as the person detector. The working principle of Faster R-CNN model is discussed in chapter III. The model takes an input image and it produces the confidence score, class label and box co-ordinates of the detected person in the image.

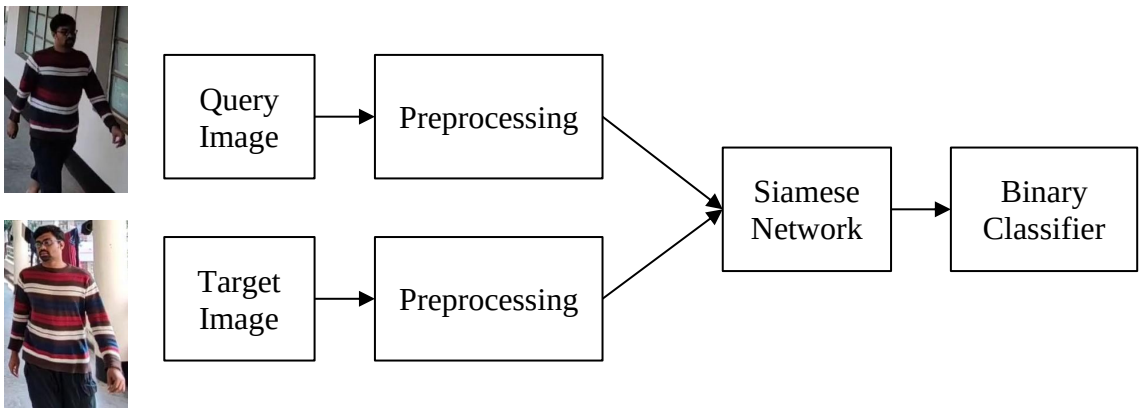

Figure 4.2: Block diagram of Person Matching model

4.2.2 Person Matching

Person matching is the process of identifying and verifying the identity of a person based on their visual appearance. This thesis mainly focuses on improving the performance of the person matching network by finding more robust, low cost, lightweight, feature extractor for the network. To do so, one common practice is to use the Siamese Neural Network architecture. Figure 4.2 demonstrates the block diagram of person matching model. In person matching, the dominant and discriminate features of target and query images are extracted in form of the feature vector. The features are extracted through the Siamese Network and the comparison is done with the distance layer. After that, a binary classifier is used to predict the similarity.

A. Preprocessing

One of the crucial parts of any image processing or computer vision task is preprocessing of the image. After reading the image, its values are converted to floating point numbers. Then the image is resized. For using in the proposed lightweight architecture, the image is converted to 105x105px dimension. In the whole process, RGB images is used. Finally, the image is normalized to 0 to 1 range. Normalization of the image helps the model to learn smoothly and avoids abrupt update in weight.

B. Siamese Network

A Siamese Network is a type of neural network architecture that consists of two or more identical sub-networks, or "Siamese twin" networks, that share the same parameters and are trained together. The output of each network is compared to determine the similarity

between the input images [45]. One of the main uses of a Siamese network is in one-shot learning, which is the ability to recognize objects or individuals after seeing them only once. This is achieved by comparing the input image to a set of reference images, and determining the similarity between the two. Two main components of the Siamese network are feature extractor and distance layer. Siamese network takes a pair of images as input and provides with a vector that represent the feature difference between image pairs. This method is mostly useful for object recognition and person matching task.

I. Feature Extractor

The idea of Siamese network is that through training it learns to generate person specific features. Feature extractor network is the one which is responsible for this process. Through iterative training and backpropagating the errors feature extractor learns to generate person specific features. Table 4.1 organizes the proposed lightweight sequential feature extractor model. The model takes 105x105 dimension images. Passes it through different settings of convolutional, maxpooling, batch nomralization layers to get the final output vector.

The model has around 27 million trainable parameters. Figure 4.3 depicts the visualization of the proposed model architecture. The authors in [45] proposed a Siamese network for one-shot learning for recognition task on the Omniglot dataset. In this thesis, their feature extractor is modified for the purpose of person matching. Different configuration of feature extractor's performance is analyzed is chapter IV. Apart from that, some pre-trained models are also used as a feature extractor. There are many pre-trained models out there for object recognition task. Among them some of the lightweight models are selected for considering as feature extractor. These are:

- VGG16
- VGG19
- ResNet50
- MobileNetV1
- MobileNetV3Large
- NASNetMobile
- MobileNetV3Small

Table 4.1: Lightweight Feature Extractor Architecture

Layer Name	Parameter Settings	Output Shape
Input	shape = (105x105x3)	105x105x3
conv2d	32, 11x11, relu	95x95x32
max_pooling2d	2x2, same	48x48x32
batch_normalization	default	48x48x32
conv2d_1	64, 9x9, relu	40x40x64
conv2d_2	64, 7x7, relu	34x34x64
max_pooling2d_1	2x2, same	17x17x64
batch_normalization_1	default	17x17x64
conv2d_3	128, 5x5, relu	13x13x128
max_pooling2d_2	2x2, same	7x7x128
batch_normalization_2	default	7x7x128
conv2d_4	256, 3x3, default	5x5x256
flatten	default	1x1x6400
dense	4096, sigmoid	1x1x4096

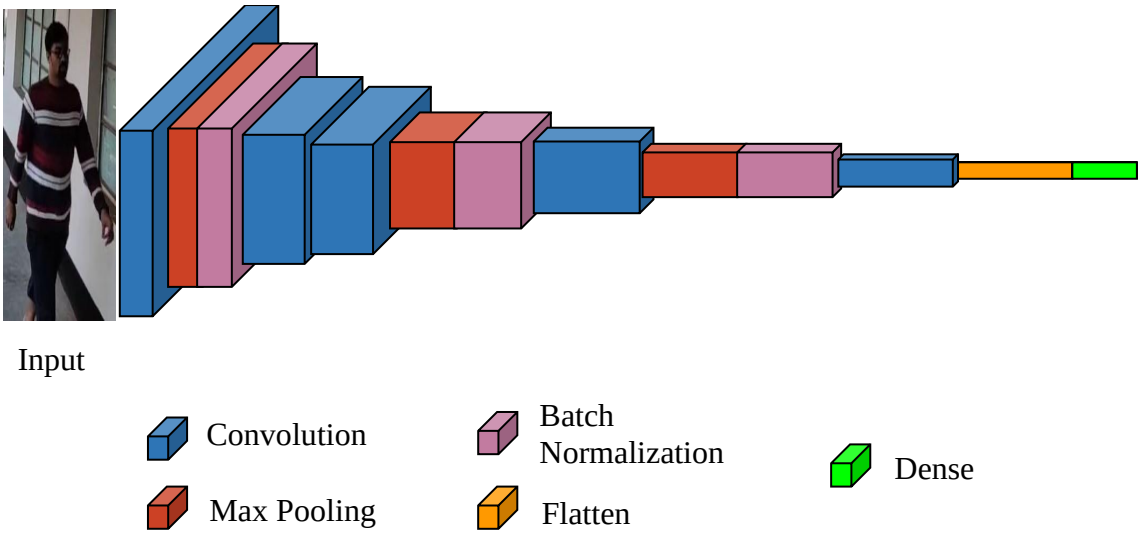

Figure 4.3: Visual representation of the proposed Feature Extractor

Table 4.2: Characteristics of the selected Feature Extractor

Model Name	Top-1 Accuracy (%)	Parameters (in million)
VGG16	71.3	138.4
VGG19	71.3	143.7
ResNet50	74.9	25.6
MobileNet1V1	70.4	4.3
MobileNetV3Small	68.1	2.9
MobileNetV3Large	75.6	5.4
NASNetMobile	74.4	5.3

All the models are pre-trained with imagenet dataset and their characteristics are presented in the Table 4.3. Here, MobileNetV3Small is the most lightweight architecture but MobileNetV3Large provides more accuracy.

II. Distance Layer

The functionality of this layer is to measure the distance between the extracted feature vectors of the query and the target images. Here, L1 distance is used as the metric to measure the distance between feature vectors. Given two feature vectors $x_1$and x_2. The L1 distance. between them is defined as-

$$L1_{distLoss} = \sum_{i=1}^{n} | x_1^i - x_2^i | \quad (4.1)$$

C. Binary Classifier

In this thesis, person marching is considered as a binary classification problem. There are only two class “match” and “not match”. Given two images the task of binary classifier is to predict how much similar the images are by providing with the probability of similarity. For this a dense layer with only one node is used along with the sigmoid activation function. When the output of this layer is greater than 0.5 then it is considered as a match otherwise as not match.

D. Training Objective

The goal of a person matching model is to be able to generate person specific features form the given image. It is achieved through minimizing the overall training loss. Total training loss is define as-

$$TotalLoss = L1_{distLoss} + \beta_{loss} \quad (4.2)$$

Equation 3.5 & 4.1 define the β_{loss} and $L1_{distLoss}$ respectively.

4.3 Conclusion

In this chapter, the methodology behind the entire proposed Person Re-ID system is discussed. Person Re-ID is a feature extraction and optimization problem. The person detection part uses the strength of pre-trained variant of Faster R-CNN. Then the person matching part extracts the person specific features and compares them to provide the final result.

CHAPTER V

Experiment & Results

5.1 Introduction

In this chapter, the results from all the methods that were pursued along the course of this thesis work is represented. The chapter is divided into multiple sections and subsections. The rest of the chapter is organized as follows: Section 5.2 and subsequent subsections 5.2.1 and 5.2.2 present useful information about the used hardware, and the software, respectively, for future reproducibility. Section 5.3 provides a detailed overview of the used datasets. Section 5.4 presents the training configuration. Section 5.5 represent the various Evaluation metrics used in the person matching stage of the model. Section 5.6 and subsequent subsection 5.6.1 present the comparative study among different settings of the models hyperparameters and architecture. Section 5.7 concludes this chapter.

5.2 Experimental Setup

In this section, useful information about the hardware used in the model training process as well as the software information regarding the toolkit versions, OS information, etc. are discussed.

5.2.1 Hardware

In machine learning and computer vision tasks, hardware configuration has a significant impact on performance. During training of a model, it takes a long time to finish. If the model is trained in CPU, it always required more time compared to when the model is trained with GPU support. In this subsection, useful information about the hardware used in the thesis work is presented.

For person detection and person matching-

- Intel Core i5-7200U CPU with 2.5GHz
- RAM 16GB
- GPU NVIDIA GeForce 940MX 2GB

- Operating System windows 10 64-bit

For creating real world test dataset-

- Device Redmi K30
- RAM 6GB
- Octa-core Max 2.2GHz
- Operating system Android
- Cameras 64MP, aperture 1.9
- Video 720p @ 30fps
- Video format ".mp4".

5.2.2 Software

In this thesis work, Python is used as the programming language. Tensorflow and Keras are used to implement the model architecture. Scikit-learn, Numpy, Pandas, Matplotlib, OpenCV, etc. are used as additional libraries. Visual Studio Code is used as the primary code editor. Online tool such as Google Colaboratory, Kaggle Notebook is used as code editor too. Online GPU of Kaggle Notebook NVIDIA Tesla P100 16GB is used as online accelerator.

5.3 Dataset

Images from the MARS dataset [46] are used for training the person marching model. It is an extension of the popular Market-1501 dataset [47]. Images were collected using 6 near-synchronized cameras in the Tsinghua university campus. There were five 1080x1920 HD cameras and one 640x480 SD camera. The dataset contains 1261 different pedestrian's images. Their predefined training set consists 625 identities and the testing set consists 636 identities. In total there are 1,067,516 images. As one of the key objectives is to reduce the need of training images, only the predefined training set of the MARS dataset is considered. From 625 identities, 550 identities are used for training, 50 identities for validation and 25 identities for testing. From each identity 200 images are randomly selected. Thus, the subset dataset contains 110,000 images for train, 10,000 images for validation and 5,000 images for test set. Examples from the subset dataset are shown in Figure 5.1.

Figure 5.1: Example images from subset dataset

As from the above discussion, it is understandable that these images are not taken from the subcontinent. The subcontinent people have different clothing style, appearance, skin tone than others. So, to test the model on the subcontinent people, a different test dataset is created. There are three people in the new test image with two different backgrounds for each person. This dataset contains total 97 images. There are two categories of this dataset. One type contains smaller person images and another set contains larger person images Figure 5.2 displays the images from the new test dataset of different variant.

5.4 Training Configuration

The models are trained in pre-trained manner which means when a timeout occurs models are re-run again. It has been done maximum 2 times. Input to the models is given as pairs of positive and negative examples. A pair consist of an anchor and a positive or negative image. The pairs are created randomly. Every image is considered as an anchor once. A positive pair is created with an anchor and a image from the same directory.

(a) Large

(b) Small

Figure 5.2: Example images from new test dataset ((a) Large, (b) Small)

Similarly, a negative pair is created with an anchor and a image from any of the other directories. So, mathematically total positive pair can be $C(200, 2 * n)$ where n is the number of unique people in the set. But this number is huge so we only one positive pair per anchor image is selected while selecting the positive image randomly. In case of negative pairs, Similarly, mathematically there can be $n * 200 * (n - 1) * 200$ negative pairs. But again, the amount is vast number of pairs which is practically difficult to handle. So only selected one image from the $(n - 1) * 200$ possible candidates. Thus, the process was able to generate 220,000 pairs in the train set, 20,000 pairs in the validation set and 10,000 pairs in the test set. All the sets have the same number of positive and negative pairs. Table 5.1 presents the hyper-parameter settings for training the feature extractor architectures. The models are implemented with early stopping on F1 Score.

5.5 Evaluation Metrics

Binary classification-based approaches use 4 ideas: TP, FP, TN, FN to calculate statistical

Table 5.1: Hyper-parameter setting for training feature extractor

Hyper-parameter	**Value**
Batch Size	16
Learning Rate	1e-4
Optimizer	Adam
Patience	25

Table 5.2: Generalization of confusion matrix

Label	Actual Positive	Actual Negative
Predicted Positive	True Positive	False Positive
Predicted Negative	False Negative	True Negative

measures such as accuracy, recall, precision, and F1 score. These can also be used to evaluate performance of a trained machine learning model.

True Positives (TP): True positives are the cases when the actual class of the data point was 1(True) and the predicted is also 1(True).

Ex: The case where the system correctly detects a match between two images.

True Negatives (TN): True negatives are the cases when the actual class of the data point was 0(False) and the predicted is also 0(False).

False Positives (FP): False positives are the cases when the actual class of the data point was 0(False) and the predicted is 1(True).

Ex: The case where the system detects a match between two different person's images.

False Negatives (FN): False negatives are the cases when the actual class of the data point was 1(True) and the predicted is 0(False).

Ex: The case where the system cannot detect a match between two images of same person.

The representation of these 4 ideas is known as a confusion matrix. Table 5.2 depicts a generalization of the confusion matrix.

Accuracy: Accuracy is a measure that tells how much accurate the result is. It is expressed in:

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (5.1)$$

Precision: Precision gives a measure that tells how much data objects are correctly and positively classified out of the all positively predicted data objects. It is expressed as:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (5.2)$$

Recall: Recall gives a measure that tells how much data objects are correctly and positively classified out of the all actually positive data objects. It is expressed as:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5.3)$$

F-measure: The F measure (F1 score or F score) is a measure of a test's accuracy and is defined as the weighted harmonic mean of the precision and recall of the test.

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (5.4)$$

5.6 Results

The results from all the experimented methods are presented in this section.

5.6.1 Quantitative Results and Analysis of Person Matching

A. Baseline Feature Extractor

Feature extraction is one of the most important part in any recognition task. In this thesis, initially the model proposed in [45] is implemented as the baseline model. The model contains 4 convolution blocks. Each block contains a max pooling layer followed by a convolutional layer. Finally, they used 4096 feature vectors for comparison. The model is trained with patience value set to 10 on F1 score which means the model waits till 10

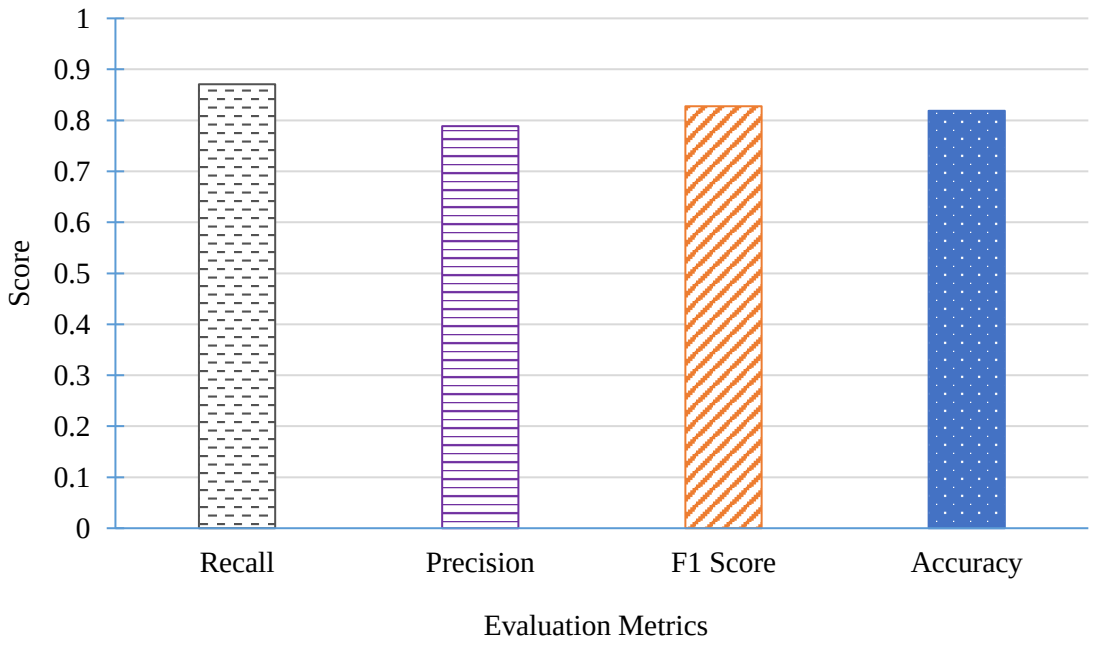

Figure 5.3: Classification performance of Baseline feature extractor

iterations where the validation F1 score cannot exceed the previously best score. All other hyper-parameters are set as per Table 5.1. Figure 5.3 depicts the precision, recall, f1 score, and accuracy found using the similar configuration. It can be seen that even though the structure was designed for recognizing the characters of Omniglot dataset, it recalls the positive pairs fairly well.

B. Modified Feature Extractor

The baseline feature extractor is modified in several ways to observe the effect of the changes on the overall performance. The changes have been done in feature vector size, dropout rate, and number of convolutional blocks. This subsection reports all the findings after the changes have been done.

I. Results after changing Feature Vector size

After the analyzing of baseline feature extractor, the different length of feature vectors is considered for feature extraction to the effect of feature vector size on the performance of the baseline model. The selected feature vector sizes are 512, 1024, 2048, and 4096. Table 5.3 summarizes the findings. From the table, it is clear that the model with feature vector size 1024 outperformed all other models in every evaluation metrics. It achieved 88.98%

Table 5.3: Performance metrics of Baseline model with different length of feature vectors.

Feature Vector Size	Recall	Precision	F1 Score	Accuracy
512	0.8662	0.7906	0.8267	0.8184
1024	**0.8898**	**0.8599**	**0.8746**	**0.8724**
2048	0.8392	0.8047	0.8216	0.8178
4096 (Baseline)	0.8708	0.7886	0.8277	0.8187

recall, 85.99% precision, 87.46% F1 score, and 87.24% accuracy which is the best score is all the evaluation metric category. The most significant changes it has shown in precision where the previous baseline model managed to achieve only 78.86% but the modified best model achieved 85.99% precision score which is more than 7% increase. Apart from that it has also shown more than 5% increase in the accuracy. The most likely reason of performing better with 1024 features is that all the produced features are of the same importance.

II. Results after changing Number of Convolutional Blocks

The baseline feature extractor contains four convolutional blocks. Each block has a convolution layer and a max pooling layer. The functionality of the convolutional layer is to get the feature map form the image using kernels. The max pooling layer is responsible for reducing the feature map size. Sometimes, a smaller number of convolutional blocks fail to extract most relevant features because of less kernel operations on the image. So, another convolutional block is added to the baseline architecture. Now, there are in total five convolutional blocks. Again, on this architecture, the performance with varied length of feature vector is observed. Table 5.4 presents the performance metrics of the model with five convolutional blocks and length of feature vector.

The maximum recall score is found with feature vector size 2048. The model with feature vector size 2048 also gained the most F1 score and Accuracy which is 84.78% and 83.86% respectively. The maximum precision score is gained for the model with feature vector size 4096. The overall F1 score and Accuracy suggest that adding an extra convolutional block doesn’t make a significant change is the model’s performance. However, it also evident that F1 score and Accuracy of the model with different feature vector size have increased

Table 5.4: Performance metrics of Baseline model with five convolutional blocks and different length of feature vector

Feature Vector Size	Recall	Precision	F1 Score	Accuracy
512	0.8396	0.8040	0.8214	0.8175
1024	0.8764	0.7263	0.8233	0.8119
2048	**0.8992**	0.8020	**0.8478**	**0.8386**
4096	0.8602	**0.8191**	0.8391	0.8351

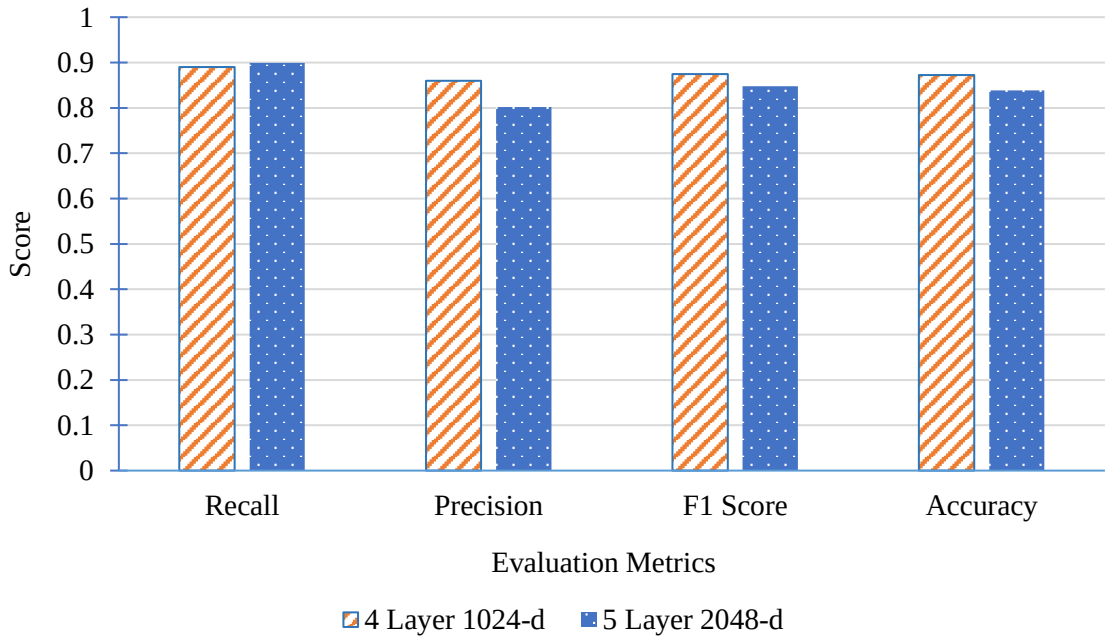

Figure 5.4: Comparison between the 4 Layer best and 5 Layer best architecture

some cases. This drives to the conclusion that after the fourth convolutional block there isn't much information left in the image to extract. Figure 5.4 depicts the comparison between the best models of four convolutional block architecture and five convolutional block architectures. It advocates that the best model of four-layer architecture has higher generalization ability than its competitor.

III. Results after changing Dropout Rate

Dropout layer allows the model to avoid overfitting issues by randomly dropping out neurons during training. The baseline model doesn't incorporate any dropout layer. To measure the performance of the feature extractor with dropout, a dropout out layer is introduced just before the final dense layer which generates output features. So, the model architecture now has 5 convolutional blocks then a flatten layer and a dropout layer and finally, a dense layer with 4096 nodes. Table 5.5 organizes the performance with different dropout rates.

The results indicates that with the increase of dropout rate, recall, F1 score, and accuracy decreases. It is observed in case of precision that as the dropout rate increases precision also increases. These phenomena can be interpreted as that when nodes are randomly dropped then model limits its prediction in case of positive pairs. Consequently, it also predicts less false positives. The most accuracy is observed with no dropout effect. It is 83.51%. Also, the most F1 score is observed with no dropout model with 83.91% score. The trend lines in Figure 5.5 clearly indicates that as the dropout rate increases both accuracy and F1 score decreases. The decrease of F1 score is steeper than that of accuracy score.

C. Proposed Feature Extractor

Long convolutional neural network layer suffers from unstable gradients and slow convergence. It is known as Internal Covariate Shift phenomenon in deep neural networks. It occurs in the training process.

Table 5.5: Performance metrics of Baseline model with five convolutional blocks and different dropout rate

Dropout Rate (%)	Recall	Precision	F1 Score	Accuracy
0	0.8602	0.8191	**0.8391**	**0.8351**
20	**0.8706**	0.7987	0.8331	0.8257
30	0.7934	0.8201	0.8065	0.8097
40	0.7616	**0.8364**	0.7972	0.8063
50	0.6498	0.8213	0.7256	0.7542

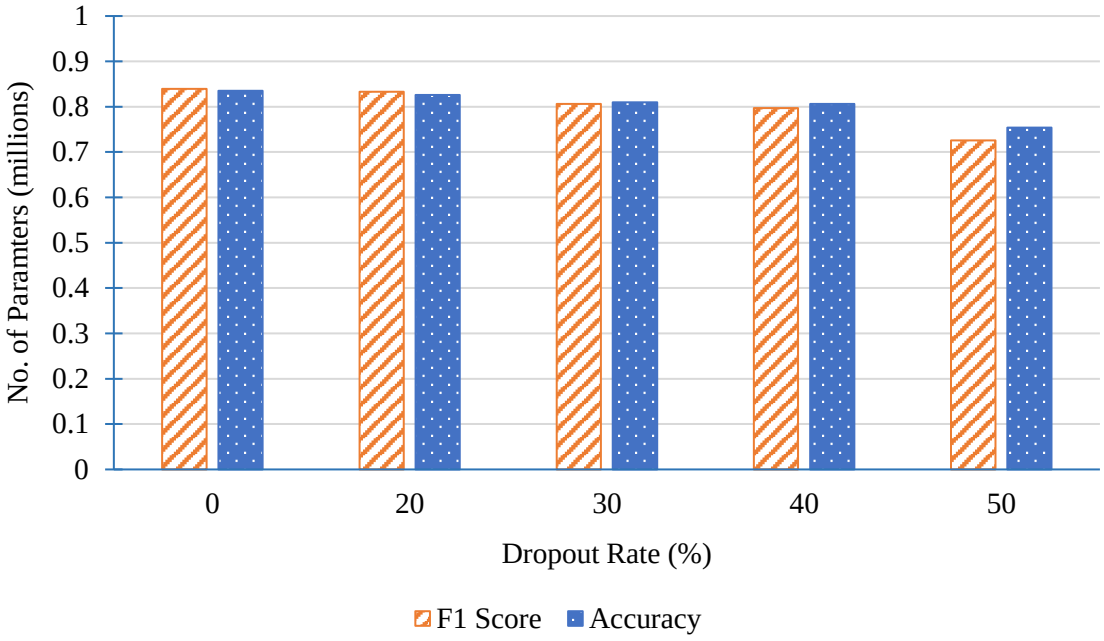

Figure 5.5: Comparison of performance of Baseline model across different dropout rates

Batch normalization is a technique that helps to mitigate this issue by normalizing the activation of each layer. The input of each layer is normalized to have zero mean and unit variance. It helps to speed up the convergence of training and improves the stability of the network.

The proposed architecture is presented in Table 4.2 and visualized in Figure 4.5. The architecture introduces convolutional blocks which contains a convolutional layer, then a max pooling layer and finally a batch normalization layer. The inclusion of batch normalization layer in the architecture is done to reduce the unstable gradients and speed up the convergence. It makes changes in the weight of the architecture in a smooth fashion. The proposed architecture is explored by changing several hyper-parameters. This section organizes and explains the results. The model's performance is observed by changing the feature vector size and the number of batch normalization layers for each feature vector size. The model is trained with the configuration discussed on Table 5.1.

Figure 5.6 shows the validation F1 score curve during the proposed model training. The proposed architecture has a total of three batch normalization layers. The architecture's performance is evaluated by changing the number of batch normalization layers in use.

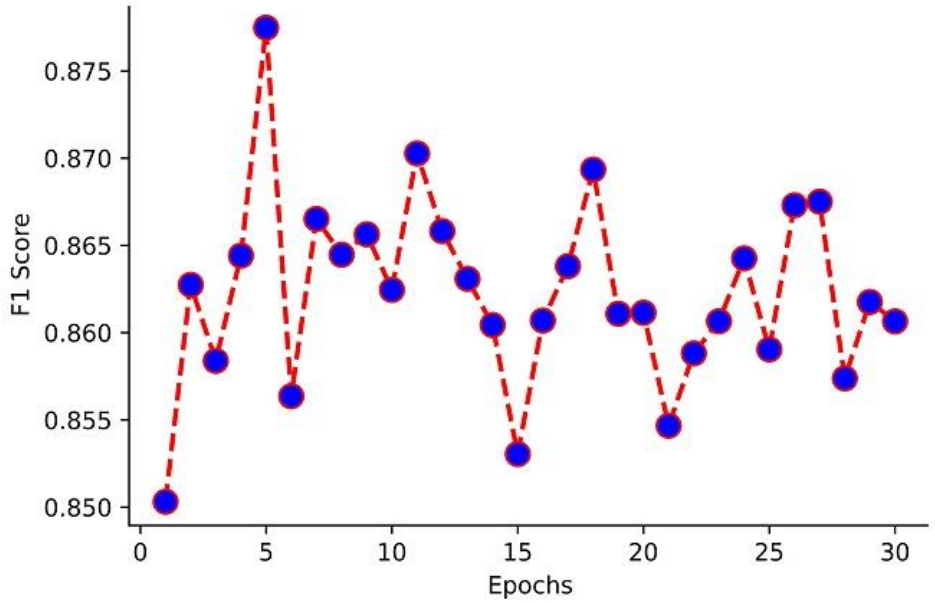

Figure 5.6: Validation F1 Score curve during proposed model training

There is total 3 categories: no batch normalization layer, 1 batch normalization layer, and 3 batch normalization layers.

I. Results after using No Batch Normalization Layer

Table 5.6 demonstrates the model's performance without any batch normalization layers. The model's feature vector size is varied with 512, 1024, 2048, and 4096 lengths.

From the table, it can be seen that proposed model achieved a better performance is all the category than any of the previous versions that have been already discussed. Most accuracy, F1 score, and precision is achieved by the model with 512 feature vector size with 86.03%, 86.50%, and 83.54% score respectively. The most recall value is recorded from the model with feature vector size 1024. Recall values are almost closer to each other for all the model but the main changes are observed in terms of precision, F1 score, and accuracy.

II. Results after using One Batch Normalization Layer

Another form of layer comprises of only one batch normalization layers. In this architecture only the second batch normalization layer is used and the rest are commented. Table 5.7 summaries the result for this case.

Table 5.6: Performance metrics of proposed Feature Extractor architecture without any batch normalization layer by varying the feature vector size

Feature Vector Size	Recall	Precision	F1 Score	Accuracy
512	0.8958	**0.8364**	**0.8650**	**0.8603**
1024	**0.9156**	0.7826	0.8439	0.8306
2048	0.8974	0.7823	0.8416	0.8331
4096	0.9104	0.8004	0.8517	0.8415

Table 5.7: Performance metrics of proposed Feature Extractor architecture with one batch normalization layer by varying the feature vector size

Feature Vector Size	Recall	Precision	F1 Score	Accuracy
512	0.8982	0.7950	0.8434	0.8333
1024	0.908	**0.8396**	**0.8724**	**0.8673**
2048	**0.9416**	0.7985	0.8641	0.852
4096	.0.9068	0.7860	0.8421	0.83

From Table 5.7, it is observed that the model with feature vector size 1024 performed better than other models in three different categories (precision, F1 score, accuracy). It has gained 83.96%, 8724%, and 86.73% in precision, F1 score and accuracy respectively. It's closest performed model which has feature vector size 2048 achieved 94.16% recall score. This model scored less in precision metric than the overall best model if this case.

III. Results after using Three Batch Normalization Layer

Table 5.8 sums up the findings of the model with three batch normalization layers and different number of feature vector size. As the number of batch normalization layer increase, the model with more feature vector are performing well. In the Table 5.8, it is observed that model with feature vector size 4096 outperformed all other models in precision, f1 score, and accuracy score category.

Table 5.8 Performance metrics of proposed Feature Extractor architecture with three batch normalization layers by varying the feature vector size

Feature Vector Size	Recall	Precision	F1 Score	Accuracy
512	0.9204	0.7924	0.8516	0.8396
1024	**0.9324**	0.8067	0.8650	0.8545
2048	0.922	0.8175	0.8668	0.8583
4096	0.9288	**0.8215**	**0.8717**	**0.8635**

The model gained 82.15%, 87.17%, and 86.35% score in precision, f1 score and accuracy respectively. Another important thing is that model with feature vector size 1024, 2048, and 4096 have all performed almost similarly.

IV. Finding the Best Version of the Proposed Architecture

Performance of different models have been analyzed in the previous sections. Here, the best version of the proposed model is found out. Table 5.9 presents the best models of each category from the perspective of the number of normalization layers used in training.

From the Table 5.9, It can be noted that all of the model has performed very closely. The model with feature vector size 4096 and three batch normalization layers (4096-Three-BN) have shown a significant improvement in recall with 92.88% score. Unfortunately, the model has also managed to gain the least precision score which is 82.15% among all the models. As a whole, the model with 1024 features and one batch normalization layer has performed better than other models in precision, f1 score, and accuracy with 83.96%, 87.24%, and 86.73% score respectively. A key observation is that all the model achieved almost similar accuracy. An important finding to take a note of is that as the number of batch normalization layer increases the overall performance of the model also increases. It proves the fact that using batch normalization layer makes the training process more stable and better convergence.

C. Performance of Pre-trained models as Feature Extractor

Pre-trained models are machine learning models that have already been trained on a large

Table 5.9: Comparison of best models with different number of normalization layers and feature vector size

Model	Recall	Precision	F1 Score	Accuracy
512-W/O-BN	0.8958	0.8364	0.8650	0.8603
1024-One-BN	0.908	**0.8396**	**0.8724**	**0.8673**
4096-Three-BN	**0.9288**	0.8215	0.8717	0.8635

dataset and are available for use at any time. The purpose of pre-training is to leverage the knowledge learned from the large dataset and use it as a starting point for training on a smaller, related task. This approach can save time and resources compared to training a model from scratch, and often results in better performance on the target task.

Based on this philosophy, some of the lightweight pretrained models are used as the feature extractor. The reason for choosing the lightweight models is that as two streams if the same model is used simultaneously it takes some time to process the image and produce feature vector. So, it is better to use a lightweight feature extractor model that can perform quickly and same a fair amount of time. The selected pretrained models are presented in section 4.3.2.

While using the pre-trained models, at first the classification layer is dropped. Then, a dense layer is added to produce the feature vector. During training, the base architecture of the pre-trained models are freeze and never trained. Only the dense layer is trained with the subset train dataset. Finally, the performances are observed by varying the feature vector size between 1024 and 2048. The findings are presented in Table 5.10 and 5.11 with VGG16 and ResNet50 as feature extractor. In Table 5.10, VGG16 model achieve best scores in all the evaluation metrics. It has achieved 78.51% F1 score and 76.99% accuracy score. The reason for VGG16 to perform well is that it has a large number of layer's and with a large number of data is given for training it can perform more computation on data.

In Table 5.11, It can be summed up that VGG16 has performed better than ResNet50 with feature vector size 2048. It has gained 82.74% and 81.63% score with F1 score and accuracy.

Table 5.10: Comparison of pre-trained models with 1024 feature vector size

Model	Recall	Precision	F1 Score	Accuracy
VGG16	**0.8406**	**0.7364**	**0.7851**	**0.7699**
ResNet50	0.784	0.6560	0.7143	0.6865

Table 5.11: Comparison of pre-trained models with 2048 feature vector size

Model	Recall	Precision	F1 Score	Accuracy
VGG16	**0.8804**	**0.7803**	**0.8274**	**0.8163**
ResNet50	0.7484	0.7056	0.7264	0.7181

Figure 5.7 shows how the addition of more feature changed the performance of the model. In case of VGG16, it has gained more evaluation score in all the metrics as the number of feature vector size increases. With 2048 feature vector size, it has improved 13.2%, 7.77%, 10.1%, and 9.82% in recall, precision, f1 score and accuracy respectively. It suggests that using 2048 feature vector size is bound to provide better result with pretrained models.

As the Figure 5.7 suggest that using 2048 feature vector size with pretrained model performs better, so few other pre-trained model's performances are also evaluated on the subset test data after training with subset train data. Table 5.12 summarizes the results.

From the Table 5.12, it is evident that VGG16 is the undisputedly better model than any others. It has managed to get the best score in all the four evaluation categories. It has gained 88.04%, 78.03%, 82.74%, and 81.63% score in recall, precision, f1 score, and accuracy respectively.

D. Performance Comparison between Modified and Proposed Architecture

Previously, best configurations for modified and proposed architecture are found out with the detailed performance analysis. In this section the best models from each analysis are compared with each other.

Table 5.13 compares among the best performed pretrained model, the best performed modified model, and the best performed proposed model. It is observed from Table 5.13

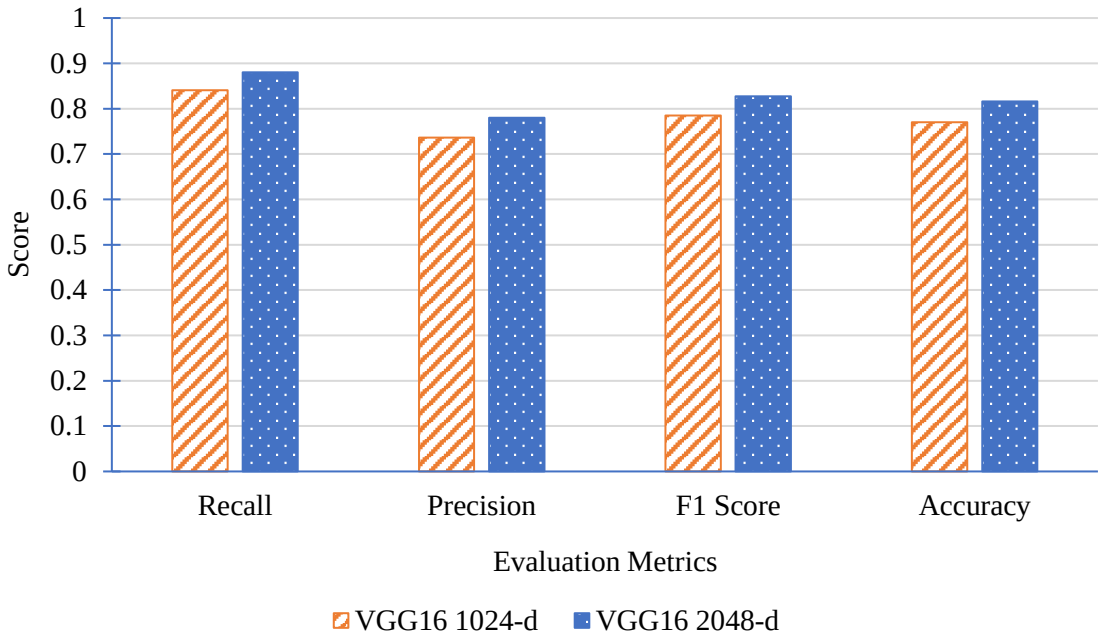

Figure 5.7: Comparison of pre-trained VGG16 models with different feature vector size

that 1024-Modified model has performed comparatively better among all of the three models. It has gained 85.99% precision score, 87.46% F1 score, and 87.24% accuracy score. However, in terms of recall, the proposed 1024-One-BN model has performed significantly better than other two models. It has gained a 90.8% recall score.

Number of trainable parameters is one of the key concerns in any deep learning model training. As the model gets deeper the number of parameters also increases. This increased number of parameters makes the training process take longer. It also increases the evaluation time which is a key issue for implementing the system in practical application. Table 5.14 depicts the parameter difference between modified and proposed networks. Figure 5.8 visualizes the changes using bar plot.

It is also evident from the table that the model managed to get almost similar score with the 1024-Modified model in other metrics. Unfortunately, it couldn't perform significantly in terms of precision metrics. So based on evaluation metrics, it can be concluded that 1024-Modified model comes as the overall best performing model.

From the Figure 5.8, it is clearly visible that the proposed model has fewer number of parameters. It can also be noted that as the size of feature vector increases the number of parameters of the proposed model also increases with a less steep slope than the modified

Table 5.12: Comparison of different pre-trained models with 2048 feature vector size

Model	Recall	Precision	F1 Score	Accuracy
VGG16	**0.8804**	**0.7803**	**0.8274**	**0.8163**
VGG19	0.8478	0.6537	0.7382	0.6994
ResNet50	0.7484	0.7056	0.7264	0.7181
MobileNetV1	0.858	0.7658	0.8092	0.7879
MobileNetV3Small	0.8444	0.7025	0.7669	0.7434
MobileNetV3Large	0.8104	0.6967	0.7492	0.7288
NASNetMobile	0.8594	0.7011	0.7722	0.7465

Table 5.13: Comparison of best models

Model	Recall	Precision	F1 Score	Accuracy
VGG16	0.8804	0.7803	0.8274	0.8163
1024-Modified	0.8898	**0.8599**	**0.8746**	**0.8724**
1024-One-BN	**0.908**	0.8396	0.8724	0.8673

Table 5.14: Comparison between the modified and proposed models based on number of trainable parameters.

Feature Vector Size	Parameters Modified Architecture (in millions)	Parameters Proposed Architecture (in millions)
512	5.9	**4.1**
1024	10.6	**7.4**
2048	20	**13.9**
4096	38.9	**27**

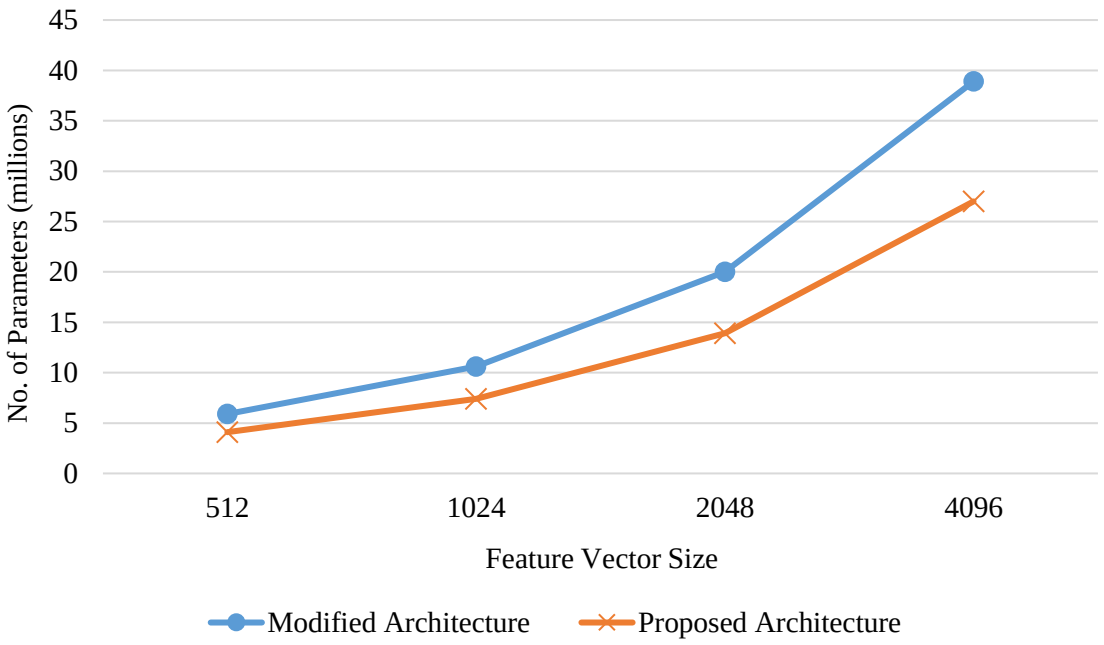

Figure 5.8: Comparison on number of parameters between modified and proposed models

Table 5.15: Train and Evaluation time comparison between modified and proposed models

Model	Average train time per epoch (ms)	Average evaluation time per image (ms)
1024-Modified	310	9.4
1024-One-BN	**24**	**0.04**

Table 5.15 shows the comparison between train and evaluation time of the proposed and models. The result from Table 5.15 suggest that the proposed model is more than 200 times faster than the modified model. So, the proposed model is more suitable in practical applications.

E. Performance Comparison using New Test data

To know how the models really performs in practical applications, a small test dataset is created which is discussed in section 5.3. Different models' performance on the new test dataset is presented in Table 5.16 and in Table 5.17.

Table 5.16: Different models performance on Large category of New test data

Model	Recall	Precision	F1 Score	Accuracy
VGG16	0.7319	**0.6454**	0.6869	0.6649
VGG19	0.7423	0.6421	0.6792	**0.6794**
ResNet50	0.5773	0.5	0.53	0.5
MobileNetV1	0.7423	0.5902	0.6575	0.6135
MobileNetV3Small	0.7938	0.6209	**0.6968**	0.6546
MobileNetV3Large	0.8350	0.5548	0.6667	0.5824
NASNetMobile	**0.8763**	0.5414	0.6692	0.5670
1024-Modified	**0.8763**	0.5483	0.6746	0.5773
1024-One-BN	0.7938	0.6062	0.6874	0.6393

In Table 5.16, NASNetMobile and 1024-Modified models have jointly achieved the maximum recall score 87.63%. The VGG16 model has achieved 64.54% precision score which is the maximum. Maximum F1 score 69.68% is achieved by MobileNetV3Small. The VGG19 model gets the maximum accuracy that is 67.94%.

Table 5.17 presents the evaluation scores of different models on small category of new test dataset. The maximum recall score of 90.72% and F1 score of 66.92% is achieved by MobileNetV1 whereas the maximum precision score of 59.52%, and accuracy score of 58.24% is achieved by NASNetMobile. In general, from the two table, it can be seen that models tend to perform better with images where person appeared larger than the images where person appeared smaller. The reason for proposed model and modified model not performing better in this test is that the subset dataset didn't have much variations in lighting and scaling.

Most of the images have the same lighting condition and mostly in all the images person have appeared in same size. Again, the reason of pre-trained models to perform better is that they already trained on various type of images that had more variations in lighting and scaling.

Table 5.17: Different models performance on Small category of New test data

Model	Recall	Precision	F1 Score	Accuracy
VGG16	0.5979	0.5087	0.5497	0.5103
VGG19	0.8649	0.4907	0.6315	0.4948
ResNet50	0.5567	0.5192	0.5373	0.5206
MobileNetV1	**0.9072**	0.5301	**0.6692**	0.5515
MobileNetV3Small	0.6082	0.5619	0.5841	0.5670
MobileNetV3Large	0.8453	0.5222	0.6456	0.5360
NASNetMobile	0.5154	**0.5952**	0.5524	**0.5824**
1024-Modified	0.6185	0.5454	0.5797	0.5515
1024-One-BN	0.8144	0.4876	0.61	0.4793

5.6.2 Qualitative Results and Analysis of Person Matching

As previously discussed, Person Re-ID has two parts: person detection and person marching. It is already stated that this thesis work only focuses on the person matching part. Figure 5.9 depicts some example outputs using the modified model with the subset test dataset. All the produced positive outputs are correctly predicted with high probability. Also, all the produced negative outputs have very low probability.

Figure 5.10 demonstrates some examples output images using the proposed model. It is clearly visible that proposed model learnt more about handling scale invariance as it has performed better in case of scaled images too. With the same target modified model predicted 0.0005 and 0.87 probability score in Figure 5.10 (a) and (c) respectively whereas proposed model predicted 0.00001 and 0.99 confidence score in Figure 5.10 (a) and (c) respectively.

The example outputs of the models in terms of new test data are presented in Figure 5.11 and Figure 5.12. Figure 5.11 displays the outputs using modified model and Figure 5.12 displays the outputs with proposed model with the new test data. From Figure 5.11 and 5.12, it is seen that modified and proposed model hasn’t performed well in new test data (Large).

Target (a) Query 1 (5e-4) (b) Query 2 (0.02) (c) Query 3 (0.87) (d) Query 4 (0.98)

Figure 5.9: Example outputs using modified model on subset test data

Target (a) Query 1 (1e-5) (b) Query 2 (5e-5) (c) Query 3 (0.99) (d) Query 4 (0.99)

Figure 5.10: Example outputs using proposed model on subset test data

In Figure 5.11, the modified model incorrectly predicted high confidence for 5.11 (a) image which doesn't match to target image. But in case of 5.11 (b), the modified model correctly predicts low confidence score. Besides, for 5.11 (c) & (d), the predictions are correct. In Figure 5.12, the proposed model failed in two cases: 5.12 (a) & (d) due to mostly illumination imbalance. But the noticeable thing is, proposed model provides greater confidence score when it predicts output correctly. It can be seen in case of 5.12 (b) & (c).

Target (a) Query 1 (0.85) (b) Query 2 (0.26) (c) Query 3 (0.96) (d) Query 4 (0.87)

Figure 5.11: Example outputs using modified model on new test data (Large)

Target (a) Query 1 (0.92) (b) Query 2 (0.12) (c) Query 3 (0.99) (d) Query 4 (0.01)

Figure 5.12: Example outputs using proposed model on new test data (Large)

Figure 5.13 and Figure 5.14 presents modified and proposed models' performance with the small category of new test data. In each figure, a target and three query images are there. In Figure 5.13, it can be summarized that the modified model has incorrectly predicted the confidence scores in two cases: (a) & (b). However, it has predicted high confidence when the query is more similar to target which is presented in 5.13 (c). In Figure 5.14, the proposed model has managed to predict all the output correctly. It has predicted more than 0.5 confidence score in (b) which indicates a match and in rest of the cases it has managed to

Target

(a) Query 1 (0.56)

(b) Query 2 (0.36)

(c) Query 3 (0.98)

Figure 5.13: Example outputs using modified model on new test data (Small)

Target

(a) Query 1 (0.0002)

(b) Query 2 (0.69)

(c) Query 3 (0.92)

Figure 5.14: Example outputs using proposed model on new test data (Small)

predict low confidence score for a not match and very high probability for a match presented in 5.14(a), (b), (c) respectively. It proves that proposed model predicts strong confidence score when it predicts correctly.

Figure 5.15 shows the overall steps of how a target image is matched with a query image from a video frame/image to get the output. Here, (a) is the target image, (b) is a frame where the person is detected and the similarity score of the target and query patch is presented in (c). The example images are taken from a video frame vehicle analyzing the model's performance on video clips. Processing each frame in a clip is a computational heavy task.

5.7 Conclusion

In this chapter, qualitative and quantitative results of the different person matching model have been analyzed. On the subset test dataset, the best performing model is found to be the modified model with 1024 feature vector size. It has gained 85.99% precision sore, 87.46% f1 score, and 87.24% accuracy. The proposed model has achieved 90.8% recall score. It

(a) Target person

(b) Target person detected in a frame

(c) Result (0.88)

Figure 5.15: Demonstration of the proposed Person Re-identification system

has also found out that pre-trained feature extractors perform poorly in this task. Between proposed and modified models, it is observed that proposed model is the lightweight with 7.4 million trainable parameters. It is also proved that the proposed model performs more than 200 times faster than the modified model. However, while testing with new test dataset it is observed that both proposed and modified model struggles to perform well. The main reason for this issue is that the new test data has various scales of images and with more illumination variations than the subset data.

CHAPTER VI

Conclusion

6.1 Summary

Person Re-identification systems is one of the major applications of Computer Vision that has the ability to revolutionize intelligent security and monitoring systems. Many developed countries use Person Re-identification systems for a multitude of purposes, from surveillance of restricted area to monitoring elderly people in smart homes.

In context of Bangladesh, Person Re-identification system has huge area of application. The security of industrial areas, private properties, and restricted areas can be ensured with the implementation of proper Perron Re-identification system. The law enforcement agencies can also utilize the benefits of this system.

In this thesis work, a multi-view Person Re-identification system is proposed. More specifically, for the person matching task two of the best performing models are introduced. A pre-trained model is used for detecting person from the image. A subset dataset is created from a large MARS dataset. An existing model is explored to modify it for performing on the person matching task. Performance of the pre-trained models are also analyzed. A novel architecture is proposed to reduce the time of training and evaluating the system for person matching. A new small size test dataset is created to evaluate the best performing models performance in terms of practical applications. The model's performances are analyzed with recall, precision, f1 score and accuracy metrics.

6.2 Limitations

No system can be perfect. Limitations are the ones that make rooms for future research. The proposed Person Re-identification system also has some limitations. They are as follows:

- Person detector network is a bottleneck for the system. It takes the most processing time.

- This thesis work didn't consider low lighting conditions and different scaled inputs while designing person matching model.
- A subset of the original dataset is used in this process due to limitations of processing facilities, thus the person matching model ought to provides less generalization.
- While designing the person matching model grayscale images were excluded from the considerations as all the training data are RGB images.
- There is no local large enough dataset for this task, so the model is trained with a dataset created outside of subcontinent.
- The Person Re-identification system is incompatible for online applications as the person detector network makes it slow. However, the model can be used in offline applications.

6.3 Future Work

The following tasks can be performed in the future:

- A custom lightweight person detector network can be designed to improve the person detector's performance which in turn can accelerate the while Person Re-Id process.
- A custom local dataset with more lighting condition variations, and scale variations can help the person matching model to adapt with practical application issues.
- Residual Networks can be added to the network to create more deeper network to extract hidden important features for re-identification task.
- Attention mechanism can be used for person matching to get more relevant features from the model.

6.4 Conclusion

A Person Re-identification system is developed. A person detector network is used to detect persons in the images. The several person matching models are analyzed to find the best model for the task. A subset dataset is created. Apart from that a small size test dataset is created. Model's performance is analyzed in terms of recall, precision, f1 score and accuracy metrics. The proposed Person-Re-indentation system is expected to perform well under right circumstances and help to put a positive impact in solving real world problems.

REFERENCES

[1] J. A. T. Olivero, C. M. B. Anilo, J. P. G. Barrios, E. M. Morales, E. J. Gachancipa and C. A. Z. Torre "Comparing state-of-the-art methods of detection and tracking people on security cameras video." *2019 XXII Symp. on Image, Signal Process. and Artif. Vis. (STSIVA)*. IEEE, 2019, pp. 1-5, doi**:** 10.1109/STSIVA.2019.8730271

[2] S. Karanam, Y. Li, and R. J. Radke. "Person re-identification with discriminatively trained viewpoint invariant dictionaries." *Proc. of the IEEE Int. Conf. on Comput. Vis.*. 2015, pp. 4516-4524, doi**:** 10.1109/ICCV.2015.513

[3] J. Jiao, W. Zheng, A. Wu, X. Zhu and S. Gong "Deep low-resolution person re-identification." *Proc. of the AAAI Conf. on Artif. Intell.*. Vol. 32. No. 1. 2018, doi**:** 10.1609/aaai.v32i1.12284

[4] Y. Huang, Z. Zha, X. Fu and W. Zhang "Illumination-invariant person re-identification." *Proc. of the 27th ACM Int. Conf. on multimedia*. 2019, pp. 365-373, doi**:** 10.1145/3343031.3350994

[5] M. S. Sarfraz, A. Schumann, A. Eberle and R. Stiefelhagen "A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking." *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recognition*. 2018, pp. 420-429, doi: 10.1109/CVPR.2018.00051

[6] C. Song, Y Huang, W. Ouyang and L. Wang "Mask-guided contrastive attention model for person re-identification." *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recognition*. 2018, pp. 1179-1188, doi: 10.1109/CVPR.2018.00129

[7] R. Hou, B. Ma, H. Cheng, X. Gu, S. Shan and X. Chen "Vrstc: Occlusion-free video person re-identification." *Proc. of the IEEE/CVF Conf. on Comput. Vis. and Pattern Recognition*. 2019, pp. 7183-7192, arXiv.1907.08427

[8] G. Wang, T. Zhang, J. Cheng, S. Liu, Y. Yang, and Z. Hou "RGB-infrared cross-modality person re-identification via joint pixel and feature alignment." *Proc. of the IEEE/CVF Int. Conf. on Comput. Vis.*. 2019, pp. 3623-3632, arXiv:1910.05839

[9] Y. Huang, Q. Wu, J. Xu, Y. Zhong and Z. Zhang "Clothing status awareness for long-term person re-identification." *Proc. of the IEEE/CVF Int. Conf. on Comput. Vis.*. 2021, pp. 11895-11904, doi: 10.1109/ICCV48922.2021.01168

[10] W. Zheng, S. Gong, and T. Xiang. "Person re-identification by probabilistic relative distance comparison." *CVPR 2011*. IEEE, 2011, pp. 649-656, doi: 10.1109/CVPR.2011.5995598

[11] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth and H. Bischof "Large scale metric learning from equivalence constraints." *2012 IEEE Conf. on Comput. Vis. and Pattern Recognition*, 2012, pp. 2288-2295. doi: 10.1109/CVPR.2012.6247939

[12] F. Xiong, M. Gou, O. Camps and M. Sznaier "Person re-identification using kernel-based metric learning methods." *Comput. Vis.–ECCV 2014: 13th European Conf., Zurich, Switzerland, Proc., Part VII 13*, 2014, pp. 1-16, doi: 10.1007/978-3-319-10584-0_1

[13] M. Farenzena, L. Bazzani, A. Perina, V. Murino and M. Cristani "Person re-identification by symmetry-driven accumulation of local features." *2010 IEEE computer society Conf. on Comput. Vis. and Pattern Recognition*, 2010, pp. 2360-2367, t: 10.1109/CVPR.2010.5539926

[14] Y. Yang, J. Yang, J. Yan, L Liao, D. Yi and S. Z. Li "Salient color names for person re-identification." *Comput. Vis.–ECCV 2014: 13th European Conf., Zurich, Switzerland, Proc., Part I 13*, 2014, pp. 536-551, doi: 10.1007/978-3-319-10590-1_35

[15] T. Matsukawa, T. Okabe, E. Suzuki and Y. Sato "Hierarchical gaussian descriptor for person re-identification." *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recognition*. 2016, pp. 1363-1372, doi: 10.1109/CVPR.2016.152

[16] Y. Wang et al. "Resource aware person re-identification across multiple resolutions." *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recognition*. 2018, pp. 8042-8051, arXiv:1805.08805

[17] K. He, X. Zhang, S. Ren and J. Sun "Deep residual learning for image recognition." *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recognition*. 2016, pp. 770-778, arXiv:1512.03385

[18] Y. Ge, X. Gu, M. Chen, H. Wang and D. Yang "Deep multi-metric learning for person re-identification." *2018 IEEE Int. Conf. on multimedia and expo (ICME)*, IEEE, 2018, pp. 1-6, doi: 10.1109/ICME.2018.8486502

[19] S. Bai, X. Bai, and Q. Tian. "Scalable person re-identification on supervised smoothed manifold." *Proc. of the IEEE Conf. on Comput. Vis. and Pattern Recognition*, 2017, pp. 2530-2539, arXiv:1703.08359v1

[20] D. Chung, K. Tahboub, and E. J. Delp. "A two stream siamese convolutional neural network for person re-identification." *Proc. of the IEEE int. Conf. on Comput. Vis.*. 2017, pp. 1983-1991, doi: 10.1109/ICCV.2017.218

[21] Z. Liu, A. McClung, H. W. F. Yeung, Y. Y. Chung and S. M. Zandavi "Top-down person re-identification with Siamese convolutional neural networks." *2018 Int. Joint Conf. on Neural Netw. (IJCNN)*. IEEE, 2018, pp. 1-8, doi: 10.1109/IJCNN.2018.8489317

[22] X. Chen, et al. "Salience-guided cascaded suppression network for person re-identification." *Proc. of the IEEE/CVF Conf. on Comput. Vis. and Pattern Recognition*, 2020, pp. 3300-3310, doi: 10.1109/CVPR42600.2020.00336

[23] W. Dong, Z. Zhang, C. Song and T. Tan "Instance guided proposal network for person search." *Proc. of the IEEE/CVF Conf. on Comput. Vis. and Pattern Recognition*, 2020, pp. 2585-2594, doi: 10.1109/CVPR42600.2020.00266

[24] N. Wojke, and A. Bewley. "Deep cosine metric learning for person re-identification." *2018 IEEE winter Conf. on Appl. of Comput. Vis. (WACV)*, 2018, pp. 748-756, arXiv:1812.00442

[25] A. Loesch, J. Rabarisoa, and R. Audigier. "End-to-end person search sequentially trained on aggregated dataset." *2019 IEEE Int. Conf. on Image Process. (ICIP)*, 2019. pp. 4574-4578, doi: 10.1109/ICIP.2019.8803643

[26] W. Liu et al. "Ssd: Single shot multibox detector." *Comput. Vis.–ECCV 2016: 14th European Conf., Amsterdam, The Netherlands, Proc., Part I 14*, 2016. pp. 21-37, arXiv:1512.02325

[27] X. Hao, S. Zhao, M. Ye and J. Shen "Cross-modality person re-identification via modality confusion and center aggregation." *Proc. of the IEEE/CVF Int. Conf. on Comput. Vis.*, 2021, pp. 16403-16412, doi: 10.1109/ICCV48922.2021.01609

[28] T. He, X. Shen, J. Huang, Z. Chen and X. Hua "Partial person re-identification with part-part correspondence learning." *Proc. of the IEEE/CVF Conf. on Comput. Vis. and Pattern Recognition*, 2021, pp. 9105-9115, doi: 10.1109/CVPR46437.2021.00899

[29] A. Aich, M. Zheng, S. Karanam, T. Chen, A. K. R. Chowdhury and Z. Wu "Spatio-temporal representation factorization for video-based person re-identification." *Proc. of the IEEE/CVF Int. Conf. on Comput. Vis.*, 2021, pp. 152-162, arXiv:2107.11878

[30] X. Gu, B. Ma, H. Chang, S. Shan, and X. Chen "Temporal knowledge propagation for image-to-video person re-identification." *Proc. of the IEEE/CVF Int. Conf. on Comput. Vis.*, 2019, pp. 9647-9656, arXiv:1908.03885

[31] T. He, X. Jin, X. Shen, J. Huang, Z. Chen and X. Hua "Dense interaction learning for video-based person re-identification." *Proc. of the IEEE/CVF Int. Conf. on Comput. Vis.*, 2021, pp. 1490-1501, doi: 10.1109/ICCV48922.2021.00152

[32] Y. Chen, H. Wang, X. Sun, B. Fan and C. Tang "Deep attention aware feature learning for person re-identification." *Pattern Recognition, Vol.* 126, 2022: 108567, arXiv:2003.00517

[33] S. Sharma, S. Sharma, and A. Athaiya "Activation functions in neural networks." *Proc. Int. J. of Eng. Appl. Sci. and Tech, 2020, Vol. 4, Issue. 12*, pp. 310-316.

[34] B. Xu, M. Wang, T. Chen and M. Li "Empirical evaluation of rectified activations in convolutional network.", *Proc. of Comput. Vis. and Pattern Recognition*, 2015, arXiv:1505.00853

[35] D. P. Kingma, and J. Ba. "Adam: A method for stochastic optimization.", Proc. of 3rd Int. Conf. for Learn. Representation, 2015, arXiv:1412.6980

[36] M. D. Zeiler "Adadelta: an adaptive learning rate method.", 2012, arXiv:1212.5701

[37] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright. "Randomized smoothing for stochastic optimization." *SIAM J. on Optim., Vol.* 22, Issue. 2, 2012, pp. 674-701, doi: 10.1137/110831659

[38] T. Dozat "Incorporating nesterov momentum into adam." Workshop track - ICRL, 2016

[39] S. Ren, K. He, R. Girshick and J, Sun "Faster r-cnn: Towards real-time object detection with region proposal networks." *Adv. in neural Inf. Process. Syst.* 28, 2015, pp. 91-99, arXiv:1506.01497

[40] K. R. Akshatha, A. K. Karunakar, S. B. Shenoy, A. K. Pai, N. H. Nagaraj and S.S. Rohatgi "Human detection in aerial thermal images using faster R-CNN and SSD algorithms." *Electronics* 11.7, 2022, pp. 1151, doi: 10.3390/electronics11071151

[41] H. Zhang, Y. Du, S. Ning, Y. Zhang, S. Yang and C. Du "Pedestrian detection method based on Faster R-CNN." *2017 13th Int. Conf. on Comput. Intell. and Secur. (CIS)*, 2017, pp. 427-430, doi: 10.1109/CIS.2017.00099

[42] C. Li, D. Song, R. Tong and M. Tang "Illumination-aware faster R-CNN for robust multispectral pedestrian detection." *Pattern Recognition, Vol.* 85, 2019, pp. 161-171, arXiv:1803.05347

[43] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei "Imagenet: A large-scale hierarchical image database." *2009 IEEE Conf. on Comput. Vis. and Pattern Recognition*, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848

[44] K. He, X. Zhang, S. Ren, J. Sun "Deep residual learning for image recognition." *Proc. of the IEEE Conf. on Comput. Vis. and pattern recognition*, 2016, pp. 770-778, arXiv:1512.03385

[45] G. Koch, R. Zemel, and R. Salakhutdinov. "Siamese neural networks for one-shot image recognition." *ICML Deep Learn. Workshop*. Vol. 2, No. 1, 2015

[46] L. Zheng et al. "Mars: A video benchmark for large-scale person re-identification." *Comput. Vis.–ECCV 2016: 14th European Conf., Amsterdam, The Netherlands, Proc., Part VI 14*, 2016, pp. 868-884, doi: 10.1007/978-3-319-46466-4_52

[47] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang and Q. Tian "Scalable person re-identification: A benchmark." *Proc. of the IEEE Int. Conf. on Comput. Vis.*, 2015, pp. 1116-1124, doi: 10.1109/ICCV.2015.133