TÉCNICO LISBOA

Deep Representations for Similarity Matching in Person Re-Identification

Francisco Gameiro Proença

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Alexandre José Malheiro Bernardino

Examination Committee

Chairperson: Prof. Joao Fernando Cardoso Silva Sequeira
Supervisor: Prof. Alexandre José Malheiro Bernardino
Member of the Committee: Prof. Jacinto Carlos Marques Peixoto do Nascimento

December 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

First, I would like to thank my supervisor for all the support and guidance through the course of this thesis, for always being available to answer my doubts and for helping me reach the end of this dissertation. His help was essential during this process.

Second, to my colleagues and friends with whom I had the best experiences during my academic path. They were always available for me and supported me no matter what. The best friends that I could have ever asked.

Finally, to my family, and in particularly, to my mother and sister, for helping me pass through all the obstacles both at personal and academic level. And to my father, to whom i dedicate this thesis and all my academic path, since he is the biggest source of inspiration and strength.

Abstract

Person Re-Identification is the task of identifying and locating a person of interest (query) through a set of pictures or videos captured by several (non-overlapping) cameras in a surveillance network. Typically, the query image is compared to a gallery of pictures of persons previously observed in the surveillance space. This task is challenged by the variability of postures, viewpoints, occlusions and illumination conditions in the camera network. Recent progress in deep learning approaches has proposed Siamese architectures and contrastive loss-functions that have proven successful in the Re-Identification Problem. However such approaches are still slow to train and have trouble in achieving real-time functionality. In this way, this dissertation aims at building an efficient Re-Identification system using a lightweight network, such as MobileNet. This Re-Identification system will be composed by siamese architecture to extract features from the query and gallery examples, in combination with a similarity matching network that will be responsible for verifying the similarity of the network inputs. This system will be trained with Contrastive and Triplet Loss in four different datasets. Our results show that this Re-Identification system can be competitive to the state-of-the-art in some datasets, despite having four times fewer network training parameters.

Keywords

Person Re-Identification; MobileNet; Deep Learning; Siamese Networks;

Resumo

A Re-Identificação de pessoas consiste na identificação e localização de uma pessoa de interesse através de um conjunto de imagens ou vídeos capturados por várias câmaras (não sobrepostas) numa rede de vigilância. Normalmente, a imagem da pessoa de interesse é comparada com uma galeria de imagens de pessoas anteriormente observadas nesse espaço de vigilância. A Re-Identificação de pessoas enfrenta vários obstáculos, dos quais se destaca, entre outros, a variabilidade de posturas, pontos de vista, oclusões e diferentes condições de iluminação na rede de câmaras. Progressos recentes em abordagens de aprendizagem profunda propuseram arquitecturas siamesas e funções de perda contrastiva que se revelaram bem sucedidas no problema de Re-Identificação. Contudo, tais abordagens ainda são lentas de treinar e têm demonstrado algumas dificuldades em alcançar a funcionalidade em tempo real. Desta forma, esta dissertação visa construir um sistema de Re-Identificação eficiente utilizando, para tal, uma rede leve, tal como a MobileNet. Este sistema de Re-Identificação será composto por uma arquitectura siamesa para extrair características da pessoa de interesse e exemplos da galeria, em combinação com uma rede correspondente que será responsável pela verificação da similaridade das entradas da rede. Este sistema será treinado com Perda Contrastiva e Perda Tripla, em quatro conjuntos de dados diferentes. Os resultados mostram que este sistema de Re-Identificação pode ser competitivo com o estado da arte em alguns conjuntos de dados, apesar de ter quatro vezes menos parâmetros.

Palavras Chave

Re-Identificação de pessoas; MobileNet; Aprendizagem Profunda; Redes Siamesas

Contents

List of Figures

List of Tables

Acronyms

ACF	Aggregated Channel Features
CCTV	Closed-circuit Television
CNN	Convolutional Neural Network
CMC	Cumulative Match Characteristic
DPM	Deformable Parts Model
FPNN	Filter Pairing Neural Network
KISSME	Keep It Simple and Straightforward Metric
LMNN	Large Margin Nearest Neighbour
LBP	Local binary patterns
LOMO	Local Maximal Occurrence
LSTM	Long Short-Term Memory
ML	Machine Learning
mAP	Mean Average Precision
MuDeep	Multi-Scale Deep Learning Model
PCCA	Pairwise Constrained Component Analysis
PCA	Principal Component Analysis
RGB	Red, Green and Blue colour space
Re-ID	Re-Identification

1

Introduction

Contents

1.1 Motivation

Public safety is an area of great importance in a world where people are feeling more unsafe in public spaces. In order to respond to this need, many Closed-circuit Televisions (CCTVs) systems are being deployed across different places and countries, allowing for the identification and tracking of different people (e.g. a terrorist) and actions (a robbery). The deployment of CCTV cameras have many other tasks of interest for society, besides the positive impact on public safety: tracking lost children in places like markets, theme parks or airports; the opportunity to study the behaviour of people in malls, universities and other buildings. Nowadays, most of those tasks rely on human work that consists mostly in identifying different people of interest and tracking them through the different cameras of the CCTV. However, as typical CCTV systems are composed of a lot of cameras for a human to watch, this process is extremely difficult to handle, and it cannot be performed flawlessly. Thus, person re-identification relying only on human work is limited to small scale scenarios.

Figure 1.1: A Closed-circuit Television map in two floors of Torre Norte - Instituto Superior Técnico.

Person Re-Identification has been studied by a few areas throughout time. In 1961, Alvin Plantinga gave the first person re-identification definition as follows: “To re-identify a particular, then, is to identify it as (numerically) the same particular as one encountered on a previous occasion” [1]. Later, Person Re-Identification was addressed by the computer vision area. The main goal of the computer vision community is to increase the efficiency of the analysis performed by humans, equipping them with powerful tools to help their work. The key objective of person re-identification is to identify a person of interest (query) through a set of gallery pictures previously captured and stored, being able to tell where the person is and where it was.

The computer vision community found a gap that should be addressed by researchers to lead the re-identification system to its full potential. The recent popularity in machine learning, more specifically in deep learning, opened a whole vast area that must be explored and used within re-identification techniques. This will contribute to improve current architectures and techniques that sometimes do not

obtain good results.

It is therefore important to address this problem and to improve the existing techniques so that they can be used in a more automatic way and without as many failures as it happens nowadays.

1.2 Problem Formulation

Person Re-Identification is a computer vision problem that aims at capturing and identifying people across different camera views and angles throughout time in a surveillance network. A standard Re-Identification architecture can be found in Figure 1.2 in which one can find the different tasks that are performed during a Person Re-identification algorithm. Two of those tasks must be emphasised: (i) **Person detection** that corresponds to the upper part of the figure and (ii) **Person Re-Identification** that corresponds to the lower part. These tasks are defined as:

- **Person Detection**: The detection of different people presented in the images. From the raw image frames, each person must be detected and a bounding box should be generated for each one of them in order to successfully retrieve all people presented in all frames. These people should be stored in the gallery to be later accessed in the person re-identification task.
- **Person Re-Identification**: This task consists in matching a photo of the person of interest (query) to a gallery set, where this person might be in. The result of this search will be a ranked list in which the best matches will be at the top of the list and the worst at the bottom.

In this dissertation, I will address only the task of Person Re-Identification. In this way, the Person Detection task is already performed and the datasets used have already detected and cropped each person. The main goal is to develop a Re-Identification system and to study different techniques that can improve existing ones.

A standard Re-Identification (Re-ID) system was already introduced. Nevertheless, in this section, I intend to analyse a Re-Identification system in more detail, specially the part corresponding to person Re-Identification. In Figure 1.3, a person Re-Identification system is presented in detail. As it can be seen, the input of the system will receive an image (single-shot) or a video (multi-shot): a person will be detected, and a bounding box will be generated. In the case of video-based, a tracking algorithm must be implemented. After this first part of detecting a person, a feature representation network will be used in order to obtain a feature vector containing important descriptors of the person of interest. Finally, this feature vector (query) will be compared against all previously stored gallery feature vectors that contain information of previously detected people. A distance-metric approach will be used in order to find out which images in the gallery are closer to the query and a list of the most similar images will be displayed. The problem that must be addressed in this architecture is the quality of the feature extractor, since each

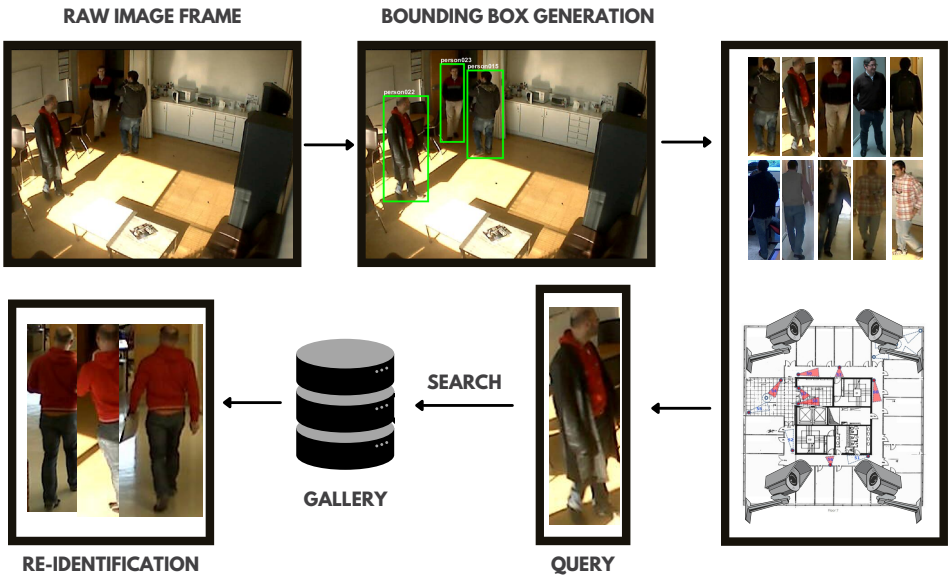


Figure 1.2: An end-to-end Re-Identification System including the task of Person Detection and Person Re-Identification. At the top, the person detection is represented where different persons are captured. At the bottom, the search for a specific query is done in the gallery and the results are returned.

person will be passed through it. It is then crucial to have a feature extractor that is able to distinguish different people just by using the information contained in the bounding boxes.

When defining a Re-Identification problem is important to define in which scenario the problem is happening since there are two possible ones [2]:

- **Close-world** - In the Close-World assumption, one can define that a query (person of interest) is always present in the gallery.
- **Open-world** - In the Open-World assumption, one can define that there is no guarantee of the presence of the query (person of interest) in the gallery. In this case the system must identify if the query is present or not and, if it is not present, add it to the gallery.

The problem can be defined as follows: considering a Re-ID problem in a close-world where G represents the gallery composed by N images, denoted as $G = \{g_i\}_{i=1}^{N}$. These images belong to different M entities $1, 2, ...M$. In this way, given a query (person of interest) image q, its id is determined by the equation

$$id = \underset{i \in 1,2,...N}{\arg\max}\ \phi(q, g_i), \tag{1.1}$$

where id is the id of the query q and ϕ is the function responsible for testing the similarity between the query and gallery images.

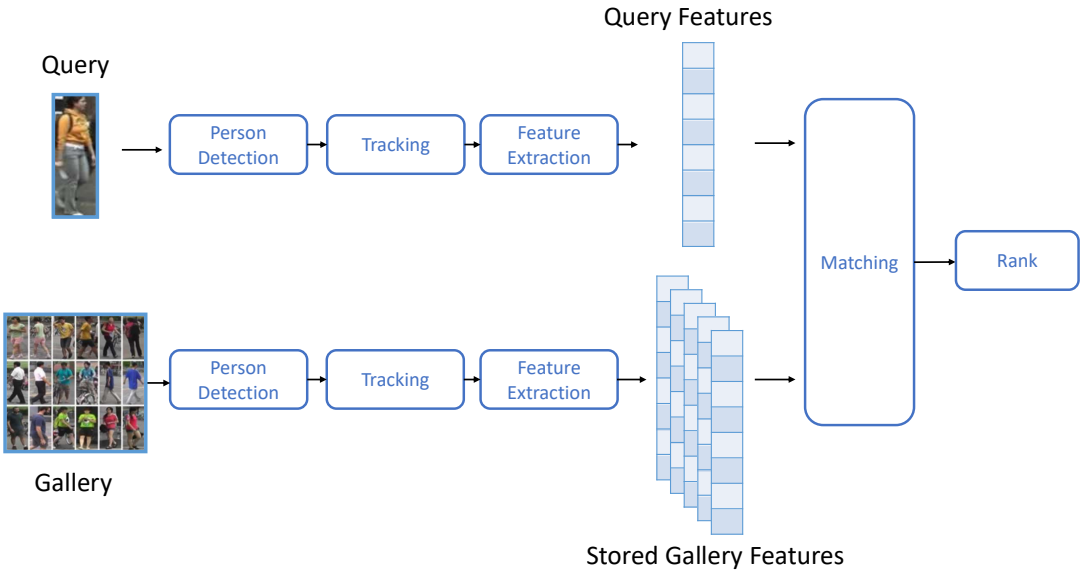


Figure 1.3: Re-Identification system architecture. A Siamese network is represented where at the top, a query is the input and at the bottom an image from the gallery is chosen.

However, if the Re-ID problem is considered to be in the open-world, it is then necessary to add another condition to (1.1). This condition is

$$\phi(q, g_{id}) > h, \tag{1.2}$$

where h is the threshold. In this case, a threshold should be defined in order to test if the match score is above this one and a Re-identification is done. If this does not happen, a new id is defined since the query might not be present in the gallery.

1.3 Challenges

The problem presented in section 1.2 poses several challenges. In fact, the Re-Identification problem continues to be an area of investigation, since the current state of the art systems cannot guarantee an outstanding accuracy when they are deployed into different environments.

As discussed before, the challenges faced by Re-Identification systems can vary. For instance, if a Re-ID system is installed, in an unknown environment, the results obtained may not be as good as desired. Other problems, as identified in Figure 1.4, are related to person image retrieval. In fact, in this

figure, some of the main difficulties presented are: different viewpoints [3] [4], changes in illumination, low-resolution images, occlusions, changing of clothes (eg: Jacket) from one frame to another.

Figure 1.4: Several person images captured from a Person Detection algorithm. Several challenges of Re-identification are presented and can be identified. The first two images represent illumination changes, while the next two images are from the same person but with a different jacket from one frame to another. The last image represents occlusion.

1.4 Objectives

The work of this dissertation aims to develop an efficient re-id pipeline close to state-of-the-art performance but with better computational speed so that it can be used in low cost hardware for real-time decision making. We focus only on the Person Re-Identification part and assuming a closed world scenario. This system will face the challenges already presented in section 1.3 through:

- Development of a deep network that is able to extract good feature representations from different persons;
- Development of a good deep similarity matching network by comparing different ones trained with different losses;
- Study the deployment of a Re-Identification system to a scenario where it was not trained on (generalisation).

1.5 Thesis Outline

This dissertation is organised in five chapters. Chapter 2 presents the Background where crucial theoretical overview is done allowing for a better understanding. Chapter 3 presents the State of the Art where some work done in the areas of interest to this thesis is presented and analysed. In Chapter 4, the methodology approach to the work to be done and presented will be explained and its implementation is carefully explained in chapter 5. Chapter 6 analyses the results obtained during this thesis.

Finally, Chapter 7 concludes with an overall summary of the work carried out during this dissertation as well as some future recommendations to the work that can be done in this area.

2

Background

Contents

During the course of this dissertation different techniques were used to achieve the final result. In order to better understand some of these methods and its future implementation, this chapter will briefly explain them. In this way, this chapter covers Deep Learning, Convolutional Neural Networks and Siamese Networks.

2.1 Deep Learning

Artificial Intelligence is a big area in Computer Science, responsible for giving intelligence to machines that allow them to complete a wide range of tasks in the most diverse fields. Artificial Intelligence has different sub-areas like Natural Language Processing, Planning and Machine Learning.

Machine Learning (ML) consists on the ability of a machine to learn how to perform a specific task through experience and different data interpretation. ML algorithms usually build a model based on the data received and the decision that must be made. The algorithm can accurately perform, through different amounts of data, the task it was designed to do. Nowadays ML is being quite used and one of its main areas is Deep Learning.

Deep Learning is gaining great importance over time since it allows for the extraction and interpretation of features from raw inputs. In order to ensure this, deep learning methods mimic the human brain, and a network with multiple layers can be built. Based on those, a network is as in Figure 2.1. As an example one can think how a deep learning algorithm can distinguish a cat from a dog: (i) The input of the network will receive as an input an image (in this example a dog); (ii) from this image, different neurons will be activated in a specific way triggering different results at the output of the network; (iii) this output will then be able to predict if the input image was from a cat or a dog.

There are different types of networks depending on the problem ought to be solved. However, in the present work, the only network to be discussed and presented is the Convolutional Neural Network (CNN), since it is the most appropriate type of network when images become part of the problem.

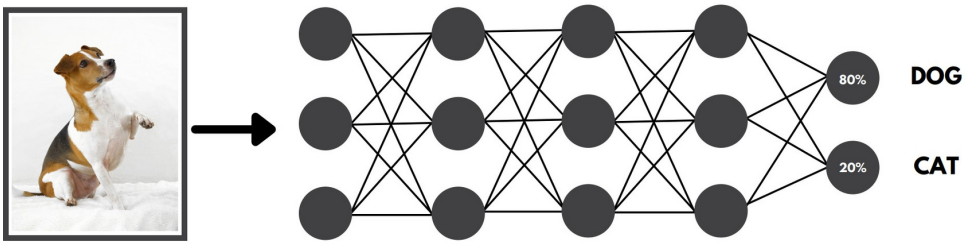


Figure 2.1: Deep Neural Network that is able to recognise if the input image is a dog or a cat.

2.2 Convolutional Neural Network

As explained before, CNNs are one of the Deep Learning networks, which are more recommended to solve computer vision problems. This type of network can interpret the different textures, objects and edges in the different figures by updating the different weights in each neuron that is part of this CNN. In order to do this, the network receives as input a figure - as explained in section 2.2.1 - and is built by a sequence of different types of layers such as Convolution Layer (section 2.2.2), Pooling Layer (section 2.2.3), or Classification Layer (section 2.2.4) that perform some actions on the input image providing an intended output. An example of a CNN can be seen in Figure 2.2.

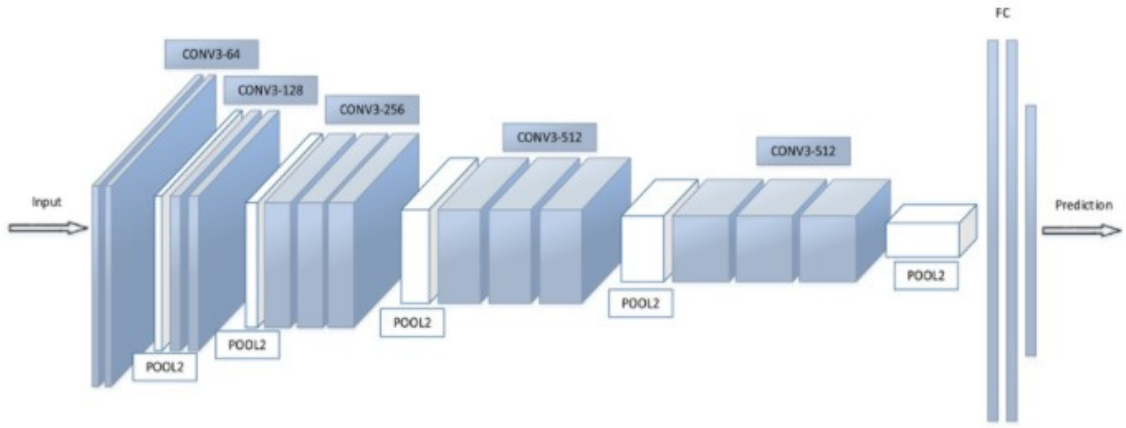


Figure 2.2: Representation of a convolution neural network. This CNN is composed of different Convolution, Pooling layers and has a Classification layer at the end. Adapted from [5].

2.2.1 Input Image

The beginning of a CNN is usually an image and is separated into three channels. In the case of a Red, Green and Blue colour space (RGB) image, the separation is into the Red, Green and Blue channels.

In this way, a CNN has to consider all the input image channels, since they have to be processed in the best possible way in order not to lose any features related to the image.

2.2.2 Convolutional Layer

A convolutional layer is an important part of a CNN, being responsible for extracting the different textures, objects and patterns that are presented in an image. In order to do this, this type of layer uses a matrix of parameters, also known as Kernel (K). As it can be seen in Figure 2.3, the Kernel matrix is smaller than the input image or previous layer, since its objective is to convolve across the input image in its width $w1$ and height $h1$ with a specific stride S and some padding P applied to the image. The Kernel

has the same depth as the input image since each channel has its convolution and the output will be a 2 dimensional activation map which will correspond to the convolution of the kernel through the image. The dimension of the activation map will be $w3 * h3$ and the dimensions can be calculated by (2.1) and (2.2).

$$w3 = \frac{w1 - w2 + 2P}{S} + 1 \tag{2.1}$$

$$h3 = \frac{h1 - h2 + 2P}{S} + 1 \tag{2.2}$$

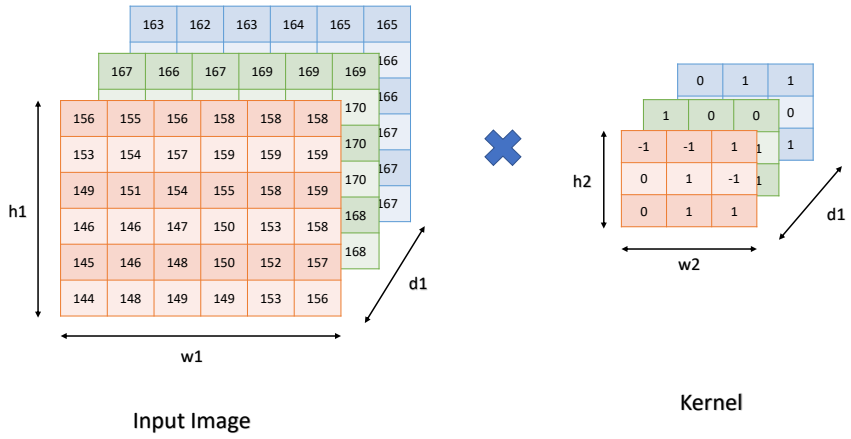


Figure 2.3: Representation of an operation performed by a Convolution layer on an input image. At the left the three channels of the image are represented and, at the right, the Kernel is represented. The kernel will go through the input image and will present the output as a result.

Nevertheless, the first convolutional layers can only extract low level features such as edges or colours. In this way, a CNN must be built by different types of this layer in such a way to allow for an extraction of high-level features. In Figure 2.4, we can observe different levels of features: as more convolutional layers are applied, more accurate features are defined and a classifier can be trained on top of these features to identify different objects in images.

2.2.3 Pooling Layer

The objective of pooling layers is the extraction of representative features, the reduction in computation complexity and the reduction of the number of parameters in consideration, which leads to less probability of overfitting. The pooling operations extract information from different slices of an image and carry them to a new and smaller slice. This process is identified in figure 2.5, where two different pooling functions are being used. There are different types of pooling functions, however, the most popular ones

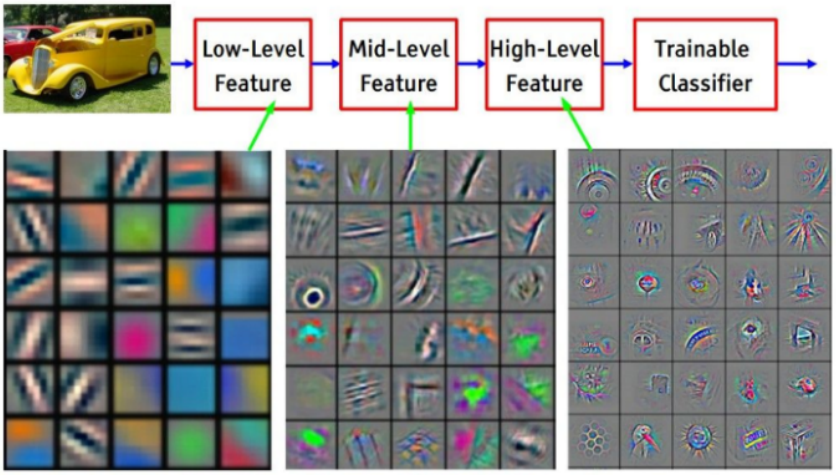


Figure 2.4: Feature representation at different levels of a Convolutional Neural Network. Adapted from [6].

are the Average Pooling and Max Pooling. When performing Average pooling, the new slice will be the average of the previous ones. In this case, a 2×2 filter was used and, from those filters, the average was calculated, resulting in the matrix presented at the right. When performing Max Pooling, the maximum of each 2×2 filter will be considered and only the max number will be in the new slice.

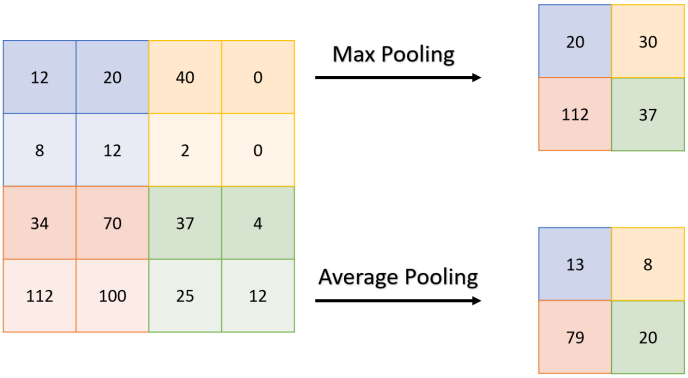


Figure 2.5: Representation of a pooling operation. At the left, a matrix 4×4 is represented and, at the right, the result of 2 different pooling operations are shown. At the top, a max pooling can be seen and, at the bottom, an average pooling is shown.

Both Convolutional Layer and Pooling layer combined are an important part of the CNN. Those are responsible for feature extraction, as more high-level features are desired more layers can be added at a cost of more computational power.

2.2.4 Fully Connected Layers

In those types of layers, all neurons are connected with each other. In this manner, a non-linear representation of the input can be mapped to an output. As it can be seen in figure 2.2, those are the last layers. In order to move from a pooling layer or a convolutional layer, their output is flattened and then fed it to the input of these fully connected layers also known as classification layers. In this way, all information obtained before is gathered up in one vector and the classification is being done from there.

2.2.5 Neural Networks training techniques

2.2.5.A Activation Functions

An Activation function has the job of stimulating the neurons in a certain way to trigger different reactions from them, and, in this way, to interpret the input images and output values at the end of each neuron in the best possible way, resulting in a specific outcome at the end of the network. There are different activation functions that are used according to the problem and the network to be built, since some of them can deal with more complex situations. The most common are Hyperbolic Tangent, ReLU, Leaky ReLU, Softmax and Sigmoid. Softmax layers are important in multi-class problems, obtaining the different probabilities of each class and then predicting the correct class. Sigmoid layer is important in binary classification since it estimates a probability between two classes.

2.2.5.B Loss Functions

With the purpose of understanding how the built network is behaving, a loss function is designed to calculate the difference between the truth labels and the obtained ones. In other words, the loss function main goal is to minimise the output by comparing a predicted output with the real one. In this way, the different weights in each neuron are updated accordingly. Some of the most common loss functions are Cross-Entropy, Weighted Cross-Entropy, Focal Loss and Dice Loss.

2.2.5.C Optimisation Algorithms

In a training phase of a neural network, the main goal is to minimise the loss function, which means, to optimise the solution. Different optimisation algorithms, such as Gradient Descent, Stochastic Gradient Descent (SGD), SGD with momentum, Adam or Adagrad, can be used in order to find the weights that best minimise the loss function.

2.2.5.D Techniques

When discussing deep neural networks, and when training is taking place, some techniques can be applied in order to improve future results. One of them is worth mentioning, Transfer Learning [7], since it significantly reduces the complexity of training a network and it also allows this work to be done without a lot of computational resources. Other technique is Batch Normalisation [8] as it helps the network to train faster and standardises the output of the layers and finally other is dropout for regularisation [9].

Throughout life, humans do not learn everything from scratch but transfer acquired knowledge to new events and tasks. This can also be done in deep learning as it has been discussed for years [10], which gives researchers a special advantage since deep learning models need big amounts of data in order to correctly train and perform well in a wide range of tasks. Also, by using this technique, a lot of time and computational resources can be saved.

In a first phase, transfer learning uses pre-trained models, i.e. state of the art models (e.g. AlexNet [11], ResNet [12], VGGNet [13]), that already proved good results in similar tasks and are already trained on large datasets such as ImageNet [14]. Many pre-trained models are CNNs, showing outstanding results in different computer vision tasks. Since CNN are a big part of this project, transfer learning will be discussed based on those networks.

A CNN, as explained beforehand, has two main parts: a convolutional base and a classifier. In Figure 2.6, these two main parts can be observed in a simplified version. Transfer learning works because those types of models were previously trained on big datasets and have already learnt how to extract features such as edges and shapes and, when applied to a new problem, this experience can be already put into practise. As seen in Figure 2.6, there are several ways to train a model depending on how much data is available to fine-tune it and to what extent this model will be used. This was one of the first contributions to Transfer Learning [10] and show the background of it, that is why it will be explained in detail, to understand the basis of transfer learning. However, a lot of work has already been done in this area. Three options may be considered:

- 1 - Train the entire model when there is a large dataset available and the problem is different from the initial one.
- 2 - Train some layers and freeze others when there is a large dataset and the problem is similar to the initial one or when there is a small dataset and the problem is different from the initial one.
- 3 - Freeze the convolutional base when there is a small dataset available and the problem is similar to the initial one.

When using transfer learning, models are initialised with different known weights and fine tuned depending on the situation. This technique shows that models can achieve good performance in different tasks they are designed for [16].

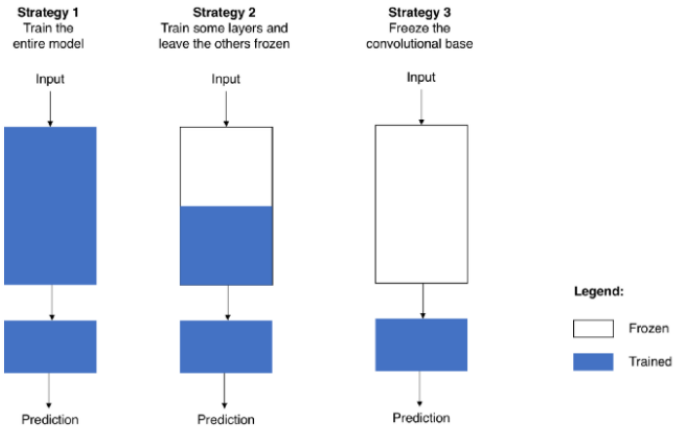


Figure 2.6: Representation of different strategies to Transfer Learning based on different situations. One of the first transfer learning architectures. Adapted from [15].

As for the Batch Normalisation [8], it is an important technique because, during a network training, an input of a layer can be affected by all proceeding layers. Small variation can produce big changes in future layers, so these ones have to continuously be updating its parameters, which leads to an unstable network. To solve this problem, batch normalisation can be used which consists in standardising the input of layers to reduce co-variance this is done in batches. In this way, the network can be faster during training and parameter initialisation is better and higher learning rates can be achieved.

Finally, as for droupout [9] it is an important technique to use. As it randomly select different neurons to be ignored during training since their weights are not updated on the backward pass. This techniques allow the network to learn the weights separately, instead of all at the same time, which has an impact when one wants to solve the overfitting issue.

2.2.6 Deep Neural Networks for classification

Deep neural networks can achieve outstanding results in different computer vision tasks such as classification. Since 2012, when the first deep neural network, AlexNet [11] was introduced in the literature, remarkable results were achieved in classification problems compared to other methods at the time. Since then, more networks have been developed and tested and nowadays a lot of different ones are at use with excellent results [17]. Networks that are worth mentioning are ResNet [12], GoogleNet [18], VGGNet [13], MobileNetV1 [19]. All of those are a key part of state of the art solutions in this area allowing for the creation of new and better networks. Figure 2.7 presents different networks, distributed spatially, and according to their accuracy and operations. When comparing those networks to new ones, one may conclude that their accuracy is lower. In fact, some of the best models are the ones with more parameters and operations, requiring more processing power that normal computers and systems do not

have. In this way, a more lightweight model was considered for this thesis, in particular the MobileNet, which presents remarkable results in the ratio Accuracy/Operations. Next section will be focused on this model.

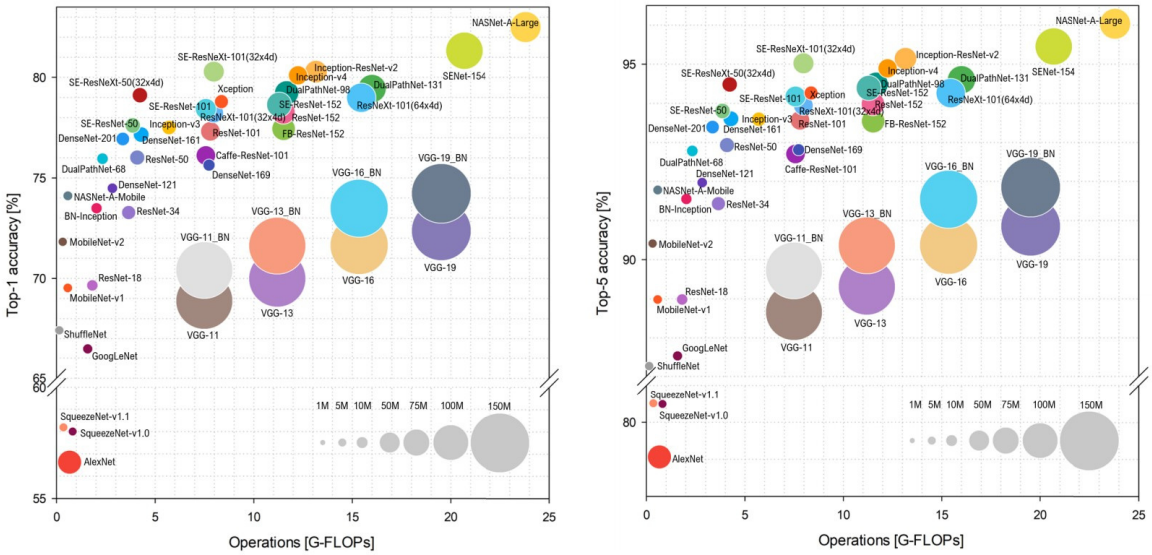


Figure 2.7: Comparison between different types of deep neural networks. Adapted from [17].

2.2.7 MobileNet

The chosen architecture for this work was MobileNet [19] [20] since this type of CNN presents a better accuracy using fewer resources, in contrast with other networks that have good accuracy at the cost of operations and computational resources. This type of network is designed to perform well on mobile devices. This network architecture started with MobileNetV1 [19], and an improved version was created based on it - the MobileNetV2 [20]. This new one has its simplicity and an improvement in the accuracy in image classification and detection.

These type of networks replaces standard convolutions with two different ones: Depth-wise Separable Convolutions and Point-wise Convolution. Figure 2.8 presents those two. At the right side, the Depth-wise convolution is represented - this layer consists in applying a single convolution filter per input channel - while at the left side, one can observe the Point-wise Convolution, where a 1×1 convolution is applied with the purpose of obtaining only one layer with the information of the different channels and of building new features.

A Standard convolution takes $h_i \times w_i \times d_i$ where h, w and d are height, weight and depth respectively, and applies a Kernel $K \in \mathbb{R}^{k \times k \times d_i \times d_j}$ to obtain $h_i \times w_i \times d_j$. This convolution has the cost of $h_i \cdot w_i \cdot d_i \cdot d_j \cdot k \cdot k$. In contrast, the depth-wise convolutions have a cost of $h_i \cdot w_i \cdot d_i(k^2 + d_j)$. In the case of

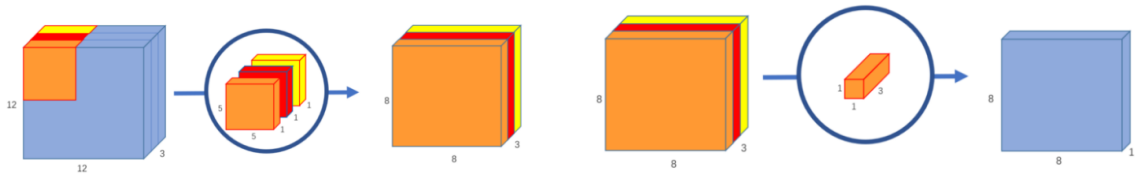


Figure 2.8: Depth-wise convolution, uses 3 kernels to transform a $12 \times 12 \times 3$ image to a $8 \times 8 \times 3$ image, Point-wise convolution, transforms an image of 3 channels to an image of 1 channel. Adapted from [21].

MobileNet, the value of k is 3 (3×3 convolutions) so the computational cost is 8 to 9 times smaller than a standard convolution.

One of the main advantages of MobileNetV2 when compared to the previous version, MobileNetV1, is the addition of an inverted residual layer. This was added since feature maps can be encoded in low-dimensional subspace and non-linear activation results in the loss of information. Based on these principles, this new layer, called Bottleneck, was built and is the base of MobileNet as will be seen further.

Table 2.1 shows the bottleneck block, that is an essential part of the network. First, there is a point-wise convolution whose main goal is to expand the low-dimensional input feature map to a higher-dimensional space that will be able to receive non-linear activation functions. In this case, the ReLU6 (modification to the original ReLU where the activation as a maximum of 6) is applied. The expansion factor is referred to as t in the paper [20]. Second, a depth-wise convolution is applied using a kernel K with 3×3, followed by an activation using ReLU6. Finally, the spatially filtered featured map suffers, once more, a point-wise convolution which causes a projection to low-dimensional subspace. This projection results in the loss of information, so it is important that the final activation is linear. When feature maps have the same dimension (initial and final) a residual connection is added between these layers enabling gradient flow in back-propagation.

Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

Table 2.1: Bottleneck residual block from MobileNetV2. Adapted from [20].

Table 2.2 presents the structure of both MobileNetV1 and MobileNetV2 architecture. They have a similar architecture, although the version 2 has several bottlenecks layers in relation to the first version.

To conclude, batch normalisation and dropout were used during training for both networks. The

number of parameters varies between 1.7M and 6.9M depending on the size of the input image that can vary between 96 to 256.

MobileNetV1

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

MobileNetV2

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Table 2.2: Representation of the MobileNet structure. At the left the structure of MobileNetV1 can be seen and at the right MobileNetV2 can be observed. Adapted from [19] and [20].

2.3 Siamese Networks

A Siamese Neural Network is composed of two or more equal networks as it is represented in Figure 2.9. In this way the same input produces the same output. The goal of the network is to produce feature vectors that are similar if the images are from the same person, and different otherwise. To compare the feature vectors an Euclidean distance can be performed based on this equation,

$$d(x, y) = \left(\sum_{k=1}^{n} (y_k - x_k)^2 \right)^{\frac{1}{2}}, \tag{2.3}$$

where x and y are the vectors, k their component index and n their length. In this way, $d(x, y)$ will have the distance between both vectors which will be big if the images are very different or small if are very similar. If instead of using the Euclidean metric one wants to learn a metric more suited to the dataset, then losses like Contrastive Loss explained in 2.3.1 and Triplet Loss explained in 2.3.2 can be an option.

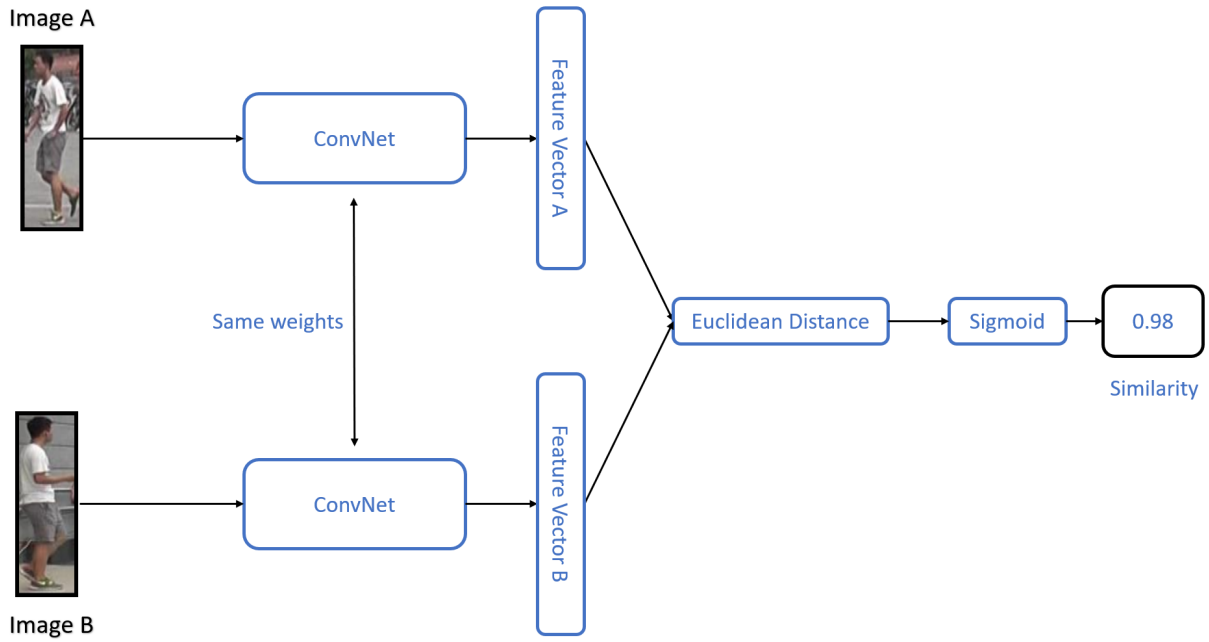


Figure 2.9: Siamese Network example. The two Convolutional Neural Network (ConvNet) are the same and have the same weight, the output will be the similarity between the input images.

2.3.1 Contrastive Loss

Contrastive Loss will receive two feature vectors as the input data. It trains the network aiming at obtaining representations of the same class closer together (positive samples) while creating a distance between different classes (negative samples). So, this loss is small when both conditions are met. In order to distinguish between vectors, a distance metric can be used. Both Euclidean and cosine distance can be used by contrastive loss, but in this case Euclidean was the one used. The goal is not to classify a pair of images, but to train the network to be able to distinguish them. The equation for this loss can be formulated as:

$$\mathcal{L} = Y * D_w^2 + (1 - Y) * \max(m - D_w, 0)^2 \tag{2.4}$$

where Y is the truth value (1 if it is the same class; 0 if not), D_w is the Euclidean distance between feature vectors, m is the parameter which defines the distance different images must be pushed away. The $\max$ function chooses the largest number among 0 and the m minus D_w the distance.

In Figure 2.10 a demonstration of this contrastive loss can be seen. At the left part, it can be seen what happens when $Y = 1$, this leads to minimising the distance (d) between the two images identified. On the other hand if $Y = 0$ the loss function will be minimised by pushing images further apart, leading the distance between them to be greater than 1.

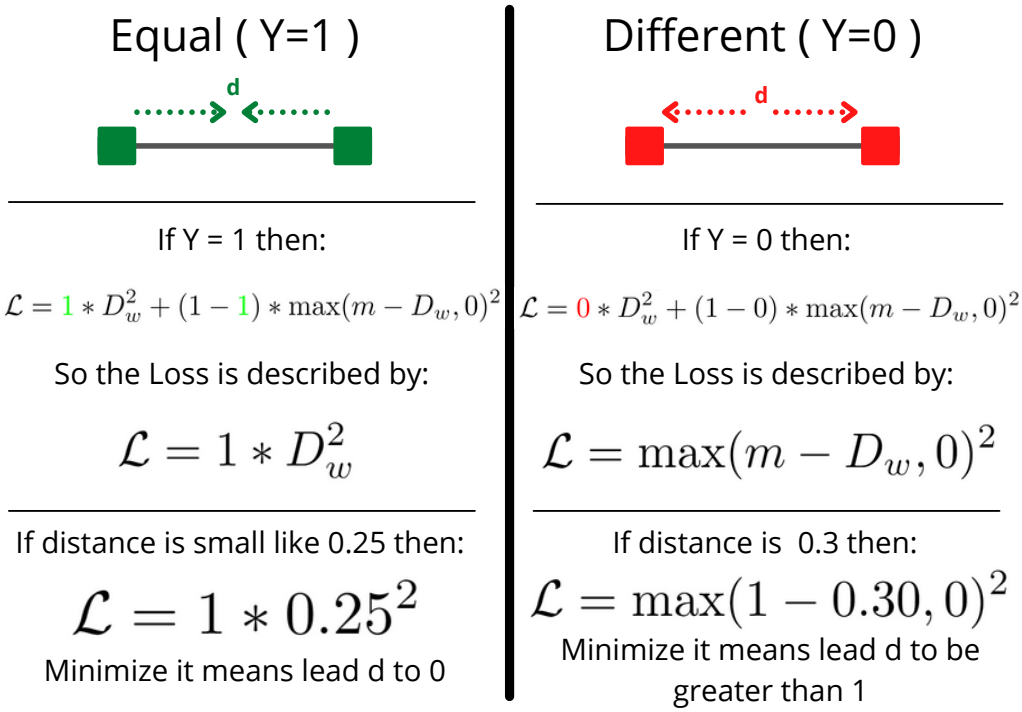


Figure 2.10: Contrastive Loss explanation training in a Siamese Network.

2.3.2 Triplet Loss

Triplet Loss receives as input data three feature vectors: an Anchor vector; a Positive vector that belongs to the same class as the Anchor vector; and a Negative vector that belongs to a different class than the Anchor vector. This Loss will have the objective of bringing the Anchor and Positive vector close together while pushing away the Anchor and Negative vectors. In order to do this the loss is formulated as:

$$\mathcal{L}\left(A, P, N\right) = \max\left(\| \mathrm{f}\left(A\right) - \mathrm{f}\left(P\right) \|^2 - \| \mathrm{f}\left(A\right) - \mathrm{f}\left(N\right) \|^2 + \alpha, 0\right), \tag{2.5}$$

where $f(.)$ is the function to obtain the feature vectors, A, P and N the Anchor, Positive and Negative vectors respectively, α is the parameter which defines the distance different images must be pushed away. The $\max$ function chooses the largest number among two. Simplifying the equation (2.5) it is obtained,

$$\mathcal{L} = \max\left({d_{AP}}^2 - {d_{AN}}^2 + \alpha, 0\right) \tag{2.6}$$

where d_{AP} is the distance between Anchor and Positive Vectors and d_{AN} is the distance between Anchor and Negative Vectors. In Figure 2.11 a brief graphical representation of triplet loss being applied to three vectors can be seen as well as the Loss function minimising explanation.

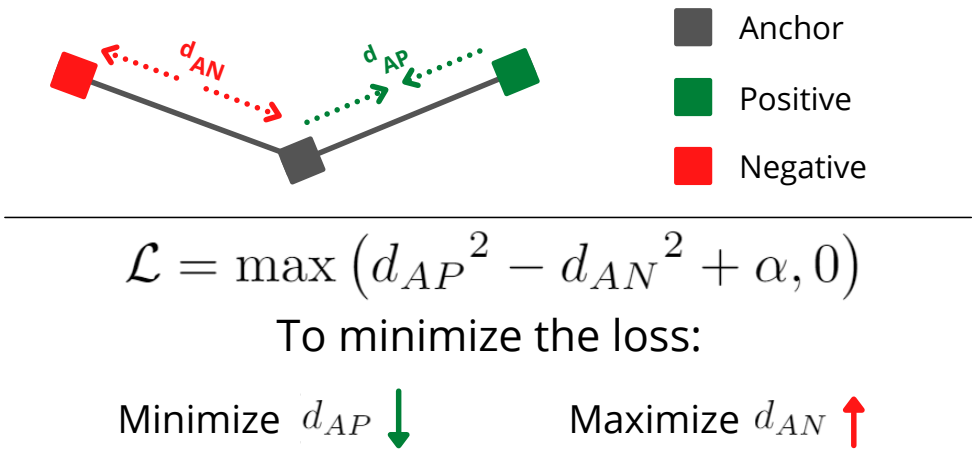


Figure 2.11: Triplet Loss training explanation in a Siamese Network.

When one wants to train a network with triplet loss, the hardness of triplets is important. Contrary to what happens for Contrastive Loss where it is enough to make positive and negative pairs for all classes. The triplets cannot be made in any way. But the loss values obtained must be taken into account. In this way the triplet can be separated as:

- **Easy Triplets** - Loss is equal to 0 because $d_{AP} + \alpha < d_{AN}$.
- **Semi-Hard Triplets** - Positive distance is smaller than the negative but the loss is greater than 0: $d_{AP} < d_{AN} < \alpha + d_{AP}$.
- **Hard Triplets** - Negative distance is bigger than the positive: $d_{AN} < d_{AP}$.

For example, considering an easy triplet, if the $d_{AP} = 0.1$, $d_{AN} = 2.1$ and $\alpha = 0.5$ then the loss is,

$$\mathcal{L} = \max\left(0.1^2 - 2.1^2 + 0.5, 0\right) = \max\left(0.01 - 4.41 + 0.5, 0\right) = \max\left(-3.9, 0\right) = 0.0, \tag{2.7}$$

the loss is 0 so there is no need for any action. Otherwise, if a semi-hard triple is considered, where $d_{AP} = 0.2$, $d_{AN} = 0.4$ and $\alpha = 0.5$ then the loss is

$$\mathcal{L} = \max\left(0.2^2 - 0.4^2 + 0.5, 0\right) = \max\left(0.04 - 0.16 + 0.5, 0\right) = \max\left(0.38, 0\right) = 0.38, \tag{2.8}$$

in this case the loss is positive. The positive loss is also seen in the hard triple, for example if $d_{AP} = 0.8$, $d_{AN} = 0.4$ and $\alpha = 0.5$ then:

$$\mathcal{L} = \max\left(0.8^2 - 0.4^2 + 0.5, 0\right) = \max\left(0.64 - 0.16 + 0.5, 0\right) = \max\left(0.98, 0\right) = 0.98. \tag{2.9}$$

With the use of calculations presented, the different scenarios that could happen when making a triplet are shown.

3

State of the Art

Contents

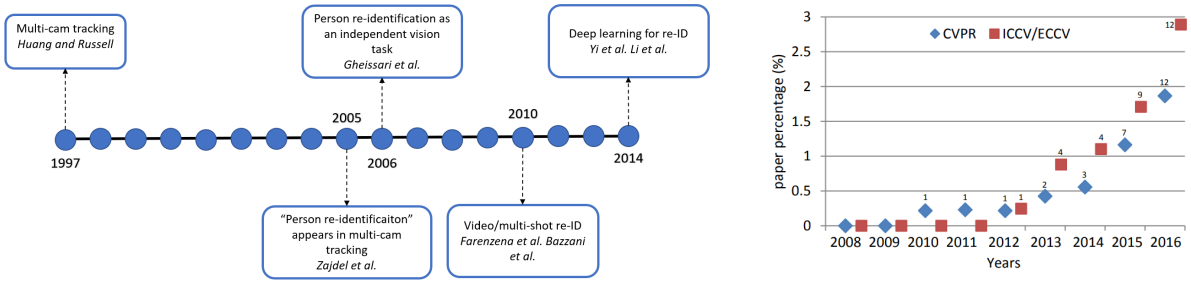
This chapter aims to review the state of the art work developed in the Person Re-Identification. In this way, this chapter is divided in three sections. In 3.1, it is presented the person re-identification evolution through time; in 3.2, it is analysed state of the art hand-craft and deep re-id systems. Finally, in 3.3, an overview of public used datasets for Re-Id is presented.

3.1 Person Re-Identification evolution

Person Re-Identification is a field that has been growing over the years and Figure 3.1(a) clearly illustrates the path that has been followed and that will be then explained. Also, Figure 3.1(b) shows that the number of papers published in this field has been growing at an exponential rate since 2008 [22]. This path started, in 1997, with Huang and Russel discussing multi-camera tracking [23] that, in a certain way, englobes the Re-ID field as well. In their paper, they proposed a Bayesian formulation to predict the appearance of objects through different cameras given a prior position in other camera views. This model includes features such as colour, object length, height and width, velocity, and time of the observation.

Even though multi-camera started in 1997, it was only in 2005 that the Re-ID was formally introduced in the literature by W.Zajdel *et al.* [24]. They tried to re-identify people, given an unique code for each person, by exploring a dynamic Bayesian network that related the labels with the features of each person. But it was only one year later, in 2006, that Re-ID was considered an independent area of study, being classified as an independent computer vision problem when Gheissari *et al.* [25] presented a spatial-temporal segmentation method to find different matches, based on colour and edges. This paper was considered an imaged-based approach since different frames were taken from the actual videos, resulting in different images. Following image-based Re-Id, video-based Re-ID was born in 2010 according to L.Bazzani *et al.* [26]. Here the minimum distance between different bounding boxes is calculated to re-identify a certain person, based on colour and features. Also, this paper shows that having multiple shots of a person can improve its re-identification.

Finally, in 2014, deep learning was introduced in Re-ID after presenting good results in other areas like image classification. In [27] [28], a Siamese Neural Network was firstly introduced to determine if a pair of input images was the same person (id) or not. Since then, deep learning approaches have been heavily explored and good results were obtained in different datasets by using this method and some variations.



(a) History path of Re-Id

(b) Papers published of Re-Id

Figure 3.1: Evolution of Re-Identification over the years displayed as chronological form (a) and in the number of published papers (b). Adapted from [22].

3.2 Person Re-Identification systems

3.2.1 Feature Extraction

As discussed in section 1.2, a Re-Identification system has several parts, all of them were addressed separately, by different researchers. The next two sections will focus on the Feature Extractor and Matching Network as well as their evolution over time. In this section the Feature Extractor for Re-Identification will be introduced and explained.

In person feature extraction, the most used features are colour and texture, even though the latter is less used. In [29], *Gray et al.* try to address the problem of the viewpoint of the camera as well as the pose of the person, so their algorithm uses colour channels (RGB, HSV and YCbCr), texture histograms and several horizontal stripes that define a person to allow the combination of simple features into a similarity function. That method is called Ensemble of Localised Feature (ELF). Different works also explore the combination of different colour channels and textures, such as [30], in which different colour channels were used with the combination of Local binary patterns (LBP) [31].

In [32] and past work done by the same author, the features of an image are extracted from each 10×10 patch taking into consideration the LAB colour space as well as the Scale-invariant feature transform (SIFT) descriptor. In [33], a person is separated into different parts (head, torso and legs) in which an HSV histogram is applied. In [34] the Local Maximal Occurrence (LOMO) is proposed, combining both HSV colour histograms and scale-invariant LBP. Both feature are currently the most reliable for person re-identification.

Despite the good results presented by low level features, some work is being done in other areas, using attribute-based features. Those contribute to better specify some of the person characteristics often attributed by the human eye like gender or height, and are trained based on low level features. Throughout the years, more and more works include those procedures [35] [36]. One of the approaches

[37] consists on transfer learning after learning different attributes in a photography dataset into a re-id dataset. Several other papers have approached this technique.

3.2.2 Matching

In section 3.2.1 several ways of obtaining feature representations were discussed. However, all types of feature extractors also need a Distance Metric in order to evaluate if there is a match between the query and the people in the gallery. The goal of metric learning is to bring vectors of the same class closer together and vectors of different classes further apart. There are some methods that will be addressed, but the most popular one is the Mahalanobis distance function that is formulated as in (3.1), where x_i and x_j are the two vectors being compared and S is a positive semidefinite matrix.

$$d(x_i, x_j)^2 = (x_i - x_j)^T S^{-1} (x_i - x_j), \tag{3.1}$$

The Euclidean distance is a particular case of (3.1) when S is the identity matrix. This popular equation led to the Keep It Simple and Straightforward Metric (KISSME) [38] method where the difference between the vectors is calculated and it is assumed to be a Gaussian distribution with zero mean. The Principal Component Analysis (PCA) is also applied in order to eliminate dimension correlations. In [39] the Large Margin Nearest Neighbour (LMNN) is proposed, where (3.1) is also used with the aim of designing a perimeter corresponding to matching pairs and to associate a penalty to those pairs who are not correct. Nevertheless, this method presents overfit and this overfitting problem is addressed in [40] where the Information-theoretic Metric Learning (ITML) is presented.

In [30], the Pairwise Constrained Component Analysis (PCCA) is proposed allowing a linear mapping function to be able to work directly on high-dimensional data, while ITML and KISSME should be preceded by a step of dimension reduction.

Finally, one of the methods presented in the literature dismisses metric learning, using techniques such as Support Vector Machine [41] and Adaboost [29] instead, in order to correctly separate identities.

3.2.3 Deep Learning System

As referred in section 3.1, deep learning models have gained a lot of popularity in Re-ID since the creation of AlexNet [11]. In fact, some work has been developed in this area due to AlexNet. Deep Learning was successfully introduced in Re-ID by [27] [28] and, since then, the number of publications in Re-ID using deep learning methods has been growing a lot.

In Re-ID there are two common techniques that are applied to solve this problem. The first method uses a CNN model for a classification purpose. Typically this CNN is a state-of-the-art model that is already pre trained on ImageNet [14] and is fine tuned for a specific Re-ID dataset where each identity

represents a different class. In this way, the model will be able to classify different ids at test time. The second common method is a Siamese Network. This network consists of two equal CNNs that share the same weights. The main goal is to put two images as an input and, in turn, obtain the similarity between them. Different losses like contrastive and triplet loss can be implemented to bring the same class closer together and different classes further apart.

The first work that proposes to learn similarity metrics from the image pixels was presented by D Yi *et al.* [28]. This method uses a Siamese network to learn colour feature, texture feature and metric at the same time. As a Siamese network it has two equal networks with the same weight that are joined by a cosine layer (cosine distance). In [27], a Filter Pairing Neural Network (FPNN) is introduced with the goal of addressing some challenging problems of re-id like misalignment or occlusion. The work presented is similar to the one presented in [28], even if this one has a patch matching layer that can handle the problems mentioned before. In [42] a re-id system (Personnet) was designed with the goal of learning high level features and a similarity metric. This network receives two images as shown in figure 3.2. Several operations are performed on these images to extract the crucial information from each one. Subsequently the resulting features are concatenated into a vector. Their similarity is calculated from this obtained vector and the result if they belong to the same person or not is displayed.

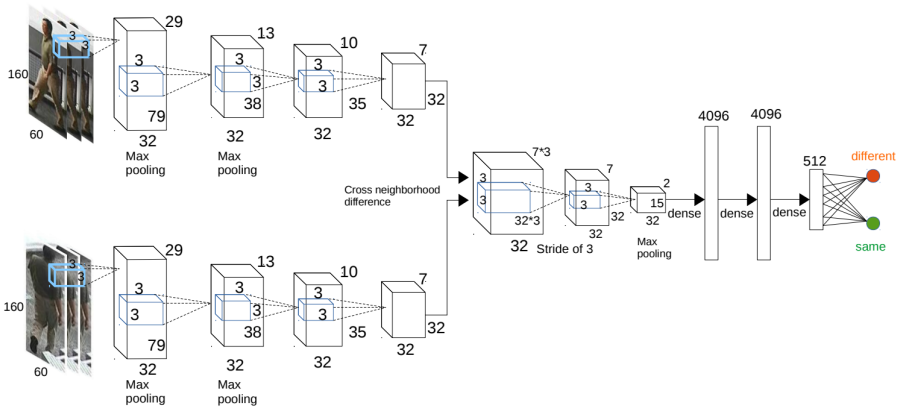
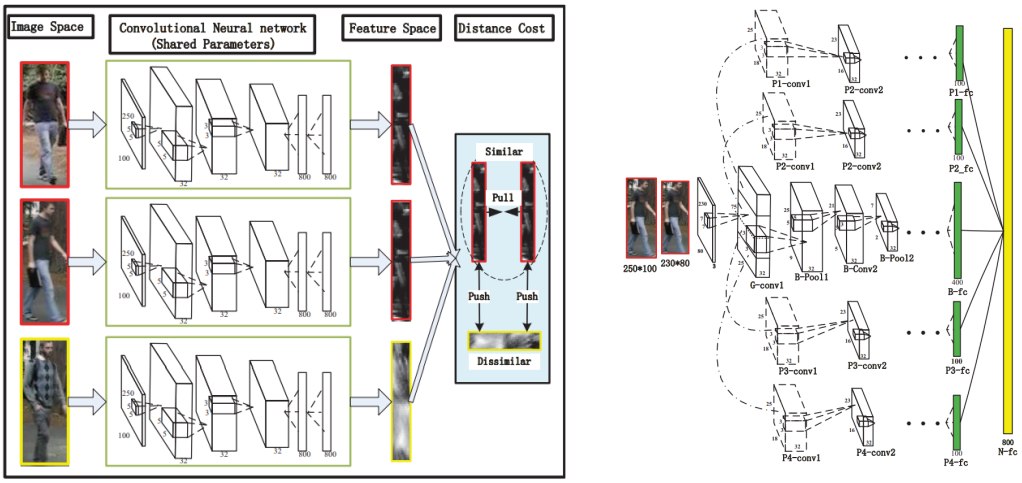


Figure 3.2: Siamese Re-Identification structure. Receiving two images as an input that are going to be passed through a CNN with similar weights and conclude if they are different or the same. Adapted from [42].

In [43], a Long Short-Term Memory (LSTM) is used for the first time in Re-Identification. In that paper, the network starts by dividing the input images into different parts to extract different local features from each one. The big advantage of using LSTMs in relation to other networks is that LSTMs are able to learn from previous images, this is, they will propagate contextual information in each cell, giving a type of memory to each one which will equip the network with better tools to analyse different images. Different works were published trying to improve this Siamese network adding some tricks to the network itself, for instance, to improve the network by adding a gating function after each convolutional layer [44],

or by adding an attention base model to retrieve better local features [45]. There are several papers that use Siamese networks and therefore image pairs at the network input. Contrary to the tendency to use pairs of images, Cheng *et al.*, in [46], present the triplet loss training where, instead of having a network similar to the one presented in Figure 3.2 , with only two input images, the network is similar to the one presented in Figure 3.3(a). The aim of that architecture is to push features of the same class closer together, while moving away features of different classes. In that paper, the architecture is able to acquire both local and global features by using the system presented in figure 3.3(b) - a multi-channel pipeline - that is able to evaluate and analyse both the different parts of the body and the body as a whole, which will make the final feature vector as presented.



(a) Triplet Loss architecture

(b) Multi-Channel pipeline

Figure 3.3: Triplet loss architecture represented in Figure 3.3(a) where the multi-channel technique is applied as represented in Figure 3.3(b). Adapted from [46]

One of the best performing systems using the Siamese Network is known as Multi-Scale Deep Learning Model (MuDeep) [47] which presents outstanding results in different benchmarks datasets and uses the ResNet50 as a backbone network to extract different features. However, on top of this network, some changes were made to improve its functioning. For instance, the introduction of a multi-scale stream layer that is able to identify some discriminant descriptor in images by analysing each scale independently. Or, in addition, the creation of a Leader-Based attention learning layer in order to give more attention to important descriptors rather than background ones, that are useless when one wants to distinguish different people. Considering that Re-ID can be a classification and a verification problem [47], it combines both of these losses in order to train the network and uses both global and local features to classify each person. In [48], both common methods are employed to train the system and some good practises, to be applied when building a Re-ID system, are presented. The system starts by

being trained for the classification task and, after that, it is trained with triplet loss to re-identify people. An example of good practise, mentioned in the literature, is the use of Data Augmentation on training data.

In [49], a similar method is followed. The author, firstly, trains the MobileNetV1 for a classification task and after that takes the classification head in order to obtain the feature vector with size of 1024×1. It then builds a similarity matching network that compares feature vectors and delivers the probability of being the same person. Figure 3.4 presents the re-identification system just discussed.

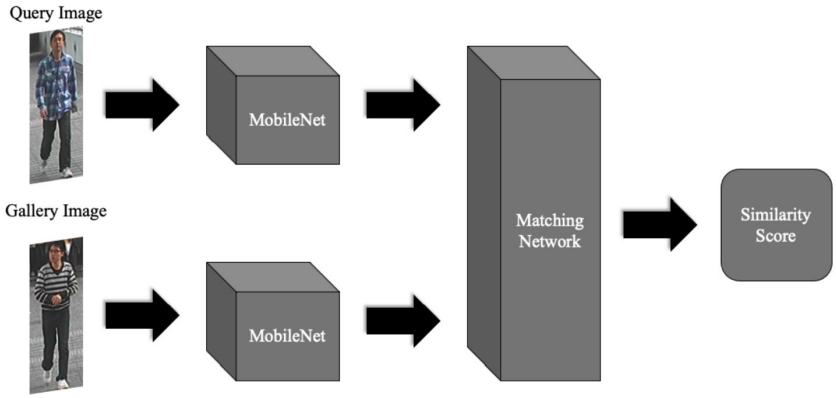


Figure 3.4: Re-Identification system that receives two images of people. It can check whether they belong to the same person or not. Adapted from [49].

As just debated, there are some good practises when designing a Re-ID system that can leverage it to obtain better results and outperform state-of-the-art mechanisms. These good practises are not directly related to the network itself, but with pre-processing data and with ranking results. In [50], a re-ranking method was presented in order to obtain better results , i.e., to obtain a better ranking list than the ones obtained by the system itself. To do so, there are different techniques addressed by several papers. In [50] , the authors make the computations presented in Figure 3.5, in which, for each image, they compute the k nearest neighbours (10 in this case). From that, they inspect the nearest neighbours and verify if the probe is one of them. Otherwise, they consider the hypotheses of the rank-k result being a false match and move on to re-rank the return list. In this way, they show some improvements in ranking results.

In [51], Zhong *et al.* discussed the importance of data augmentation and, more specifically, debated a new method of data augmentation - Random Erasing. Random Erasing consists in randomly selecting a rectangle in a figure and erasing those pixels. As mentioned, this is a random event whose main goal is to prevent overfitting when training the model and making it more robust to occlusions.

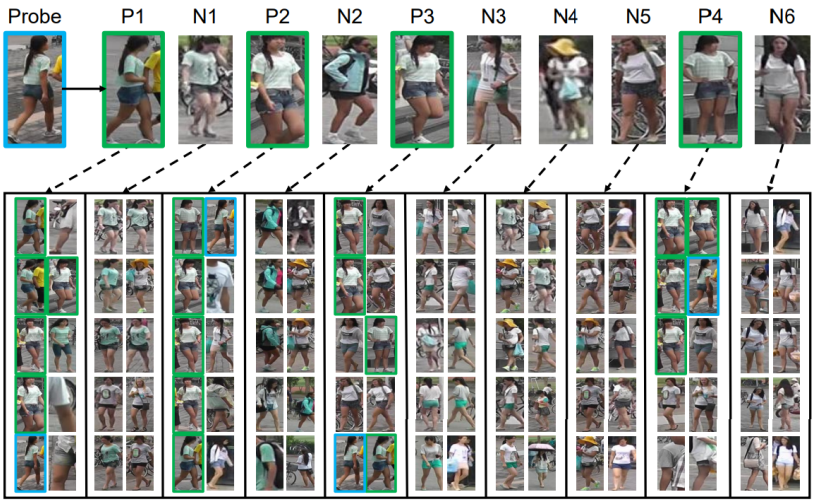


Figure 3.5: Re-Ranking technique. The k nearest neighbours are calculated for each image and a search for the query in the neighbours is done. Results are updated accordingly. Adapted from [50].

3.3 Datasets

In order to train a Re-Identification system and then evaluate it, there are different public available datasets. Those are an important part allowing the system to understand which features are more important and relevant to extract, while evaluating the system performance in the Re-ID task. In this way, it is important for these datasets to have some of the challenges presented in section 1.3 such as illumination changes and scale variations so that models can learn how to surpass them.

In Table 3.1, the most used datasets for the close-world Re-ID task, more specifically for deep learning, are presented. This table is divided into two sections: (i) Single-Shots, that includes 15 image datasets and (ii) Multi-Shot that includes 8 video datasets. For each one, different parameters are described, such as Time, #ID, #Images, Image Size and Evaluation.

As it can be seen throughout the years, there is an increase in the number of datasets. Taking a closer look at the numbers in the table, specifically to #Images and #Cameras, the number of images has increased because a deep learning system is better once there is more available data to train it. The increase in the number of cameras is due to simulating a real CCTV system. It is thus important to have a more practical dataset that can approximate to the real world. In what concerns image size, this is an important fact to take into consideration when building a network since some images must be resized to the input size of the network. Other important parameters to consider are the Labels. In older datasets, all images were labelled by hand and their corresponding bounding boxes were generated manually as well. However, with the growth of datasets, and in order to simulate real world detection systems, these bounding boxes and labels started to be done automatically with the use of methods like Deformable Parts Model (DPM) [71] and Aggregated Channel Features (ACF) [72]. It is also important to mention

Single-Shot Datasets							
Dataset	Time	#ID	#Cameras	#Images	Image size	Label	Evaluation
VIPeR [29]	2007	632	2	1264	fixed	hand	CMC
iLIDS [52]	2009	119	2	476	vary	hand	CMC
GRID [53]	2009	250	8	1275	vary	hand	CMC
CAVIAR [54]	2011	72	2	610	vary	hand	CMC
PRID2011 [55]	2011	200	2	1134	fixed	hand	CMC
WARD [56]	2012	70	3	4786	vary	hand	CMC
CUHK01 [57]	2012	971	2	3884	fixed	hand	CMC
CUHK02 [58]	2013	1816	10 (5 pairs)	7264	fixed	hand	CMC
CUHK03 [27]	2014	1467	2	13164	vary	hand/auto	CMC
RAiD [33]	2014	43	4	1264	vary	hand	CMC
PRID 450S [59]	2014	450	2	900	vary	hand	CMC
Market-1501 [60]	2015	1501	6	32668	fixed	hand/auto	CMC/mAP
DukeMTMC [61]	2017	1404	8	36411	fixed	hand/auto	CMC/mAP
Airport [62]	2017	9651	6	39902	fixed	auto	CMC/mAP
MSMT17 [63]	2018	4101	15	126441	vary	auto	CMC/mAP
Multi-Shot Datasets							
Dataset	Time	#ID	#Cameras	#Images	Image size	Label	Evaluation
PRID-2011 [55]	2011	200	2	400(40k)	fixed	hand	CMC
iLIDS-VID [64]	2014	300	2	600(44k)	vary	hand	CMC
HDA+ [65]	2014	64	13	16844	vary	hand	CMC
MARS [66]	2016	1261	6	20715(1M)	fixed	auto	CMC/mAP
Duke-Video [67]	2018	1812	8	4832(-)	fixed	auto	CMC/mAP
Duke-Tracklet [68]	2018	1788	8	12647(-)	fixed	auto	CMC/mAP
LPW [69]	2018	2731	4	7694(590K)	fixed	auto	CMC/mAP
LS-VID [70]	2019	3772	15	14943(3M)	fixed	auto	CMC/mAP

Table 3.1: Re-Identification Datasets. Divided into single-shot and multi-shot.

that these types of automatic detection originate bounding boxes that deviate a bit from perfect ones which have an impact in Re-ID systems accuracy as already shown in [27]. Finally, the last parameter is Evaluation that corresponds to how these datasets were evaluated in their presentation. Two mainly used methods are Cumulative Match Characteristic (CMC) and Mean Average Precision (mAP) both of them will be explain in section 4.5.

The Datasets presented were captured in different scenarios. For example, GRID [53] was collected in an underground station, CAVIAR [54] in a shopping mall, CUHK01 [57], CUHK02 [58], CUHK03 [27]. DukeMTMC [61], Market1501 [60] and HDA+ [65] in university campus and iLIDS [52], Airport [62] at an Airport hall.

The VIPer [29] was one of the first Re-ID Datasets and some of its samples can be seen in Figure 3.6. This is the most tested benchmark and it also constitutes a challenging dataset due to illumination changes and pose variation. Nevertheless, considering that it is a very small dataset, it is not suitable for a deep learning approach.

CUHK01 and CUHK02 are two famous datasets and do not have a lot of images. CUHK01 has 971

people and each one has 4 images from different viewpoints captured by 2 cameras making 3884 images, while CUHK02 has 1816 people captured from 5 pair disjoint cameras which makes 7264 images. Although they are small datasets, they are good to test the system viability. In contrast, Market1501 has 32688 images captured by 6 cameras and is one of the most popular single shot Re-Id datasets which is an obvious choice when a deep learning approach wants to be taken. Nevertheless, this dataset has different distractors, - some bounding boxes that were badly made but were put in the datasets to test Re-Id systems against bad bounding box generations.

Figure 3.6: Image examples from Viper dataset. Adapted from [29].

When discussing multi-shot datasets, there are only a few, such as PRID2011 [55] and iLIDS-VID [64], however, more datasets have been developed recently with even more images and bounding boxes. One multi-shot dataset is HDA+ [65] that was proposed to be a test-bed for an automatic Re-ID system. This dataset has 30-minute video from 13 disjoint cameras with automatic detection of people provided by aggregated channel features(ACF).

4

Methodology

Contents

In this chapter, the approach adopted to achieve the objectives and solve the problem presented in 1.4 are explained. The Overall Structure of the Re-Id system is discussed in 4.1, where the architecture is explained. During this chapter all the components of this Overall Structure are addressed.

4.1 Overall Structure

The architecture chosen to address the problem is presented in Figure 4.1. In this system, it is worth mentioning three main processing blocks: (i) pre-processing block, (ii) Feature Extractor and (iii) Matching Network.

A Pre-processing block aims to prepare the different images before feeding them into the network itself. It has the job of resizing and standardising the images, before they are fed into the network. Beyond that, it is also responsible for data augmentation, i.e. increase the amount of training images based on the already available images. The work done by the pre-processing block will be further discussed in section 4.2.

The Feature Representation network is the core of the system as it is responsible for producing the features that best represent each person. This feature extractor will receive the output of the pre-processing block - an image (person) - and from that image, a feature vector will be produced. This block will produce a feature vector containing information of the input images. A more detailed description about its operation is explained in section 4.3.

Finally, the last block worth mentioning is the Matching network whose task is to bring images of the same class closer together while pushing images from different classes further apart. Throughout the work developed, this similarity matching network took different formats, which was necessary in order to better define the baseline for the work and to explore different hypotheses that could lead to the development of an even better network. An analysis of the changes made in this block is discussed in section 4.4.

All the blocks described in the previous paragraphs compose the system of Re-Identification developed in this thesis. Throughout the work, different approaches were taken in order to understand the contribution of each block. In order to evaluate the results, the typical evaluation metrics in Re-Identification are used, as it will be introduced in section 4.5.

In summary, the steps defined for the development of this dissertation were the following:

- To build a Pre-processing block to standardise the images, and to do some data augmentation;
- To use multiple datasets and fine-tuning an existing network (MobileNet) for each one, in order to produce a feature extraction network;
- To obtain the Baseline Results, replacing the similarity matching network by Euclidean distance;

- To build and train a similarity matching network;
- To obtain a rank list as the output of the pipeline results;
- To use well-known evaluation metrics to test the quality of the results;

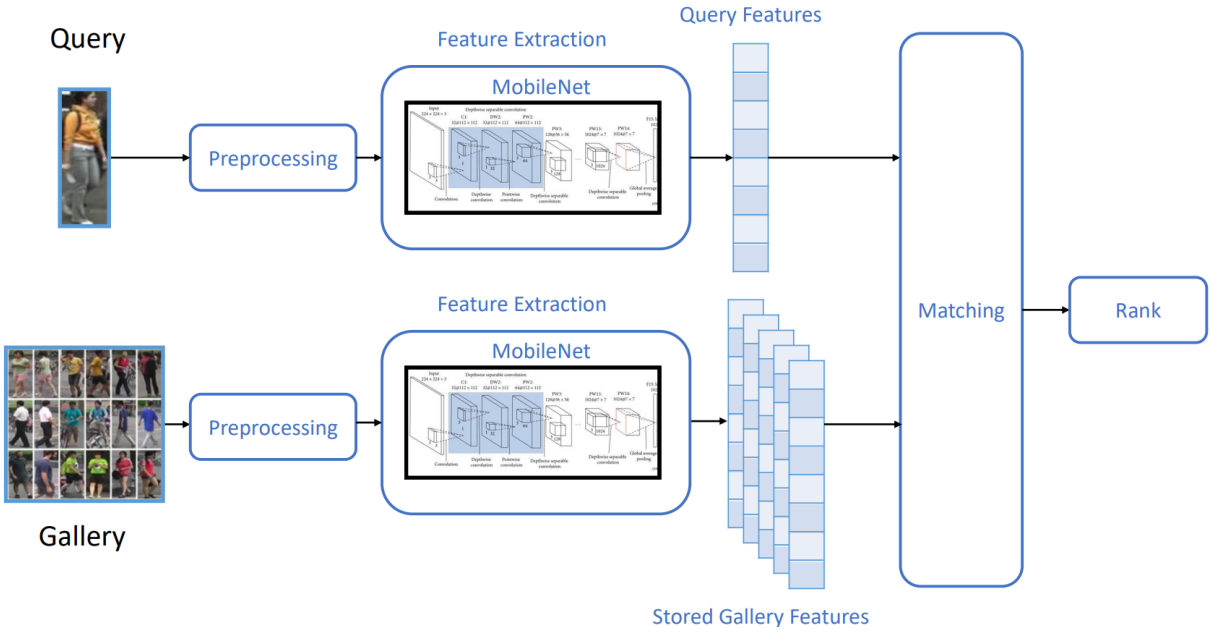


Figure 4.1: Pipeline architecture of the Re-Identification system developed during this dissertation.

4.2 Preprocessing

As it can be seen in Figure 4.1, the Pre-processing block is the first one of the Re-identification system. The pre-processing block takes the input image and resizes it to match the input of the feature extraction network, optionally performing some transformations on them for data augmentation.

As mentioned before, one of the functions is responsible for performing actions on the images, when loading them from different datasets. These actions are mostly related to the standardisation of images before they enter the CNN, since the size can vary from image to image and from dataset to dataset. This pre-processing then consists of resizing all images to obtain a size of 128×128 at the network input. This is important since the network that will be used later only takes squared images and, as most of the datasets have rectangular ones, a resizing must be done to adjust them to the network input. In order to resize all images, a bi-linear interpolation is done; this consists in adding pixels to the image by calculating the colour and intensity of the four nearest ones. In Figure 4.2(a) , the original image and the resized one are shown: one can see the resizing being done and, as a result, the expansion of the

image to the point it becomes a bit distorted. So, a question related to this topic can be considered: instead of doing bi-linear interpolation, could a simple padding be better, like the one also presented in Figure 4.2(a)? This question will be discussed in chapter 6. Another action that is essential is to scale the image pixels between 1 and -1.

(a) Original Image, Resized Image with linear interpolation, Padded image.

(b) Result of the Data Augmentation block in the same image

Figure 4.2: Result of the pre-processing block in images. At 4.2(a) the resize result and the padding are shown. At 4.2(b)different data augmentation results can be observed.

As mentioned before, the first function of the pre-processing block was related to standardisation. The second function is related to data augmentation, since it can improve network results as stated in literature [73]. Data augmentation allows the amount of training data to increase since it creates new images from already existing ones to build a bigger training dataset. Since some of the datasets used in this work are not fully diversified nor have many images, data augmentation becomes crucial and can greatly improve the feature extractor as it is going to be shown in chapter 6. Some augmentation techniques were used although they were grouped into weak and strong techniques. The weak data augmentation are:

- Rotation - Rotate the original image between -20º and 20º. This aims at changing the original images as they were obtained in different angles by the camera, making the network more robust to different viewpoints.
- Zoom - Random zoom in the image. This aims at changing the original image to capture closer people to the camera. Once again, it improves the network with respect to different viewpoints.
- Translation - Random translation of the image in width and height to improve different viewpoints.
- Shear range - To distort an image along an axis to simulate different angles.
- Horizontal flip - Mirror an image in the horizontal direction to capture different viewpoints.
- Brightness - Consists in changing the brightness of the image with the goal of simulating different illumination changes.

The strong data augmentation method is Random Erasing [51]. It shows that Re-Identification systems benefit a lot by using it. This method consists in randomly erasing, or not, an area of pixels in an image. This is a good practise since the network can learn how to deal with occlusions. The results of data augmentation can be seen in Figure 4.2(b).

We also test whether the image size matters for a Re-Identification task. The most common size is 128×128, but in some datasets this may result in loss of information that should be avoided. Thus, two new image sizes will be tested, in particular 224×224 and 256×256.

4.3 Feature Extraction

The feature representation network is a key part of the Re-Identification system since it has the responsibility to produce the best feature representations for the task that bring the same classes closer to each other while pushing the others to a distance. As a continuation of the network chosen in the work [49], the MobileNetV1 was chosen as the CNN for the feature extractor, having some advantages when compared to other networks. For example, it has fewer parameters that lead to less training time without losing too much in terms of accuracy, and also, it takes less disk space which can be an advantage when a Re-Identification system wants to be deployed. On an attempt to find a more lightweight architecture the MobileNetV2 [20] was used instead of MobileNetV1 [19], on the expectation that the residual connections on MobileNetV2 could bring advantages, this was not the case, as shown in chapter 6, and therefore the MobileNetV1 was the chosen one. These networks can receive different sizes, ranging from 96 to 256, at the beginning so a $(128 \times 128 \times 3)$ was chosen as the image network input, 128 being the width and height and 3 the number of colour channels (RGB).

The training of this network from scratch can lead to overfitting since there is not too much data to train. In this way, a technique discussed in 2.2.5.D (Transfer Learning) was used and the MobilenetV1 weights from Imagenet [14] were used to initialise all layers of the network.

As discussed in section 3.2.3 , there are two common methods that can be used when someone wants to build a Re-Identification system. In this case, both were used as stated in [48]: the Classification and the Siamese network. At the beginning, the network was trained for classification purposes with a part of the dataset. The original classification head of the MobileNetV1 was modified to adapt the output to the number of identities present in each dataset as can be seen in Figure 4.3. This new classification head is constituted with an Average Pooling layer, two fully dense layers with 1024 neurons, a dropout layer established at 0.5 and, finally, a softmax layer with the size of the different ids at the training data. Then, the Siamese network was built based on the weights obtained from the previous training. The softmax and droupout layer of the classification block were removed and the resulting head is as shown in the figure 4.4. This network is now used in a Siamese architecture to train the Re-ID task jointly with

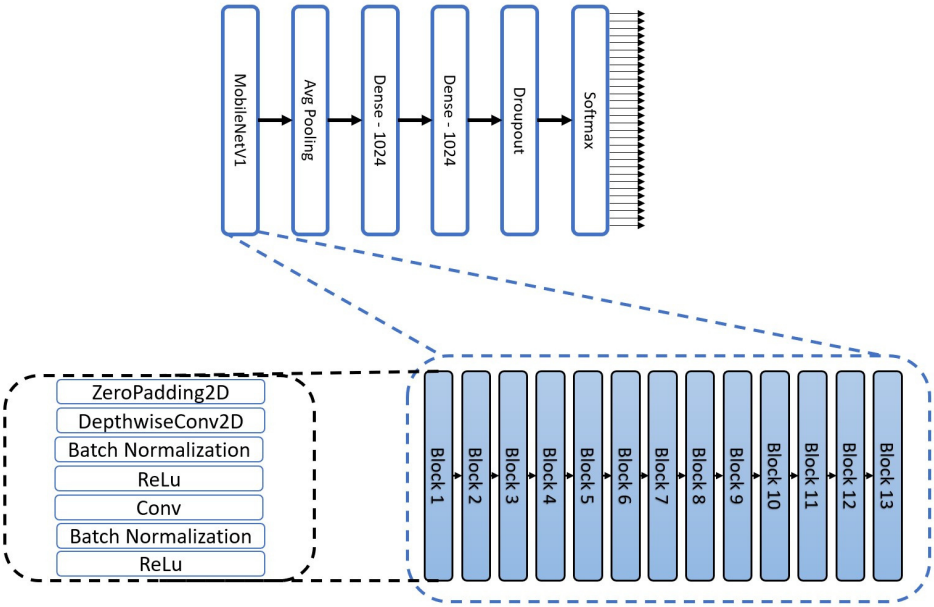


Figure 4.3: Feature Extraction Network structure that was trained for the classification task. At the bottom of the figure, the structure of the MobileNetV1 can be seen, where at the left side the standard MobileNetV1 block is represented. At right, the MobileNetV1 structure is shown. At the top, the feature extractor is shown with the addition of a classification head to allow training for classification purposes.

the matching network, to be presented, for the Re-Identification task.

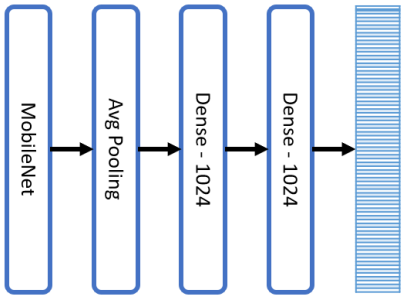


Figure 4.4: The feature extractor architecture is represented, it starts with the MobileNet whose classification head was truncated and a new one was made. After MobileNet, an Average Pooling layer, two fully dense layers with 1024 neurons, a dropout layer established at 0.5 to produce the feature vector as shown.

4.4 Matching Network

As stated in 3.2.2, throughout the years, several matching techniques were used to compare two feature vectors. The methods described are mainly related to hand crafted systems and explain how, from two feature vectors, could one obtain a good feature representation that can assess the similarity of the input patterns through and appropriate distance metric. Nevertheless, one important technique has gained

reputation in matching networks as referred to in section 3.2.3, that is to use a deep networks to distance vectors apart or bring them closer depending on their id. In this work, this last approach will be taken, and a similarity matching network will be built and trained to compute a similarity score between the two input patterns.

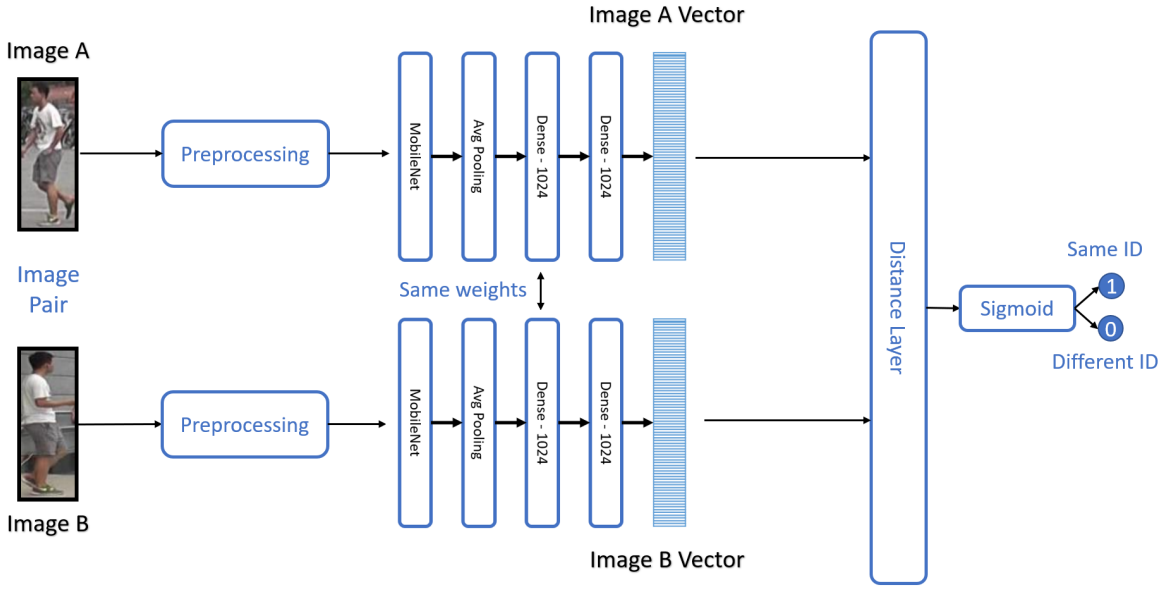


Figure 4.5: An end to end Re-Identification system is represented. It starts with two images at the network input being pre-processed. They then go through the feature extraction network where each one produces a vector. Finally the vectors produced are compared with each other using the Euclidean distance. Then this value goes through a sigmoid function where the result of 1 (belong to the same id) or 0 (do not belong to the same id) is shown.

In order to define a baseline, this work starts by defining the matching network as the simplest possible form: the Euclidean distance. This technique will take two feature vectors obtained by the network, as discussed above in 4.3, and will calculate the distance.

In a first instance, a Siamese network was built as shown in Figure 4.5. As it can be seen, both branches have the same composition and the same weights - it is indeed the same network duplicated - originating, as a consequence, two feature vectors that will go through a last comparison layer where the distance is calculated and passed through a sigmoid that will check if the pair images belong to the same id (greater than 0.5 in the sigmoid output) or not (less than 0.5 in the sigmoid output), it is then possible to sort the images according to their distance to the query. To compare with this baseline, we will change the Similarity Matching Network to allow a train with the Contrastive loss [74] or Triplet Loss [75].

Contrastive loss will allow the images to be distanced together or apart depending on their id, as explained in section 2.3.1. In this way, the metric learning can be trained contrary to what was happening in Euclidean Distance (Figure 4.5). The structural implementation of the Contrastive loss can be seen in

Figure 4.6. The differences, in relation to this figure, are a new batch normalisation layer and the loss, as presented in equation (2.4).

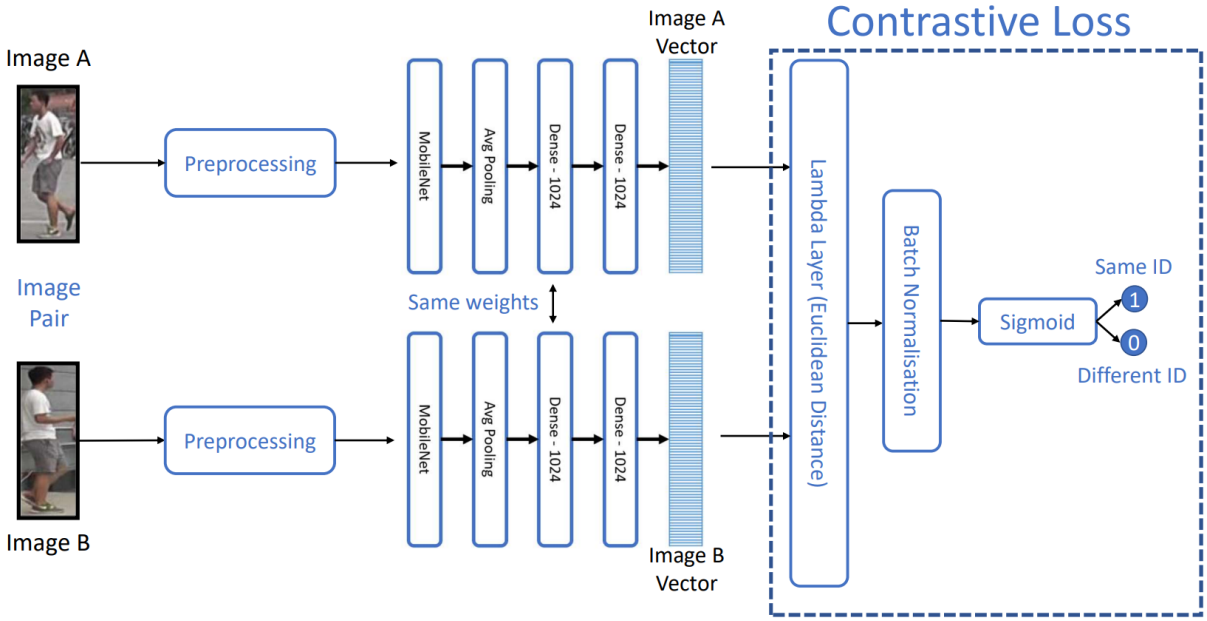


Figure 4.6: The implementation of the Constrastive Loss in the Matching Network can be seen. The first part of the network is the feature extractor that produces two feature vectors, as explained. These vector go through a Lambda Layer, that calculates the Euclidean distance, after they suffer a Batch Normalisation and finally the Sigmoid layer will produce the final result. The big difference is that all the part inside the dotted line is trained in a deep manner being Contrastive the Loss.

Triplet Loss will also allow to distance images together or apart depending on their id at the same time, as explained in 2.3.2. In this way, the whole network can be re-trained in a similar manner to the process that happens for Contrastive Loss. The training process can be found in Figure 4.7, where inside the triple loss block, the vectors are normalised and the distance between the anchor-positive and anchor-negative is calculated and consequently the loss itself. This will change some weight values that will allow to distance the classes apart and therefore improve the network.

4.5 Evaluation Metrics

In the specific case of Re-Identification, most of the papers are based on two main evaluation metrics, namely Rank Accuracy which can be combined with CMC and mAP. In this way, this dissertation will also have those two as evaluation metrics to compare the different results obtained with the state of the art and between them. However, this cannot consist on a comparison of a single sample, because the results may vary due to several factors. In this way, the differences seen have to be statistically significant in order to tell with confidence if a method is better than other. In order to do so, it will be

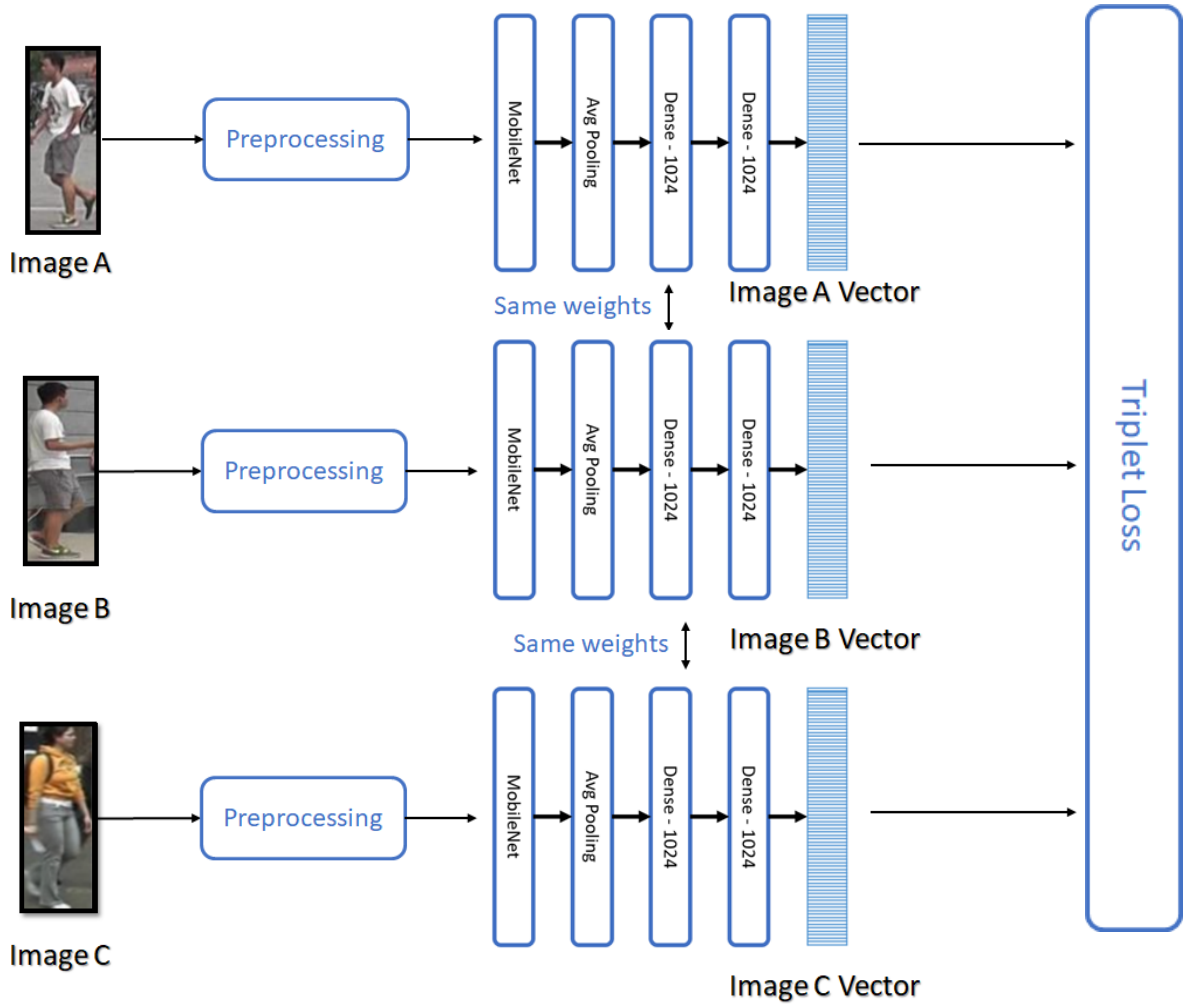


Figure 4.7: The implementation of the Triplet Loss in the Matching Network can be seen. The first part of the network is the feature extractor that produces two feature vectors, as explained. These vectors go through a Batch normalisation, and then the anchor-positive and anchor-negative is calculated and subsequently the loss associated to each triplet.

used a statistical hypothesis test. All three methods referred to are explained in this section. In addition, metrics to evaluate a classification network performance are also shown since they will be used in the first part of the work.

4.5.1 Metrics to evaluate training performance

This dissertation will also focus on the evaluation of the performance of the trained classification network. In order to do this, the most appropriated metrics must be combined since only one metric cannot translate how good or bad a system is. In this way, there were chosen 4 metrics: Accuracy, Precision, Recall and F1 score.

Accuracy is the ratio between the correct prediction and the total number of predictions, it can be defined as:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}, \tag{4.1}$$

this metric tells how much accurate the systems is. However, if one class has more data than other, for example 98% of class A and 2% of class B then, even if the classifier is bad and chose always class A as the correct one, the system will have 98% of accuracy. That is why other metrics should be used.

Precision is the number of true positives (positives correctly classified) divided by the number of all positive results (positives even if badly classified) like:

$$Precision = \frac{True\ positives}{True\ positives + False\ Positives.} \tag{4.2}$$

Recall is the number of true positives (positives correctly classified) divided by the number of all samples that should be classified as positive like:

$$Recall = \frac{True\ positives}{True\ positives + False\ Negatives.} \tag{4.3}$$

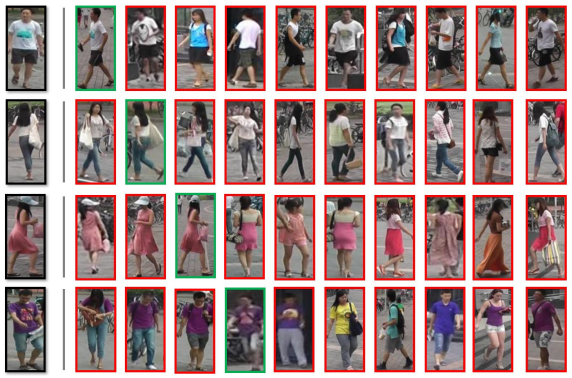
Finally, the F1 score is the combination of Precision and Recall translating how many times a system is able to get the correct classification. This metric tells how accurate a system is by looking at these two metrics. The equation that translates this metric is:

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}} \tag{4.4}$$

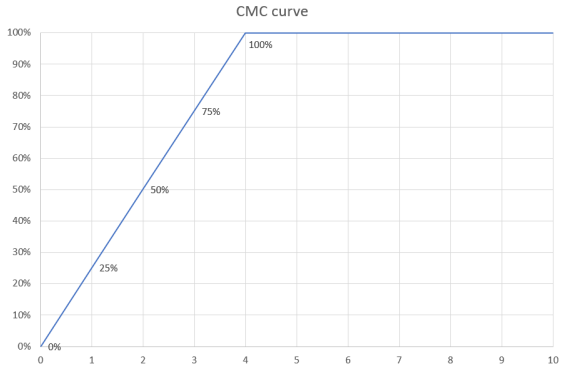
4.5.2 Rank-k accuracy & CMC curve

When discussing rank-k accuracy, one can say that a query is given a rank-k when it appears at the k position returned by the Re-Identification system. The main goal of the system is to return all the correct matches at the first positions of the list. If, for 10 queries, half of them return the match in the first position of the list and the others in different positions further down in the list, then it can be said that this system has 50% rank-1 accuracy.

As the system cannot be perfect, and some matches cannot be at the beginning of the list, it is important to look at higher ranks like rank-5, rank-10 and rank-20. The accuracy of these ranks can be viewed using CMC. In Figure 4.8(a) and in 4.8(b) a graphical example of the rank representation and a CMC curve is shown.



(a) Graphical rank representation



(b) CMC curve example

Figure 4.8: Figure 4.8(a) shows the different matches for 4 different queries and the final list returned by the system. As it can be seen, the rank-1 accuracy is 25% since there is only one query that returned the correct match in the first position($\frac{1}{4} * 100 = 25\%$). The CMC curve is also shown in Figure 4.8(b) and the result for each rank is demonstrated. At rank-5 the accuracy is 100% since all the correct matches are found. The images of the same id as the query are represented in green frame, while if it is of a different id, by a red frame.

4.5.3 MaP

When there is more than one per query presented in the gallery, the Rank-k accuracy is not the best metric to be used since it only reports the first appearance. As it can be seen in figure 4.9, there are four systems; even if all of them have the same Rank, they are different and some of them are better than others. In this way, mAP should be used. This metric consists in calculating the mean Average Precision of all queries, as seen in

$$mAP = \frac{1}{n} \sum_{k=1}^{n} AP_k, \tag{4.5}$$

where n denotes the number of queries and AP the average precision.

Calculating the average precision (AP) for each query is essential and can be calculated using the equation shown below:

$$AP = \frac{1}{m} \sum_{i=1}^{x} \left(Precision@i \times rel@i \right), \tag{4.6}$$

where m is the number of correct matches for a given query, x each position of the returned list, $Precision@i$ is the precision at the position i and $rel@i$ the relevance function - is 1 if the sample is correct and 0 otherwise.

A good Re-Identification system should have both metrics at a high percentage, and both metrics should be used together since they can translate more information of the system to the results.

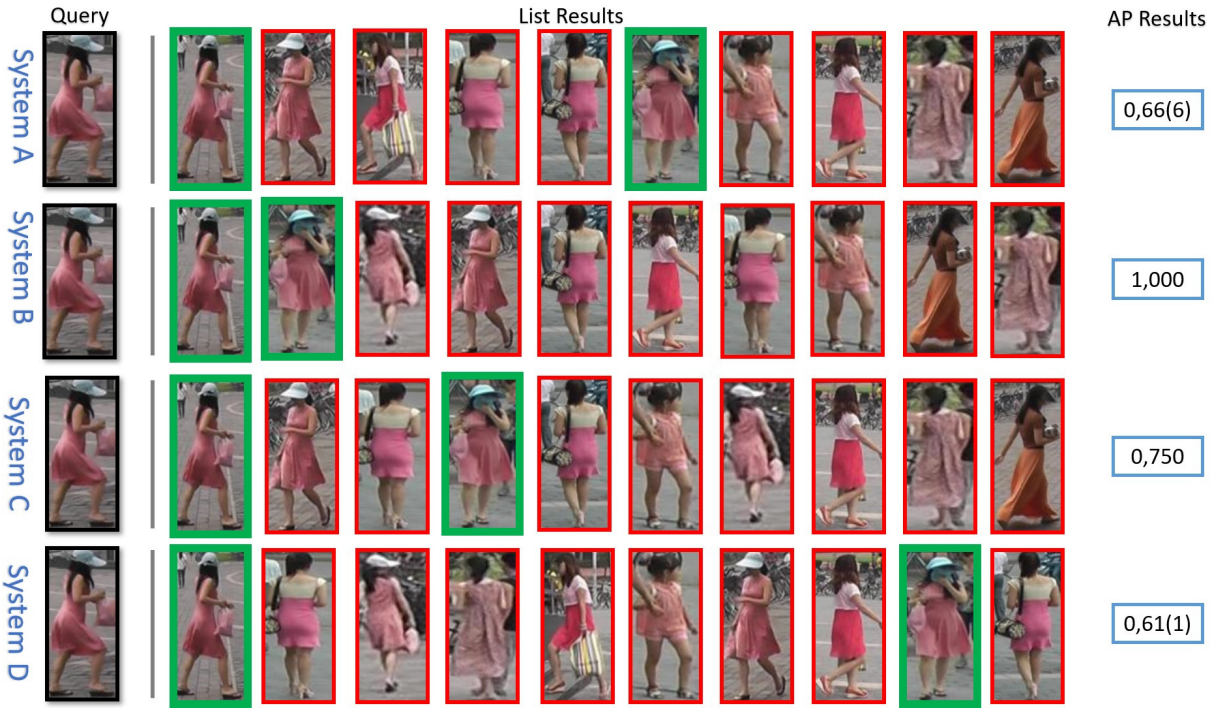


Figure 4.9: For the same query, four systems with rank-1 accuracy are represented. Considering 2 matches for the query and a returned list with 10 positions, the Average Precision for each one is $AP_a = \frac{1}{2}(\frac{1}{1} * 1 + \frac{1}{2} * 0 + \frac{1}{3} * 0 + \frac{1}{4} * 0 + \frac{1}{5} * 0 + \frac{2}{6} * 1 + \frac{2}{7} * 0 + \frac{2}{8} * 0 + \frac{2}{9} * 0 + \frac{2}{10} * 0) = \frac{2}{3}$, $AP_b = \frac{1}{2}(\frac{1}{1} * 1 + \frac{2}{2} * 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0) = 1.00$, $AP_c = \frac{1}{2}(\frac{1}{1} * 1 + 0 + 0 + \frac{2}{4} * 1 + 0 + 0 + 0 + 0 + 0 + 0) = \frac{3}{4}$ and $AP_d = \frac{1}{2}(\frac{1}{1} * 1 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + \frac{2}{9} * 1 + 0) = \frac{11}{18}$. The images of the same id as the query are represented in green frame, while if it is of a different id, by a red frame.

4.5.4 Statistical Hypothesis test

A Statistical Hypothesis test is a procedure to show which of the two hypotheses is better, using data produced by both hypotheses and comparing them. In this case, this statistical test is needed in order to compare different systems and decide which one performs better.

There are several statistical tests, however, for each situation, the most suitable one must be chosen since they are designed for specific situations that fullfill specific requirements. In order to choose the most suitable one for this project, the procedure presented in [76] helps choosing which one is better considering different requirements:

- What kind of data is collected ? - In this case what is being compared is the results obtained in each system, namely the mAP.
- How many independent variables will be used ? - In this case only one independent variable is taken into account, since it is manipulated. This variable is the system itself.
- Which design will be used ? - An experimental design will be used instead of a correlational one, since different conditions for the independent variable will be tested.

- Independent Measures or Repeated Measures? - Since each time a measure is repeated in the same conditions, it can be said that repeated measures exist.
- Parametric or Non-Parametric ? Parametric test assumes that the data is normally distributed and the groups have equal variance. In this case, it cannot be assumed that the data is parametric since it does not follow a normal distribution - it is assumed to be a non-parametric data.

In this way, following this procedure, it can be assumed that the test to be chosen is the Wilcoxon Signed-rank test as it is used to compare two groups that have the same conditions and participants. However, for this assumption, it was assumed that the data was not normally distributed. This assumption was made as the worst case scenario and since it does not put any future validations in question, this assumption can be done. If with the Wilcoxon test one can not validate with confidence that one system is better than other, it is then necessary to see if data has a Gaussian distribution and the T-Test (version of the Wilcoxon for non-parametric data) must be done. As for the course of the work, this non-parametric function works. After using this test, where a group of samples is compared to another group, but in different conditions, a confidence interval is the output. This interval will state if the hypothesis of one being better than other can be rejected or not.

In this dissertation, the Wilcoxon test will be used to compare results produced by different systems that analyse different similarities and produce the different ranked lists, and obtain the system results based on Re-ID metrics namely mAP. In order to do this, 10 samples of each system results will be produced and compared against each other, as it will be further explained in chapter 6 and are compared against each other. First, if we obtain a $p - value > 0.05$, this means that the approaches are not statistically different, better saying, we can not state anything about the data. On the contrary, if the $p - value < 0.05$ the approaches are statistically different, since now we can reject the null hypothesis that states there is not a statistically significant difference between results of the proposed method and the other methods compared During the course of this work, only this confidence level will be used and the test will be done as this example explained. If the Wilcoxon test can, in fact, state with a confidence level that a system is better than other (analysed in terms of mAP), an asterisk will mark that system.

5

Implementation

Contents

This chapter will address the implementation of the Re-Identification system and all the experiments referred to in chapter 4. All the steps in each phase of the project will be described in detail. This chapter will address the datasets used to implement the Re-Id system in 5.1, the implementation of the feature extraction block in 5.2 and Deep Metric Learning implementation in 5.3. The implementation code for all the experiences made in this dissertation can be found in the following GitHub: *https://github.com/francisco-p/Deep_Re-ID.git*

5.1 Datasets

As previously stated there are a lot of different datasets for Re-Identification, each one with their own characteristics. From all the datasets presented in section 3.3, four were chosen due to their characteristics.

The first one is CUHK01 [57], captured in the "Chinese University of Hong Kong, with a total size of 3884 images and 971 identities. Each identity has a total of 4 images: two from the side and 2 from the front/back. Compared to more recent datasets, this one has fewer images so it can be a big challenge for deep learning. Nevertheless, it may be a good dataset to test the benefits of data augmentation and to understand whether a small amount of data can produce good results. Some examples of CUHK01 can be seen in Figure 5.1(a) in which the image size, in each sample, is 60x160 RGB.

The second dataset is CUHK02 [58], similar to CUHK01. This dataset was also captured in the Chinese University of Hong Kong although this one has a larger number of images and identities than CUHK01; more specifically 7264 images and 1816 identities. These images were captured from 5 pairs of disjoint cameras and each person has 4 images. This is an improvement when compared to CUHK01 since it has more images. Some examples can be seen in Figure 5.1(b): the image size in each sample is 60x160 RGB.

(a) Examples of CUHK01 dataset **(b)** Examples of CUHK02 dataset

Figure 5.1: Examples of Re-Identification datasets (CUHK01 and CUHK02) used in this dissertation.

The third chosen dataset is Market-1501 [60] captured in Tsinghua University. This is probably the

most popular dataset nowadays, since it has a lot of identities and images, which makes it a good candidate for deep networks. It has a total of 32668 images and 1501 identities captured by six different cameras. While the first two datasets were captured and labelled by hand, this one was using DPM. The image size in this dataset is 64x128 RGB.

Finally, the last chosen dataset is HDA+ [65]. This dataset was captured in a Portuguese University, Instituto Superior Técnico. In contrast to the previous ones that were single shot, this dataset belongs to the multi-shot class. It has a total of 16844 images from 65 different people, captured by 12 different cameras that were recorded simultaneously for 30 minutes. The number of images obtained only considers the not occluded ones, using the tools provided by the authors. The images size varies in the dataset since they were obtained by ACF. This method detect the person and creates a bounding box around it, which may lead to different bounding boxes if the person in question is far from the camera. Thus, the result of the image size will vary depending how this person appear in the general frame as there is no standardisation done by the algorithm.

(a) Examples of Market-1501 dataset

(b) Examples of HDA+ dataset

Figure 5.2: Examples of Re-Identification datasets (Market-1501 and HDA+) used in this dissertation.

In order to use these datasets for Re-Identification purposes, it is crucial to divide correctly the training and testing set. Moreover, to be able to compare with the results of the state-of-the-art papers, it is essential to follow the same procedures for dataset division. In this way the division of each dataset used in all the work carried out (for the classification and Re-ID task) will be explained.

CUHK01 has 971 entities, as it is stated in [57]. From the 971 entities, 100 are chosen to test and the rest of the dataset is used for training. In this way, the training dataset has (871) - where the number between curved brackets represents the number of entities - and the test dataset (100). In Figure 5.3 a more graphic explanation can be found.

Figure 5.3: Division of the CUHK01 dataset between training and testing. Each square represents an image and in this case each person has 4 images so each one has 4 squares. The green colour represents the training set that are all people from 1 to 870. The yellow colour represents the test set, from 871 to 971.

CUHK02 has 1816 entities, and similarly to CUHK01, 100 entities are chosen to test, and the rest of the dataset is for training. In this way, the training dataset has (1716), and test dataset (100). In Figure 5.4 a more graphic explanation can be found.

Figure 5.4: Division of the CUHK02 dataset between training and testing. Each square represents an image, in this case each person has 4 images so each one has 4 squares. The green colour represents the training set that are all people from 1 to 1715. The yellow colour represent the test set, from 1716 to 1816.

Market-1501 has 1501 entities. However, the division is different from the stated above since the procedures presented in the dataset papers were followed. In [60] it is explained that the dataset should be divided in half: this is, 750 for training and 751 for testing. In this way, training dataset (750), and test dataset (751). In Figure 5.5 a more graphic explanation can be found.

Figure 5.5: Division of the Market-1501 dataset between training and testing. Each square represents an image, in this case each person has different number of images. The green colour represents the training set that are all people from 1 to 750. The yellow colour represents the test set, from 751 to 1501.

HDA+ has 66 entities and the approach taken was like the one in Market-1501. The dataset should be divided in half: this is 33 for training and 33 for testing. In this way, training dataset (33), and test dataset (33). In Figure 5.6 a more graphic explanation can be found.

Figure 5.6: Division of the HDA+ dataset between training and testing. Each square represents an image, in this case each person has different number of images. The green colour represents the training set that are all people from 1 to 33. The yellow colour represents the test set, from 34 to 66.

5.1.1 Gallery creation and Testing

To test the system, it is important to create both the gallery and queries. After the creation of the gallery and the query list, all the feature vectors are obtained. Then, each query feature vector is compared with each gallery feature vector, obtaining a ranked list with the different gallery identities. At the top of the list, there are images that the system considers to be more similar to the person of interest (query). Based on this list, the rank-k accuracy and mAP can be calculated.

The literature states that, in order to create the gallery and query sets, one or more images per person of the dataset must be in the gallery set. Having more than one image is better to understand the system viability to return multiple images of the same person. As for the query set, there is no predefined number. Taking this into consideration, and in order to evaluate this work, the sets are composed as follows: (i) **Gallery Set**: selection of two images of each person present in the dataset. (ii) **Query Set**: selection of 100 random people from the test dataset. In the case of HDA+, only 33 people can be selected.

Following these rules, gallery sets will have the size of 1942 (CUHK01), 3632 (CUHK02), 3002 (Market-1501) and 132 (HDA+), while query sets will have the size of 100 (CUHK01), 100 (CUHK02), 100 (Market-1501) and 33 (HDA+). Each query will have two images of itself present in the gallery. In order for the system to reach 100% of mAP result, the returned list has to have the two images with the same id as the query at the first and second positions for the 100 queries made.

The statistical comparison of systems is important due to the effect of natural variability of visual patterns, and only one sample does not allow that. In this work, each system was tested for 10 different galleries and queries sets. The creation of these 10 different galleries and queries set depends on the datasets. In CUHK01 and CUHK02, as there are a limited number of images per person (4), it is difficult to have different galleries. In this way, for these datasets, the gallery is always the same. Although the queries will always have the same ids but different images each time. As for Market-1501 10 different galleries and queries can be obtained. Regarding the gallery, as there are lot of images per id (~ 20), choosing only two from each one will contribute to very different gallery compositions, as for the queries only 100 ids must be chosen from 750 allowing for very different query list composition. Finally, for HDA+, both the query and the gallery will always have the same ids but with different images each time. The results obtained were the average and standard deviation of all 10 rank-k and mAP results. This procedure will allow to perform the Wilcoxon test to verify if one method is better that another, with statistical significance.

5.2 Feature Extraction Analysis

In this part of the work, the focus will only be on developing a good and efficient feature extractor. All the feature extractors presented below will produce feature vectors that will allow for the comparison between gallery and query images using a simple Euclidean distance, obtaining this way the ranking results. They consist in:

- **Baseline** - In order to define the baseline for this work, the network previously developed in [49] will be used. It consists on a MobileNetV1 where the classification head is removed and a new one is added similarly to the procedure explained in 4.3. This will be the baseline and a point of

comparison for future changes.

- **Baseline + 2048** - It consist on the same feature extractor as the baseline, as shown in figure 4.4, but instead of having dense layers with size of 1024 at the end of the network, it has layers with size of 2048 to verify if more information can be retrieved with bigger dense layers.
- **Baseline + MobileNetV2** - It consists on the feature extractor structure discussed in 4.3, and like the Baseline. Nevertheless, a substitution of the backbone network is made and MobileNetV1 is replaced by MobileNetV2.
- **Baseline + Padding** - Instead of using linear interpolation, for resizing an image, as discussed in 4.2, the benefits of using a padding in images will be analysed and explored as shown in figure 4.2(a).
- **Baseline + Data Augmentation** - Performing the first group of data augmentation techniques which include Rotation, Zoom, Translation, Shear Range, Horizontal flip and Brightness as stated in 4.2.
- **Baseline + Random Erasing** - Performing random erasing on the different datasets as stated in 4.2.

As discussed, the Baseline will be built, and changes will always be made based on it. This is, the Baseline is always used, and the different changes are compared against it, in order to find out whether it is beneficial for the final network or not. Finally, if the changes are good, a final feature extractor will be trained encompassing all techniques that improve the network.

In addition to the changes to be done, as already presented, other experiments will be carried out as referred in 4.2; this is resizing the input images. This experiment will be performed based on the improved baseline, occurring at a different time of others experiments. The first image input is 128×128 as already mentioned. At this, a 224×224 and 256×256 input image size wants to be tested to to check whether there is any improvement when compared to the 128×128 size. In order to do this, the network will be re-trained but with a change in the input sizes in the network. Consequently, the network will have more parameters and take longer to train.

In order to do all the experiments, it is important to discuss how the different datasets will be divided and how the network will be trained. As already explained in 5.1, a division of the training partition between training and testing is done, even if inside the training part of the dataset, it is necessary to divide between training and validation. So, considering that the network will be trained for classification purposes, the best way to divide them is, for each person, to set aside some images for training and the other images for validation. In CUHK01 and CUHK02, it is very straightforward. For the four existing images for each person, one is for validation and the rest is for training. However, for Market-1501 and

HDA+, the number of images per person is not so straightforward which means that the split is 33% for validation and the rest for training. This can be seen in figure 5.7.

Figure 5.7: Dataset split for Feature Extraction training. From the top to the bottom, it can be seen graphically the division in CUHK01, CUHK02, Market and HDA+ respectively. Green means training images, orange means validation and yellow means test.Each square represents an image belonging to a person.

5.3 Deep Metric Learning

After the experiments in section 5.2 are completed, there will be a feature extraction network. It will be able to extract feature representations that will contribute to differentiate among people and, if the experiments are successful, the results of this network are already promising using the Euclidean Distance baseline as matching function. In this way, with access to a good feature extractor, the construction of a matching network, as explained in 4.4 , can be done.

The construction of the Matching Network is a Deep Metric Learning approach. This approach will not teach a network to classify people into different classes, as it is done by the feature extractor, but instead it will teach how to differentiate people from each other.

5.3.1 Contrastive Loss

The structure presented in figure 4.6 will be built as well as the Contrastive Loss definition, as in equation (2.4). Therefore, what was needed was to train the added block (dotted block), since it was initialised with random weights. However, in the course of this work, it was realised that retraining all layers of the network gives even better results than training just the new block.

Contrary to the training that was done for the feature extraction network, where only one image was needed as an input to the feature extractor at this time a pairs of images are used to train the Siamese network using the Contrastive Loss. In this way, both positive pairs (pair having two images of the same

class) and negative pairs (pairs having two images of different classes) must be created. To balance the training dataset, an equal number of positive and negative pairs will be created. The number of positive pairs created should be the highest possible. This is, for a person, all its images will be paired together. If a person has four images of itself, then 6 positive pairs (result of all possible combinations between the images) of this person can be created.

However, the datasets used are different and each one needs to be analysed carefully when creating the pairs. The creation of pairs will be both for training and validation parts of the datasets. For CUHK01 and CUHK02, there is not too much data for each person. The solution is to perform some data augmentation on existing images in order to create more pairs. For these datasets, each image was augmented into two new ones, so as to make 8 images for each person and a total of 28 pairs per person. This augmentation is only performed on training data with the validation part left unchanged. An example of this procedure is explained graphically in figure 5.8. For Market and HDA+ there is no need for data augmentation as there is a lot of images per person. In this way, for the Market, the maximum number of positive pairs for each person is created and it can vary since not all people have the same number of images, in consequence an average 20 positive pairs are created per person in this dataset. For HDA+, as there is a lot of images per person, only 20 are considered for each one. Accordingly to what was said, the number of pairs varies from one dataset to another. For CUHK01 the dataset has 44318 pairs, CUHK02 has 91626 pairs, Market has 345078 pairs and HDA+ has 164688 pairs.

Figure 5.8: Demonstration of the distribution for matching network for CUHK01 and CUHK02 datasets. At the top, the distribution of training (green), validation (orange) and test (yellow) dataset. In the middle, the augmentation is already done for the training dataset, which leads to 8 images instead of 4. In the validation dataset there are 100 people with 4 images each. Finally, at the bottom, the creation of pairs can be seen, which leads to 28 positive pairs and 28 negatives in compensation.

5.3.2 Triplet Loss

The structure shown in figure 4.7 will be implemented as the triplet loss. After the feature extraction, a normalisation layer will contribute to normalise all images and then the euclidean distance between

anchor-positive and anchor-negative will be calculated to allow for the loss calculation as explained in section 2.3.2. Having this loss, the training procedure can begin and the whole network can be re-trained (similar to what happens in contrastive loss) with triplet loss, where triplet of images are sent to the network being two of the same id and one of a different one.

The big problem when implementing the triplet loss is to make triplets good enough to improve the network and to obtain better results. Triplets that have a loss equal to 0 do not teach anything new and can make the network worse. In this way, for this dissertation it is used offline mining triplet. This consists in an offline manner (not during training) to make triplets and to make a forward propagation through the network to calculate the loss value. If the loss is 0, the triplet is discarded, otherwise the triplet is saved because it belongs to the semi-hard triplets class or hard triplet class as discussed in section 2.3.2.

In order to produce different triplets, a similar procedure to the one used for making pairs was adopted. For each dataset all the positive pairs per id were identified and made. After that, all negative vectors from different classes were added to the pair, making a triplet, in an exhaustive manner, this is using all images available, and the loss was calculated. In this way all triplets that will have a positive loss were identified and prepared to be the training data. For CUHK01, CUHK02 and HDA+, besides the time used to make a triplets, no further problems were identified. However, for Market-1501, as there are a lot of images per id and a total of approximately 11000 images, this performance would imply a huge amout of time spent. In this way, it was opted to choose a maximum of 8 images per id and the same procedure for other dataset was replicated in this condition.

For this implementation, data augmentation was not used for any dataset. For CUHK01, from the whole dataset, were identified 992077 triplets were identified to train the networks. For CUHK02, 1324176 triplets could be identified. For Market-1501 more triplets could be found as there are more images, in this way, 8141090 triplets were identified. Finally in HDA+ 151474 triplets were found.

6

Results

Contents

In this chapter, the different hypotheses presented in chapter 5 will be implemented. The results of the experiences will be analysed with the aim to check whether they are beneficial for the system and worthy to be implemented. To obtain these results a GPU GeForce GTX 1080 Ti was used. In addition to the results presented in this chapter, Appendix A shows the results obtained for an experiment for which no conclusions were drawn. In Appendix B, it can be seen a graphical representation of the results obtained during this work.

6.1 Feature Extraction

In this section, the results for the Feature Extractor will be presented, as well as the experiments made. The work developed in this section is based on the implementation presented in section 5.2.

To obtain the baseline results, the procedure explained before was followed. The MobileNetV1 (Feature Extractor) was fine-tuned for each dataset as the Classification task until convergence was achieved; the loss used was the categorical cross-entropy since this is a multi-class problem. The optimiser was Stochastic Gradient Descent (SGD) with batch size of 16, learning rate of 0.01 and learning rate decay of 0.1 every 10000 batches, similar to [49]. The baseline results expressed in Rank-k accuracy, k = 1, 5, 10, and mAP for each dataset are presented in Table 6.1. The table is divided into two parts: (i) Classification (ii) Ranking Results.

Baseline	Classification					Ranking Results									
						Rank-1		Rank-5		Rank-10		Rank-20		MAP	
	Accuracy	Loss	Precision	Recall	F1 score	Value	SD	Value	SD	Value	SD	Value	SD	Value	SD
CUHK01	0.50	2.72	0.40	0.50	0.43	54.70	2.49	72.10	2.12	80.90	1.51	85.70	1.35	42.41	1.83
CUHK02	0.51	2.50	0.41	0.51	0.44	38.80	3.16	59.30	2.79	67.00	2.19	77.00	2.05	27.57	1.73
Market-1501	0.81	0.74	0.82	0.81	0.80	45.80	5.42	70.00	2.90	78.00	2.14	86.40	2.06	41.85	3.14
HDA+	0.99	0.04	0.99	0.99	0.99	56.97	3.78	67.01	4.15	75.46	5.50	79.70	6.50	53.66	2.76

Table 6.1: Baseline results for all datasets according to the work developed by [49].

In the first part, the Classification results are related to the fine-tuning of the CNN, which is the MobileNetV1 in this case. The results of the fine-tuning are classified according to parameters like Accuracy, Loss, Precision, Recall and F1 Score. When analysing them, one can see that for the first two datasets (CUHK01 and CUHK02) the results are : 50% of accuracy and has a high loss. However, for Market-1501 and HDA+ the results are better with accuracy reaching 81.4% and 99.2%, respectively for Market-1501 and HDA+. Although the goal is not to obtain a classification network, these parameters are a good indication of what can be expected from the final networks.

In the second part, the Ranking results are presented. In this part two fields can be seen: (i) Value field that shows the mean value of the results for the 10 different queries and galleries and (ii) SD (Standard Deviation) field is the distribution around the mean of all 10 results obtained. Table 6.1 presents values for each rank and mAP.

Parameters	Classification for CUHK01					Classification for CUHK02				
	Accuracy	Loss	Precision	Recall	F1 score	Accuracy	Loss	Precision	Recall	F1 score
Baseline	0.504	2.717	0.40	0.50	0.43	0.513	2.500	0.41	0.51	0.44
Baseline + 2048	0.548	2.488	0.44	0.55	0.47	0.562	2.198	0.47	0.56	0.49
Baseline + MobileV2	0.460	2.971	0.36	0.46	0.39	0.467	2.768	0.37	0.47	0.40
Baseline + Padding	0.378	3.425	0.29	0.38	0.31	0.406	3.113	0.32	0.41	0.34
Baseline + DataAug	0.836	0.733	0.78	0.84	0.80	0.825	0.741	0.76	0.82	0.78
Baseline + RandomErasing	0.670	1.710	0.58	0.67	0.61	0.660	1.497	0.57	0.66	0.60

Table 6.2: Classification Results for the neural networks built for the different experiments in CUHK01 and CUHK02 dataset.

As theoretically predicted, since hand craft systems do not achieve high results, the baseline ones were low. Therefore, it is intended to increase them through the experiments presented in 5.2. The results obtained are presented in two parts: (i) Classification results for each network, in each dataset, presented in tables 6.2 and 6.3 and (ii) Ranking results for each network in each dataset, presented in tables 6.4, 6.5, 6.6 and 6.7.

Parameters	Classification for Market-1501					Classification for HDA+				
	Accuracy	Loss	Precision	Recall	F1	Accuracy	Loss	Precision	Recall	F1
Baseline	0.814	0.744	0.82	0.81	0.80	0.992	0.036	0.99	0.99	0.99
Baseline + 2048	0.833	0.663	0.84	0.83	0.83	0.991	0.036	0.99	0.99	0.99
Baseline + MobileV2	0.837	0.648	0.84	0.84	0.83	0.992	0.038	0.99	0.99	0.99
Baseline + Padding	0.802	0.803	0.80	0.80	0.79	0.989	0.046	0.99	0.99	0.99
Baseline + DataAug	0.913	0.336	0.92	0.91	0.91	0.996	0.032	1.00	1.00	1.00
Baseline + RandomErasing	0.900	0.397	0.90	0.90	0.89	0.995	0.027	0.99	1.00	0.99

Table 6.3: Classification Results for the neural networks built for the different experiments in both Market-1501 and HDA+ dataset.

In Table 6.2 and 6.3, the Classification Results for all datasets are shown. In this case, it can be seen that some of them are not successful, like the replacement of MobileNetV1 for MobileNetV2, the size of 2048 for the feature extractor and the Padding. However, there are others – like DataAug and RandomErasing - that seem to be promising since they increase classification values. From these results, one can expect that both DataAug and RandomErasing consist on a successful experiment in each dataset.

The ranking results for CUHK01 can be seen in Table 6.4. Beyond the values and the standard deviation already explained, this table also shows the difference in relation to the baseline, where the positive difference is represented in green, while the negative one is shown in red. For the first 3 experiments, the results obtained for mAP were not good. Nevertheless, some improvements were obtained for DataAug and RandomErasing. In addition to a large positive difference, this system is proven, by the Wilcoxon test, to be better than the baseline for the mAP evaluation.

The Ranking results for the CUHK02 dataset can be seen in Table 6.5. Similar to CUHK01, MobileNetV2 and Padding also show worse results. However, in this case, the 2048 shows little improve-

CUHK01 / Parameters	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Baseline	54.70	2.49	-	72.10	2.12	-	80.90	1.51	-	85.70	1.35	-	42.41	1.83	-
Baseline + 2048	48.70	2.49	-6.00	71.70	1.27	-0.40	81.90	1.92	1.00	88.20	1.47	2.50	39.18	1.60	-3.23
Baseline + MobileV2	46.60	3.90	-8.10	69.60	2.91	-2.50	80.50	1.80	-0.40	87.20	1.17	1.50	37.82	2.06	-4.59
Baseline + Padding	42.80	4.42	-11.90	62.60	2.84	-9.50	73.00	2.90	-7.90	81.20	3.19	-4.50	33.25	1.83	-9.16
Baseline + DataAug *	67.40	2.11	12.70	81.20	2.96	9.10	82.80	3.09	1.90	90.00	1.26	4.30	53.34	1.40	10.93
Baseline + RandomErasing *	58.50	2.84	3.80	77.10	2.21	5.00	84.40	1.28	3.50	90.30	1.19	4.60	47.95	1.78	5.54

Table 6.4: Ranking results for the Feature Extraction Network in CUHK01. Several experiments were made based on the baseline in order to improve it. All of them can be seen in the rows of the table.
* This shows that the null hypothesis that the system is not better than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

CUHK02 / Parameters	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Baseline	38.80	3.16	-	59.30	2.79	-	67.00	2.19	-	77.00	2.05	-	27.57	1.73	-
Baseline + 2048	41.10	2.39	2.30	56.70	2.00	-2.60	64.40	1.69	-2.60	72.20	1.66	-4.80	28.17	1.09	0.61
Baseline + MobileV2	32.30	2.49	-6.50	56.20	3.68	-3.10	61.30	3.47	-5.70	68.50	3.35	-8.50	25.93	1.71	-1.64
Baseline + Padding	26.30	2.49	-12.50	49.80	0.98	-9.50	60.20	1.99	-6.80	68.30	2.53	-8.70	21.85	1.05	-5.72
Baseline + DataAug *	55.70	2.37	16.90	74.70	1.62	15.40	81.60	2.37	14.60	88.30	2.19	11.30	43.99	1.53	16.43
Baseline + RandomErasing *	42.40	2.37	3.60	64.90	2.51	5.60	74.90	2.12	7.90	83.90	1.70	6.90	33.87	1.39	6.30

Table 6.5: Ranking results for the Feature Extraction Network in CUHK02. Several experiments were made based on the baseline in order to improve it. All of them can be seen in the rows of the table.
* This shows that the null hypothesis that the system is not better than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

ment. Both DataAug and RandomErasing clearly remain the better systems.

Market-1501 / Parameters	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Baseline	45.80	5.42	-	70.00	2.90	-	78.00	2.14	-	86.40	2.06	-	41.85	3.14	-
Baseline + 2048	44.00	4.96	-1.80	65.80	5.21	-4.20	78.40	3.01	0.40	84.70	3.13	-1.70	41.01	3.04	-0.84
Baseline + MobileV2	46.70	3.55	0.90	74.00	4.56	4.00	83.00	3.87	5.00	87.90	3.27	1.50	43.80	2.18	1.95
Baseline + Padding	41.90	4.91	-3.90	67.20	3.79	-2.80	75.60	2.91	-2.40	85.10	3.18	-1.30	39.02	2.17	-2.83
Baseline + DataAug *	60.40	3.20	14.60	80.40	3.75	10.40	87.10	2.30	9.10	93.00	2.32	6.60	55.07	2.59	13.22
Baseline + RandomErasing *	57.80	4.94	12.00	81.00	2.93	11.00	88.20	2.04	10.20	94.50	1.50	8.10	54.97	3.30	13.13

Table 6.6: Ranking results for the Feature Extraction Network in Market. Several experiments were made based on the baseline in order to improve it. All of them can be seen in the rows of the table.
* This shows that the null hypothesis that the system is not better than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

The Ranking results for the Market-1501 dataset can be seen in Table 6.6. In this dataset, the 2048 and Padding experiments obtained worse results than the baseline. Also, MobileNetV2, DataAug and RandomErasing presented greater results. However, according to the Wilcoxon test, only the last two are significantly better systems.

The Ranking results for the HDA+ dataset can be seen in Table 6.7. When analysing the table, the same behaviour as the one verified in CUHK01 and CUHK02 can be seen. There is only an improvement in both data augmentation techniques, while other experiments do not show great improvements. Contrary to the other datasets, the increase in rank-k and mAP is smaller for the data augmentation

HDA+ / Parameters	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Baseline	56.97	3.78	-	67.01	4.15	-	75.46	5.50	-	79.70	6.50	-	53.66	2.76	-
Baseline + 2048	54.24	4.97	-2.72	68.79	3.33	1.78	73.64	3.33	-1.82	81.82	2.71	2.12	53.29	2.20	-0.38
Baseline + MobileV2	53.64	6.22	-3.33	63.94	3.70	-3.06	66.37	3.94	-9.09	72.12	3.53	-7.57	47.80	1.97	-5.86
Baseline + Padding	49.70	2.78	-7.27	60.00	4.24	-7.00	63.03	4.45	-12.42	74.55	3.88	-5.15	47.03	1.74	-6.63
Baseline + DataAug	58.49	4.70	1.52	70.31	4.24	3.30	76.37	4.02	0.91	81.21	3.78	1.51	55.11	1.81	1.45
Baseline + RandomErasing *	56.67	3.84	-0.30	71.52	4.11	4.51	74.85	5.60	-0.61	83.34	4.93	3.64	55.55	2.44	1.89

Table 6.7: Ranking results for the Feature Extraction Network in HDA+. Several experiments were made based on the baseline in order to improve it. All of them can be seen in the rows of the table.
* This shows that the null hypothesis that the system is worse than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

techniques.

After analysing all the tables for all datasets, one can conclude that Baseline + 2048, Baseline + MobileNetV2 and Baseline + Padding do not improve the results, contrary to what was expected. In contrast, the data augmentation techniques show an improvement in comparison to the baseline. Considering all the experiments, one can conclude that the data augmentation techniques should be applied to the baseline. In this way, the MobileNetV1 and 1024 extraction are kept since MobileNetV2 and 2048 extraction do not show any improvement. Also, the padding is not applied. The results of the baseline that encompasses the improvements discussed are presented in Table 6.8.

CUHK01	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Baseline	54.70	2.49	-	72.10	2.12	-	80.90	1.51	-	85.70	1.35	-	42.41	1.83	-
Improved Baseline *	73.40	1.36	18.70	90.00	1.34	17.90	92.90	0.83	12.00	94.00	1.55	8.30	59.49	1.13	17.08
CUHK02	**Ranking Results**														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Baseline	38.80	3.16	-	59.30	2.79	-	67.00	2.19	-	77.00	2.05	-	27.57	1.73	-
Improved Baseline *	59.00	2.10	20.20	78.80	1.94	19.50	85.20	1.47	18.20	92.00	1.10	15.00	47.57	0.78	20.00
Market-1501	**Ranking Results**														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Baseline	45.80	5.42	-	70.00	2.90	-	78.00	2.14	-	86.40	2.06	-	41.85	3.14	-
Improved Baseline *	67.60	6.04	21.80	87.10	3.62	17.10	92.40	2.06	14.40	97.00	1.48	10.60	61.69	3.89	19.85
HDA+	**Ranking Results**														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Baseline	56.97	3.78	-	67.01	4.15	-	75.46	5.50	-	79.70	6.50	-	53.66	2.76	-
Improved Baseline *	62.13	4.12	5.16	77.88	3.05	10.87	81.21	3.26	5.76	84.24	3.53	4.55	58.17	2.61	4.51

Table 6.8: Improved baseline with successful experiments. Data Augmentation techniques which show good results were added to the baseline obtaining the improved baseline.
* This shows that the null hypothesis that the system is worse than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

Changing the image size could have some impact in the results and this experiment is seen in Table 6.9. Three different sizes were tested: the 128×128 is the Improved Baseline already presented, 224×224 and 256×256 were the newly obtained results.

Image Size	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
CUHK01															
128x128 [3]	73.40	1.36	-	90.00	1.34	-	92.90	0.83	-	94.00	1.55	-	59.49	1.83	-
224x224 [1]	83.20	1.40	9.80	94.00	1.25	4.00	96.43	1.38	3.53	98.43	0.96	4.43	67.81	1.36	8.33
256x256 [2]	76.53	1.86	3.13	88.90	1.30	-1.10	94.33	1.45	1.43	95.90	1.16	1.90	65.09	1.46	5.61
CUHK02															
128x128 [3]	59.00	2.10	-	78.80	1.94	-	85.20	1.47	-	92.00	1.10	-	47.57	0.78	-
224x224 [1]	63.90	2.07	4.90	85.20	2.23	6.40	90.70	1.49	5.50	93.70	1.10	1.70	55.37	1.65	7.80
256x256 [2]	62.30	2.79	3.30	85.60	1.96	6.80	91.40	1.28	6.20	97.10	0.70	5.10	52.06	0.88	4.49
Market-1501															
128x128 [3]	67.60	6.04	-	87.10	3.62	-	92.40	2.06	-	97.00	1.48	-	61.69	1.83	-
224x224 [1]	68.70	3.95	1.10	89.10	3.41	2.00	95.00	2.76	2.60	96.63	1.92	-0.37	63.73	2.69	2.03
256x256 [2]	69.37	3.31	1.77	87.83	3.34	0.73	92.90	2.43	0.50	96.13	1.87	-0.87	64.14	2.82	2.45
HDA+															
128x128 [1]	66.67	3.32	-	75.76	5.25	-	80.61	4.53	-	88.40	4.41	-	61.55	1.83	-
224x224 [3]	62.12	4.12	-4.55	70.61	5.76	-5.15	74.55	5.62	-6.06	81.82	4.69	-6.58	58.07	2.51	-3.48
256x256 [2]	63.04	5.73	-3.63	72.43	4.78	-3.34	74.24	5.78	-6.37	83.94	4.08	-4.45	58.28	2.37	-3.27

Table 6.9: Baseline improvement based on size. Different sizes were tested as the image input for the network. $^{(1)(2)(3)}$ The number corresponds to the position of the system among the three shown. Where 1 corresponds to the best and 3 to the worst. The comparison is made using the Wilcoxon test for mAP metric.

Overall, increasing the size of the input images means better results. For CUHK01, the 224×224 is clearly the best size among the three. As for rank-1 accuracy and mAP, the improvement is as much as 9%. For CUHK02 the results are similar: the size of 224×224 is the best one. This can be due to the fact that resizing the image from 60×160 to 128×128 can result in loss of information as not all pixels are represented. However, for Market-1501, the improvement is not that large since there is only a 2% increase in the different fields. In this case there is no loss of information when resizing the original image as its size is 64×128. For HDA+, there is no improvement when the image size is different which goes against what was analysed for the other datasets. However, in this dataset all images have different sizes, which may imply in loss of information when resizing them, and therefore, worst results.

When finishing all the desired experiments, the desired Feature Extractor is finally obtained for each dataset. For CUHK01 and CUHK02, the best size for the feature extractor it is undoubtedly the one presented in Table 6.9 with size 224×224. For the Market-1501, the size chosen was also 224×224. Although 256×256 shows a small improvement, it is not worth it, since it would require more parameters and a longer training time. Finally, for the HDA the size chosen was 224×224 to match the other datasets.

6.2 Matching Network

In this section, the results for the Matching Network, and more specifically the ones obtained when the matching network is trained with the Contrastive loss and Triplet Loss, are presented. At first, while addressing the problem, it was acknowledged that, even if results were improved when the matching network was trained, this improvement was not as good as previously expected. Thus, it was decided to retrain all layers that had been previously trained for classification. As a conclusion, the results obtained were better when all layers of the network were re-trained with Contrastive or Triplet Loss. The results followed the same procedure as mentioned in the previous chapter 5 and were obtained for 10 different galleries and queries. The results are from the implementation explained in section 5.3.

6.2.1 Contrastive Loss

For the CUHK01 dataset, the training is performed as explained: (i) 44318 pairs are created; (ii) based on them, the entire network is re-trained until achieving convergence, which takes approximately 6 hours and 200 epochs. The loss used was the Contrastive Loss, the optimiser was Stochastic Gradient Descent (SGD) with batch size of 32, learning rate of 10^{-4} and learning rate decay of 0.1 every 10000 batches. The Matching Network results for CUHK01 dataset are presented in Table 6.10.

CUHK01	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Euclidean Distance	83.20	1.40	-	94.00	1.25	-	96.43	1.38	-	98.43	0.96	-	67.81	1.36	-
Contrastive Loss *	82.40	1.28	-0.80	94.10	0.83	0.10	95.80	0.75	-0.63	97.30	0.64	-1.13	72.56	0.86	4.75

Table 6.10: Results of the Re-Identification system for CUHK01.This system contains a Matching Network trained with Contrastive loss besides the already known Improved Baseline (Euclidean Distance).
* This shows that the null hypothesis that the system is not better than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

As it can be seen in table 6.10 , the Matching Network shows some improvements. Although the rank accuracy decreases some percentage, the mAP value improves about 5%. This implies, that despite losing some positions for the first identification, at the beginning of the list, in some cases, the two images per query are better identified, which increases the mAP. In this way, one can say that the Matching Network is better than the Euclidean Distance confirmed by the Wilcoxon test for the mAP.

For the CUHK02 dataset, the training is done similarly to CUHK01: 91626 pairs are created and the all network is re-trained based on them. The Network was re-trained until convergence, which takes approximately 8 hours and 300 epochs; the loss used was the Contrastive Loss, the optimiser was Stochastic Gradient Descent (SGD) with batch size of 32, learning rate of 10^{-3} and learning rate decay of 0.1 every 10000 batches. The Matching Network results for CUHK02 dataset are presented in Table 6.11.

CUHK02	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Euclidean Distance	63.90	2.07	-	85.20	2.23	-	90.70	1.49	-	93.70	1.10	-	55.37	1.65	-
Contrastive Loss *	69.50	2.33	5.60	90.70	1.79	5.50	94.70	0.90	4.00	96.00	0.89	2.30	59.39	1.55	4.02

Table 6.11: Results of the Re-Identification system for CUHK02. This system contains a Matching Network trained with Contrastive loss besides the already known Improved Baseline (Euclidean Distance).
* This shows that the null hypothesis that the system is not better than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

As for CUHK02, the addition of the matching network implies an increase of almost 5% in the majority of the fields in interest. The rank-1 accuracy improves 5.60% with this addition. In this way, it can be said that the matching network largely improves the CUHK02 system, making it clearly better than the old one.

As for Market-1501, since there were more photos per id, a higher number of pairs was achieved (345078). Having a higher number of pairs leads to a longer training time. The model took 24h and 40 epochs to train until convergence. The loss used was the Contrastive Loss, the optimiser was Stochastic Gradient Descent (SGD) with batch size of 32, learning rate of 10^{-3} and learning rate decay of 0.1 every 10000 batches. The baseline results for the Market-1501 dataset are presented in Table 6.12.

Market-1501	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Euclidean Distance	68.70	3.95	-	89.10	3.41	-	95.00	2.76	-	96.63	1.92	-	63.73	2.69	-
Contrastive Loss *	73.40	4.57	4.70	92.50	2.33	3.40	95.70	1.79	0.70	97.30	1.19	0.67	70.04	3.03	6.31

Table 6.12: Results of the Re-Identification system for Market-1501. This system contains a Matching Network trained with Contrastive loss besides the already known Improved Baseline (Euclidean Distance).
* This shows that the null hypothesis that the system is not better than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

Regarding the Market-1501 dataset, the results achieved are promising, since there is a big increase in Rank-1 and mAP. Even if the increase in other fields is lower, the difference is still positive. In this way, once again, a better system with the addition of the matching network, is achieved.

Finally, for HDA+, as there are multiple images for a single id, only a few can be chosen for each one, otherwise a lot of time would be necessary until convergence, so 164688 pairs were created. The model took 8h and 100 epochs to train until convergence. The loss used was the Contrastive Loss, the optimiser was Stochastic Gradient Descent (SGD) with batch size of 32, learning rate of 10^{-4} and learning rate decay of 0.1 every 10000 batches. The baseline results for HDA+ dataset are presented in table 6.13.

The HDA+ is the dataset that shows the highest growth by reaching 10% in rank-1 accuracy and 5% in mAP. In this way, a promising system for this multi-shot dataset is achieved.

In general, all datasets show good improvements when this matching network is added and when

HDA+	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Euclidean Distance	62.12	4.12	-	70.61	5.76	-	74.55	5.62	-	81.82	4.69	-	58.07	2.51	-
Contrastive Loss *	73.03	3.70	10.91	81.82	3.50	11.21	86.37	2.79	11.82	95.15	2.78	13.33	62.22	1.92	4.15

Table 6.13: Results of the Re-Identification system for HDA+. This system contains a Matching Network trained with Contrastive loss besides the already known Improved Baseline (Euclidean Distance).
* This shows that the null hypothesis that the system is not better than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

the network is retrained with Contrastive Loss.

6.2.2 Triplet Loss

The results obtained for the triplet loss will be discussed. In order to train the network, it was necessary to make valid triplets, so the procedure discussed in section 5.3.2 was applied. In order to make all the triplets some computation time will be needed as a forward propagating is done to obtain each triplet loss. For CUHK01 it was necessary 3 hours to obtain the final triplets, for CUHK02 it was 8 hours, for Market-1501, being the biggest dataset, it took 20h considering that only an average of 10 images per people was chosen to make the triplets. Finally, when using HDA+, it took 3h.

At training time, for CUHK01, 992077 triplets are created, and the entire network is re-trained based on them until achieving convergence, which takes approximately 1.5 hours and 50 epochs. The loss used was the Triplet Loss, the optimiser was Adam with batch size of 16, learning rate of 10^{-6} and margin parameter of triplet loss being 0.5. The Matching Network results for CUHK01 dataset are presented in Table 6.14.

CUHK01	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Euclidean Distance	83.20	1.40	-	94.00	1.25	-	96.43	1.38	-	98.43	0.96	-	67.81	1.36	-
Contrastive Loss *	82.40	1.28	-0.80	94.10	0.83	0.10	95.80	0.75	-0.63	97.30	0.64	-1.13	72.56	0.86	4.75
Triplet Loss *	82.10	1.58	-1.10	93.80	1.17	-0.20	95.50	0.92	-0.93	97.10	0.70	-1.33	72.45	1.12	4.64

Table 6.14: Results of the Re-Identification system for CUHK01. This system contains a Matching Network trained with Triplet loss besides the already obtained results for Constrastive Loss and Improved Baseline (Euclidean Distance).
* This shows that the null hypothesis that the system is not better than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

Analysing the results of Table 6.14, it can be concluded that the behaviour of the triplet loss is similar to what happens for Contrastive loss. It is worst in all ranks, which can be translated as not finding the first match in the first positions of the query list. However, the mAP is higher which indicates that the second image per query is being returned in the first positions of the list, probably after the first image per query. It is not possible to extract a solid conclusion regarding the results obtained when comparing

both systems, since those results were very similar.

As for CUHK02, 1324176 triplets are created, and the entire network is re-trained based on them until achieving convergence, which takes approximately 2 hours and 20 epochs. The loss used was the Triplet Loss, the optimiser was Adam with batch size of 16, learning rate of 10^{-7} and margin parameter of triplet loss being 0.4. The Matching Network results for CUHK02 dataset are presented in Table 6.15.

CUHK02	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Euclidean Distance	63.90	2.07	-	85.20	2.23	-	90.70	1.49	-	93.70	1.10	-	55.37	1.65	-
Contrastive Loss *	69.50	2.33	5.60	90.70	1.79	5.50	94.70	0.90	4.00	96.00	0.89	2.30	59.39	1.55	4.02
Triplet Loss *	73.10	1.70	9.20	90.40	1.85	5.20	93.10	1.45	2.40	95.80	0.75	2.10	61.45	1.05	6.08

Table 6.15: Results of the Re-Identification system for CUHK02. This system contains a Matching Network trained with Triplet loss besides the already obtained results for Constrastive Loss and Improved Baseline (Euclidean Distance).
* This shows that the null hypothesis that the system is not better than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

By analysing the CUHK02 results it can be concluded that the triplet loss presents good results. They are higher compared to the feature extractor and some fields, like rank-1 and mAP, are higher compared to Contrastive Loss. This shows that triplet loss can have some impact in the results, achieving a better system than the one obtained for Contrastive Loss. The results may be different from the CUHK01 dataset as, in this case, there is more training data.

As for Market-1501, 8141090 triplets are created, and the entire network is re-trained based on them until achieving convergence, which takes approximately 3 hours and 20 epochs. The loss used was the Triplet Loss, the optimiser was Adam with batch size of 16, learning rate of 10^{-7} and margin parameter of triplet loss being 0.4. The Matching Network results for Market-1501 dataset are presented in Table 6.16.

Market-1501	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Euclidean Distance	68.70	3.95	-	89.10	3.41	-	95.00	2.76	-	96.63	1.92	-	63.73	2.69	-
Contrastive Loss *	73.40	4.57	4.70	92.50	2.33	3.40	95.70	1.79	0.70	97.30	1.19	0.67	70.04	3.03	6.31
Triplet Loss *	72.20	3.06	3.50	91.50	2.16	2.40	96.00	1.41	1.00	97.90	0.70	1.27	67.97	2.75	4.24

Table 6.16: Results of the Re-Identification system for Market-1501. This system contains a Matching Network trained with Triplet loss besides the already obtained results for Constrastive Loss and Improved Baseline (Euclidean Distance).
* This shows that the null hypothesis that the system is not better than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

The results obtained in Table 6.16 are not as expected. Since, although there is an improvement in relation to the baseline, there is no improvement in relation to Contrastive Loss, concluding that contrastive is better in this case. This is due to the fact that the triplets were not made in a more exhaustive way since computationally it would be more complicated to obtain them

As for HDA+, 151474 triplets are created, and the entire network is re-trained based on them until achieving convergence, which takes approximately 2 hours and 20 epochs. The loss used was the Triplet Loss, the optimiser was Adam with batch size of 16, learning rate of 10^{-6} and margin parameter of triplet loss being 0.5. The Matching Network results for HDA+ dataset are presented in table 6.17.

HDA+	Ranking Results														
	Rank-1			Rank-5			Rank-10			Rank-20			MAP		
	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Euclidean Distance	62.12	4.12	-	70.61	5.76	-	74.55	5.62	-	81.82	4.69	-	58.07	2.51	-
Contrastive Loss *	73.03	3.70	10.91	81.82	3.50	11.21	86.37	2.79	11.82	95.15	2.78	13.33	62.22	1.92	4.15
Triplet Loss *	74.55	2.78	12.42	84.55	3.70	13.94	86.97	3.60	12.42	90.91	3.78	9.09	68.12	1.25	10.05

Table 6.17: Results of the Re-Identification system for HDA+. This system contains a Matching Network trained with Triplet loss besides the already obtained results for Constrastive Loss and Improved Baseline (Euclidean Distance).
* This shows that the null hypothesis that the system is not better than the baseline is negative (for mAP metric) and can be rejected at a confidence level of 5%.

Discussing the results for HDA+, it can be concluded that there is an improvement when compared to the the feature extractor results. In addition, when comparing both losses, it can be seen that triplet loss presents better results in all field with the exception exception of the Rank-20. Improving the mAP results around 10% from the feature extractor and 6% from the Contrastive Loss shows that training with triplet loss is a big addition to this dataset.

In conclusion, triplet loss training can improve the overall results in all datasets and, in some cases, it also proves to be better than contrastive loss.

6.3 Generalisation across datasets

In this chapter, the adaptation of a trained system for a specific dataset will be tested on different ones to understand the adaptability of the built systems to different scenarios.

In this way, it was chosen to leave the CUHK02 dataset outside of this experiment as it is very similar to the CUHK01 dataset and this last one was sufficient to reach conclusions.

Taking this into account, each system already obtained in 6.2 will be used. Those systems were trained for classification and fine-tuned with the matching network added for each dataset. Then, those will be tested on an unfamiliar dataset that was not seen by them.

Following what was just described, the system designed for CUHK01 dataset was tested on Market-1501 and HDA+. The results from these tests can be seen in table 6.18.

As it can be seen, the results obtained in table 6.18 are low, since rank-1 accuracy only reached 19% and mAP 14% for both datasets. Then, it can be concluded that the CUHK01 dataset is not good in terms of adaptability for different environments, since it is a small dataset and only has 4 images per person.

Trained on CUHK01	Tested on Market-1501					Tested on HDA+				
	Ranking Results					Ranking Results				
	Rank-1	Rank-5	Rank-10	Rank-20	MAP	Rank-1	Rank-5	Rank-10	Rank-20	MAP
	19.30	32.23	36.56	43.29	13.83	19.12	19.93	21.75	25.06	14.38

Table 6.18: Adaptation of the model trained for the CUHK01 dataset to Market-1501 and HDA+. The model was first traineed for the CUHK01 dataset and then tested for Market-1501 and HDA+.

The results shown in table 6.19 reflect the moment where Market-1501 was trained and tested for CUHK01 and HDA+.

Trained on Market-1501	Tested on CUHK01					Tested on HDA+				
	Ranking Results					Ranking Results				
	Rank-1	Rank-5	Rank-10	Rank-20	MAP	Rank-1	Rank-5	Rank-10	Rank-20	MAP
	54.17	64.07	69.14	74.51	35.29	22.37	24.44	24.98	27.57	15.48

Table 6.19: Adaptation of the model trained for the Market-1501 dataset to CUHK01 and HDA+. The model was first trained for the Market-1501 dataset and then tested for CUHK01 and HDA+.

Looking at the results, one may conclude that that for the CUHK01 dataset the results are not bad, reaching 54% of rank-1 accuracy and 74% for rank-20 which is promising, mainly because this system has never seen this dataset before. As for HDA+, the results are bad. From this, it can be concluded that the system trained on Market is a system that is adaptable to other single shot datasets, since it has a lot of images taken from different viewpoints.

Finally, the last system was trained for HDA+ and tested for CUHK01 and Market-1501. The results can be seen in 6.20.

Trained on HDA+	Tested on CUHK01					Tested on Market-1501				
	Ranking Results					Ranking Results				
	Rank-1	Rank-5	Rank-10	Rank-20	MAP	Rank-1	Rank-5	Rank-10	Rank-20	MAP
	39.33	52.67	56.33	62.00	24.47	16.10	26.20	31.40	36.80	12.48

Table 6.20: Adaptation of the model trained for the HDA+ dataset to CUHK01 and Market-1501. The model was first trained for the HDA+ dataset and then tested for CUHK01 and Market-1501.

The results obtained in table 6.20 are the worst adaptations obtained, since rank-1 and mAP fall short behind the results previously obtained for the same datasets, but trained in different scenarios. In this way, it can be concluded that HDA+ is not a good dataset in terms of adaptability.

6.4 Comparison with State-of-the-Art

In this section, the final results of this dissertation will be presented as a proposed model for a Re-Identification system. In addition, the proposed model will be compared against state-of-the-art systems

in each evaluated dataset. This will allow to verify if the proposed model is competitive.

For the CUHK01 dataset, the big majority of state-of-the-art papers analysed only evaluate its performance in rank accuracy and the mAP value is not referred. That is why, in Table 6.21, it is not presented any value for mAP, with the exception of the proposed model.

CUHK01	Ranking Results				
	Rank-1	**Rank-5**	**Rank-10**	**Rank-20**	**MAP**
Proposed Model	82.40	94.10	95.80	97.30	**72.56**
FPNN (2014) [27]	27.87	64.00	75.00	87.00	-
mFilter (2014) [32]	34.30	55.00	65.30	-	-
MTDnet (2016) [77]	78.50	96.50	97.50	-	-
PersonNet (2016) [42]	71.14	90.07	95.00	98.06	-
JLML (2017) [78]	87.00	**97.20**	**98.60**	**99.40**	-
GOG-NFST_exp (2019) [62]	55.60	77.70	84.80	-	-
MuDeep (2019) [47]	**87.55**	96.63	98.38	-	-

Table 6.21: Comparison of state-of-the-art model against the proposed model for the CUHK01 dataset.

In Table 6.21, the results of the rank accuracy can be seen for different state-of-the-art systems. FPNN and mFilter are the only methods not based on deep-learning, and present a much inferior performance. In what regards other results concerning deep systems, *MuDeep* shows great results, achieving 87.55% for rank-1 accuracy. Comparing the results, one can see that the proposed model presents competitive results almost reaching the best ones. However, it must be taken into account the number of parameters of each network. As for the proposed model, only 5 M parameters are needed against the 25M for the MuDeep system which is a big difference. In the literature, the other state-of-the-art systems do not present the number of parameters, so a fair comparison cannot be made.

There are not many works that have used the CUHK02 dataset. However, this dataset is useful to assess how the re-id system behaves when there are a lot of different ids but not too much training data. As it can be seen in Table 6.22, the Proposed Model is better when compared to the one presented in [62] even though it has fewer parameters in the network.

Regarding the HDA+ dataset, there are not state-of-the-art papers that follow the explained procedure for retrieving the dataset images. So, the results presented in Table 6.23 are proposed as a benchmark for this dataset.

Market-1501 is probably the most widely used dataset in Re-ID nowadays. Several works report their results thoroughly, so, for this dataset, we can deepen our analysis. One column was added to

CUHK02	Ranking Results				
	Rank-1	**Rank-5**	**Rank-10**	**Rank-20**	**MAP**
Proposed Model	**69.50**	**90.70**	**94.70**	**96.00**	**59.39**
GOG-NFST_exp (2019) [62]	57.90	79.30	85.70	-	-

Table 6.22: Comparison of state-of-the-art model against the proposed model for the CUHK02 dataset.

HDA+	Ranking Results				
	Rank-1	**Rank-5**	**Rank-10**	**Rank-20**	**MAP**
Proposed Model	73.03	81.82	86.37	95.15	62.22

Table 6.23: Proposed model results for the HDA+ dataset.

the standard results Table, as it can be seen in Table 6.24, regarding backbone networks used in each paper.

Market-1501	Ranking Results					Backbone
	Rank-1	**Rank-5**	**Rank-10**	**Rank-20**	**MAP**	
Proposed Model	73.40	92.50	95.70	**97.30**	70.04	MobileNetV1
TriNet (2017) [75]	84.92	94.21	-	-	69.14	ResNet-50
JLML (2017) [78]	85.10	-	-	-	65.50	JLML-ResNet39
PCB (2018) [79]	92.30	97.20	**98.20**	-	77.40	ResNet-50
SGGNN (2018) [80]	92.30	96.10	97.40	-	82.80	ResNet-50
MG-CAM (2018) [81]	83.30	-	-	-	74.30	ResNet-50
LocalCNN (MG) (2018) [82]	95.90	-	-	-	91.50	ResNet-152
BoT Baseline (2019) [83]	95.43	-	-	-	85.90	ResNet-50
VA-ReID (2019) [84]	**96.23**	98.69	-	-	**91.70**	SEResNext
Pyramid (2020) [85]	96.10	**98.70**	-	-	89.00	ResNet-50

Table 6.24: Comparison of state-of-the-art models against the proposed model for the Market-1501 dataset. A new column was added regarding the backbone networks used in each paper.

When analysing Table 6.24, the best model is VA-ReID [84] with a 91.70% for mAP result. The proposed model cannot follow the other models in terms of accuracy. However, in terms of mAP, it is competitive when compared to the models of 2017. In fact, more recent models have a number of parameters higher than the proposed model. In Fig. 6.1, a comparison of mAP results and the number of parameter used in each system can be seen. This figure emphasises the fact that, despite the proposed model not having high mAP results as others, it is the one that uses fewer parameters - around a quarter

of the majority. This is important since this model takes less training time and can be deployed in mobile devices.

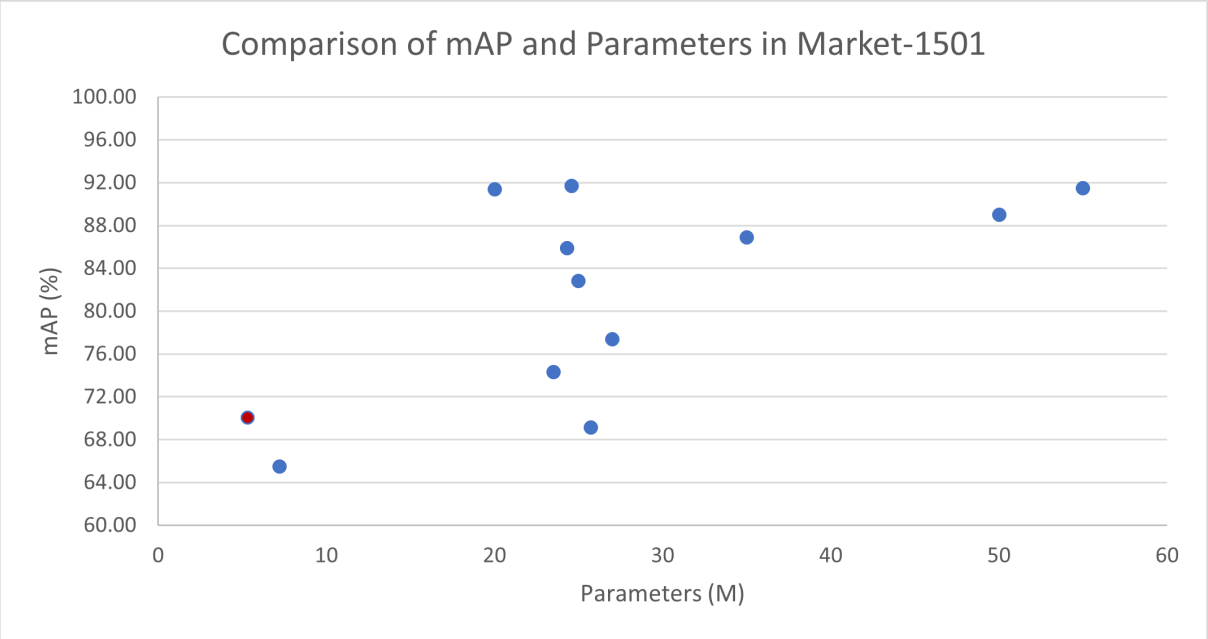


Figure 6.1: Comparison of mAP results against the number of parameters in the systems presented in Table 6.24. In red, it can be seen the proposed model position. The number of parameters for some of the networks was approximated.

7

Conclusions

Contents

We have proposed an effective re-id system based on a MobileNetV1 backbone for a Re-Identification system and a similarity matching network block trained with Contrastive and Triplet Loss. The system was validated in 4 different datasets.

The Pre-processing Block was able to standardise all images and to perform data augmentation which proved to be beneficial for the datasets presented. Beyond that, resizing those images also proved to be a crucial process since it contributed for the improvement of current results in some datasets.

The Feature Extractor Block reached one of the main objectives of this work: to extract good feature vectors. Despite using a lightweight network, MobileNet, it was possible to obtain good results using only the feature extractor in combination with Euclidean Distance.

The Similarity Matching Network was able to split even more the different classes, using two types of Losses - the Contrastive and Triplet Loss. First of all, the combination of the classification training of the feature extractor and, then, the fine-tuning of the added matching network for all layers of the system show a good result. This proved to be an efficient procedure in order to achieve a better system.

Looking specifically at the loss functions, it can be concluded that the Contrastive Loss is an effective implementation, since, for the majority of the datasets, the results improved by 5%. As for the Triplet Loss, it presents even better results than the Contrastive Loss in some datasets. In conclusion, the similarity matching network has fulfilled the objective set at the beginning of this dissertation.

All the objectives we intended to accomplish were reached. A good Re-Identification system was built and it can reach 70% or more for rank-1 accuracy, almost in all evaluated datasets. In further ranks, beginning at 5, the accuracy is around 90%, which states that, in 5 images displayed, the probability of obtaining the desired one is very high and promising. In conclusion, even if the results proved to be good, there is margin for improvement allowing the proposed model to be more competitive with the best state-of-the-art models. In this way, section 7.1 will describe which improvements can be performed to the work at hands.

7.1 Future work

In order to improve the results, and to achieve an even more competitive system for Re-Identification, different strategies can be applied to this work.

Some techniques that can be implemented are: (i) to add attention based systems that analyse the person image, in a specific way, and that can, then, achieve better feature vectors; (ii) to use local features that divide each person into different parts, which may lead to a better analysis than when the person as a whole is analysed; (iii) to continue the loss study, but using the quadruplet loss, this is, to use 4 images to train the networks instead of the three or two used in triplet and contrastive loss; (iv) to combine this network with a real-time system, where time is a variable, since, in the short-term,

people will not be able to appear in two distant cameras at the same time and the results obtained would not reflect this reality. Beyond the work developed, testing this system in other datasets could also be interesting to check whether the system maintained these results. Also, another option is to use lightweight networks other than MobileNet.

Bibliography

[1] A. Plantinga, “Things and persons,” in *The Review of Metaphysics*, 1961, pp. 493—-519.

[2] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. Hoi, “Deep learning for person re-identification: A survey and outlook,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[3] S. Karanam, Y. Li, and R. J. Radke, “Person re-identification with discriminatively trained viewpoint invariant dictionaries,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4516–4524.

[4] S. Bak, S. Zaidenberg, B. Boulay, and F. Bremond, “Improving person re-identification by viewpoint cues,” in *2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2014, pp. 175–180.

[5] Arc, “Convolution neural network,” 2018, last accessed 25 September 2021. [Online]. Available: https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05

[6] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European Conference on Computer Vision*. Springer, 2014, pp. 818–833.

[7] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in *International Conference on Artificial Neural Networks*. Springer, 2018, pp. 270–279.

[8] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning*. PMLR, 2015, pp. 448–456.

[9] P. Baldi and P. J. Sadowski, “Understanding dropout,” *Advances in Neural Information Processing Systems*, vol. 26, pp. 2814–2822, 2013.

[10] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[15] P. Marcelino, "Transfer learning from pre-trained models," 2018, last accessed 27 September 2021. [Online]. Available: https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751

[16] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems*, vol. 27, pp. 3320–3328, 2014.

[17] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[21] C.-F. Wang, "A basic introduction to separable convolutions," 2018, last accessed 27 September 2021. [Online]. Available: https://towardsdatascience.com/a-basic-introduction-to-separable-convolutions-b99ec3102728

[22] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016.

[23] T. Huang and S. Russell, “Object identification in a bayesian context,” in *IJCAI*, vol. 97. Citeseer, 1997, pp. 1276–1282.

[24] W. Zajdel, Z. Zivkovic, and B. J. Krose, “Keeping track of humans: Have i seen this person before?” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 2081–2086.

[25] N. Gheissari, T. B. Sebastian, and R. Hartley, “Person reidentification using spatiotemporal appearance,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 2. IEEE, 2006, pp. 1528–1535.

[26] L. Bazzani, M. Cristani, A. Perina, M. Farenzena, and V. Murino, “Multiple-shot person reidentification by hpe signature,” in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 1413–1416.

[27] W. Li, R. Zhao, T. Xiao, and X. Wang, “Deepreid: Deep filter pairing neural network for person reidentification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 152–159.

[28] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Deep metric learning for person re-identification,” in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 34–39.

[29] D. Gray and H. Tao, “Viewpoint invariant pedestrian recognition with an ensemble of localized features,” in *European Conference on Computer Vision*. Springer, 2008, pp. 262–275.

[30] A. Mignon and F. Jurie, “Pcca: A new approach for distance learning from sparse pairwise constraints,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2666–2672.

[31] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face recognition with local binary patterns,” in *European Conference on Computer Vision*. Springer, 2004, pp. 469–481.

[32] R. Zhao, W. Ouyang, and X. Wang, “Learning mid-level filters for person re-identification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 144–151.

[33] A. Das, A. Chakraborty, and A. K. Roy-Chowdhury, “Consistent re-identification in a camera network,” in *European Conference on Computer Vision*. Springer, 2014, pp. 330–345.

[34] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, “Person re-identification by local maximal occurrence representation and metric learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2197–2206.

[35] R. Layne, T. M. Hospedales, S. Gong, and Q. Mary, “Person re-identification by attributes.” in *Bmvc*, vol. 2, no. 3, 2012, p. 8.

[36] X. Liu, M. Song, Q. Zhao, D. Tao, C. Chen, and J. Bu, “Attribute-restricted latent topic model for person re-identification,” *Pattern recognition*, vol. 45, no. 12, pp. 4204–4213, 2012.

[37] Z. Shi, T. M. Hospedales, and T. Xiang, “Transferring a semantic representation for person re-identification and search,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4184–4193.

[38] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, “Large scale metric learning from equivalence constraints,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2288–2295.

[39] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification.” *Journal of Machine Learning Research*, vol. 10, no. 2, 2009.

[40] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, “Information-theoretic metric learning,” in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 209–216.

[41] Y. Zhang, B. Li, H. Lu, A. Irie, and X. Ruan, “Sample-specific svm learning for person re-identification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1278–1287.

[42] L. Wu, C. Shen, and A. v. d. Hengel, “Personnet: Person re-identification with deep convolutional neural networks,” *arXiv preprint arXiv:1601.07255*, 2016.

[43] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang, “A siamese long short-term memory architecture for human re-identification,” in *European Conference on Computer Vision*. Springer, 2016, pp. 135–153.

[44] R. R. Varior, M. Haloi, and G. Wang, “Gated siamese convolutional neural network architecture for human re-identification,” in *European Conference on Computer Vision*. Springer, 2016, pp. 791–808.

[45] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, “End-to-end comparative attention networks for person re-identification,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3492–3506, 2017.

[46] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, “Person re-identification by multi-channel parts-based cnn with improved triplet loss function,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1335–1344.

[47] X. Qian, Y. Fu, T. Xiang, Y.-G. Jiang, and X. Xue, "Leader-based multi-scale attention deep architecture for person re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 371–385, 2019.

[48] J. Almazan, B. Gajic, N. Murray, and D. Larlus, "Re-id done right: towards good practices for person re-identification," *arXiv preprint arXiv:1801.05339*, 2018.

[49] J. P. L. Mira, "Efficient deep learning method for person re-identification," *Instituto Superior Técnico*, no. February, 2021.

[50] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1318–1327.

[51] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 001–13 008.

[52] W.-S. Zheng, S. Gong, and T. Xiang, "Associating groups of people." in *BMVC*, vol. 2, no. 6, 2009, pp. 1–11.

[53] C. C. Loy, T. Xiang, and S. Gong, "Multi-camera activity correlation analysis," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1988–1995.

[54] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino, "Custom pictorial structures for re-identification." in *Bmvc*, vol. 1, no. 2. Citeseer, 2011, p. 6.

[55] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person re-identification by descriptive and discriminative classification," in *Scandinavian conference on Image analysis*. Springer, 2011, pp. 91–102.

[56] N. Martinel and C. Micheloni, "Re-identify people in wide area camera network," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition workshops*. IEEE, 2012, pp. 31–36.

[57] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 31–44.

[58] W. Li and X. Wang, "Locally aligned feature transforms across views," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3594–3601.

[59] P. M. Roth, M. Hirzer, M. Köstinger, C. Beleznai, and H. Bischof, "Mahalanobis distance learning for person re-identification," in *Person re-identification*. Springer, 2014, pp. 247–267.

[60] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1116–1124.

[61] M. Gou, S. Karanam, W. Liu, O. Camps, and R. J. Radke, "Dukemtmc4reid: A large-scale multi-camera person re-identification dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 10–19.

[62] M. Gou, Z. Wu, A. Rates-Borras, O. Camps, R. J. Radke *et al.*, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 3, pp. 523–536, 2018.

[63] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer gan to bridge domain gap for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 79–88.

[64] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by video ranking," in *European Conference on Computer Vision*. Springer, 2014, pp. 688–703.

[65] D. Figueira, M. Taiana, A. Nambiar, J. Nascimento, and A. Bernardino, "The hda+ data set for research on fully automated re-identification systems," in *European Conference on Computer Vision*. Springer, 2014, pp. 241–255.

[66] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification," in *European Conference on Computer Vision*. Springer, 2016, pp. 868–884.

[67] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Ouyang, and Y. Yang, "Exploit the unknown gradually: One-shot video-based person re-identification by stepwise learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5177–5186.

[68] M. Li, X. Zhu, and S. Gong, "Unsupervised person re-identification by deep learning tracklet association," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 737–753.

[69] G. Song, B. Leng, Y. Liu, C. Hetang, and S. Cai, "Region-based quality estimation network for large-scale person re-identification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[70] J. Li, J. Wang, Q. Tian, W. Gao, and S. Zhang, "Global-local temporal representations for video person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3958–3967.

[71] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2009.

[72] P. Dollár, R. Appel, S. Belongie, and P. Perona, “Fast feature pyramids for object detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[73] J. Wang, L. Perez *et al.*, “The effectiveness of data augmentation in image classification using deep learning,” *Convolutional Neural Networks Vis. Recognit*, vol. 11, pp. 1–8, 2017.

[74] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” *arXiv preprint arXiv:2004.11362*, 2020.

[75] A. Hermans, L. Beyer, and B. Leibe, “In defense of the triplet loss for person re-identification,” *arXiv preprint arXiv:1703.07737*, 2017.

[76] A. Field and G. Hole, *How to design and report experiments*. Sage, 2002.

[77] W. Chen, X. Chen, J. Zhang, and K. Huang, “A multi-task deep network for person re-identification,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[78] W. Li, X. Zhu, and S. Gong, “Person re-identification by deep joint learning of multi-loss classification,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 2194–2200.

[79] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, “Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline),” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 480–496.

[80] Y. Shen, H. Li, S. Yi, D. Chen, and X. Wang, “Person re-identification with deep similarity-guided graph neural network,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 486–504.

[81] C. Song, Y. Huang, W. Ouyang, and L. Wang, “Mask-guided contrastive attention model for person re-identification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1179–1188.

[82] J. Yang, X. Shen, X. Tian, H. Li, J. Huang, and X.-S. Hua, “Local convolutional neural networks for person re-identification,” in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1074–1082.

[83] W. Li, X. Zhu, and S. Gong, “Harmonious attention network for person re-identification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2285–2294.

[84] Z. Zhu, X. Jiang, F. Zheng, X. Guo, F. Huang, X. Sun, and W. Zheng, “Aware loss with angular regularization for person re-identification,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 13 114–13 121.

[85] G. Chen, T. Gu, J. Lu, J.-A. Bao, and J. Zhou, “Person re-identification via attention pyramid,” *IEEE Transactions on Image Processing*, vol. 30, pp. 7663–7676, 2021.

A

Combine Re-Identification system

The results in section 6.2 present an established end-to-end system that was trained in each dataset. However, one research question could be asked at this point: could the accuracy and results be improved if the feature extractor block, presented in section 5.2, was trained with the combination of all datasets?

To implement this research question, the training part of all datasets, with the exception of CUHK02 that was excluded due to its similarities with the CUHK01 dataset, was combined into a big one. This combination is shown in Figure A.1. The CUHK01 dataset is the only one that, before the combination, suffered data augmentation in order to be of the same size of the other datasets.

After the construction of this dataset, it is already possible to train the feature extractor network with the big dataset and to obtain a global feature extractor. Then, this feature extractor network can be tested specifically for each dataset.

In addition to the feature extractor, a matching network trained with Contrastive Loss can be added. This time, the fine-tuning was implemented for each dataset so that better results could be achieved.

The results obtained for this research question are presented in Table A.1. This table is divided into three sub-tables that represent, respectively, one dataset: CUHK01, Market-1501 and HDA+. Also, each dataset table is divided into two different parts: (i) the Feature Extraction that represents the network

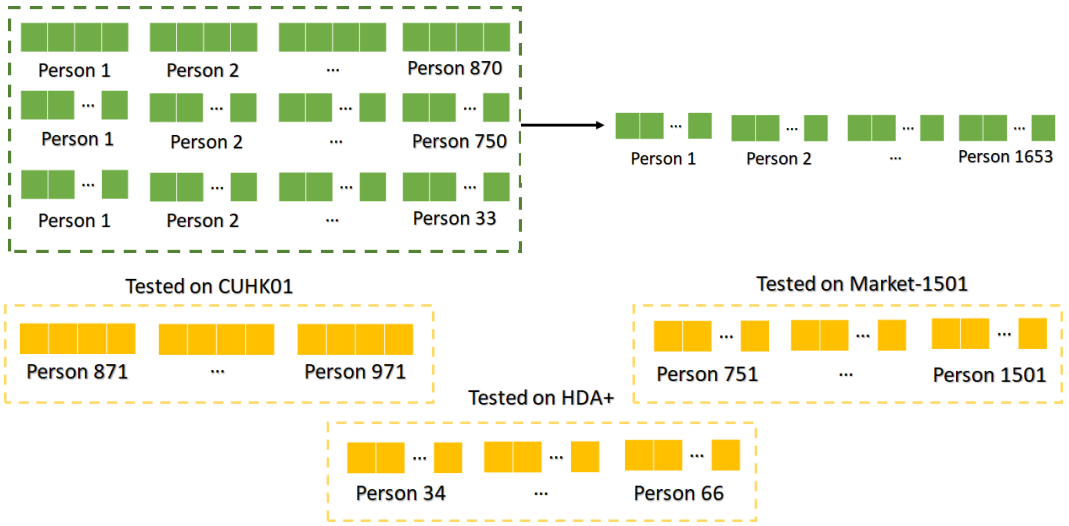


Figure A.1: Demonstration of the distribution of the dataset for a combined training. At the top, the combination of all datasets for training can be seen, where the green squares represent an image for training. This makes a dataset with 1653 people with different images for each one. At the bottom, the test part of the dataset is represented and each one is tested separately. The yellow squares represent the test image.

trained with the combined dataset and (ii) the Matching Network part in which the matching network is added and trained for a specific dataset. The results obtained for the Feature Extractor are compared against the results obtained for each dataset in section 6.1, while the results obtained for the Matching Network compare to the results obtained in each dataset, using a matching network, in section 6.2.

CUHK01		Ranking Results														
		Rank-1			Rank-5			Rank-10			Rank-20			MAP		
		Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Feature Extraction	Without extra training data	83.20	1.40	-	94.00	1.25	-	96.43	1.38	-	98.43	0.96	-	67.81	1.36	-
	With extra training data	77.60	1.80	-5.60	90.80	1.25	-3.20	94.20	0.87	-2.23	95.70	1.27	-2.73	65.94	1.62	-1.87
Matching Network	Without extra training data	82.40	1.28	-	94.10	0.83	-	95.80	0.75	-	97.30	0.64	-	72.56	0.86	-
	With extra training data	86.33	1.25	3.93	92.33	0.47	-1.77	93.67	0.47	-2.13	97.00	0.82	-0.30	73.42	0.18	0.86
Market		**Ranking Results**														
		Rank-1			Rank-5			Rank-10			Rank-20			MAP		
		Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Feature Extraction	Without extra training data	68.70	3.95	-	89.10	3.41	-	95.00	2.76	-	96.63	1.92	-	63.73	2.69	-
	With extra training data	71.20	6.10	2.50	87.70	3.90	-1.40	93.70	2.72	-2.00	96.80	1.99	0.17	62.76	4.37	-0.96
Matching Network	Without extra training data	73.40	4.57	-	92.50	2.33	-	95.70	1.79	-	97.30	1.19	-	70.04	3.03	-
	With extra training data	60.50	0.50	-12.90	85.00	0.00	-7.50	90.00	0.00	-5.70	96.00	0.00	-1.30	61.09	0.13	-8.95
HDA		**Ranking Results**														
		Rank-1			Rank-5			Rank-10			Rank-20			MAP		
		Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif	Value	SD	Dif
Feature Extraction	Without extra training data	62.12	4.12	-	70.61	5.76	-	74.55	5.62	-	81.82	4.69	-	58.07	2.51	-
	With extra training data	50.91	2.97	-11.22	59.40	5.62	-11.21	63.64	4.69	-10.91	74.55	3.88	-7.27	47.16	1.84	-10.91
Matching Network	Without extra training data	73.03	3.70	-	81.82	3.50	-	86.37	2.79	-	95.15	2.78	-	62.22	1.92	-
	With extra training data	66.67	0.00	-6.36	81.82	0.00	0.00	81.82	0.00	-4.55	84.85	0.00	-10.30	56.47	0.00	-5.75

Table A.1: Results for a combine system trained and fine tune for C1.

In a first analysis, and only considering the feature extraction part, it can be concluded that any dataset shows improvements. Even with the addition of more images and different environments, it seems that this training part did not help to identify good descriptors for comparing different people.

In what concerns the matching network part, results were expected to be better since the training was more specific for each dataset, this is, fine-tuned with the dataset to be tested. However, this is not the case and bad results were obtained for this experiment as well.

From those results, it is difficult to take a strong and consolidated conclusion, as the results are worse than the previously obtained ones. Coming back to the research question made, one may conclude that the answer is no: adding extra training data does not reflect positively in the results.

B

Graphical Representations of the results

Figures B.1 and B.2 show graphical representations of the work carried out by the built Re-Identification system. In those figures, it is possible to analyse two ranked lists from two different moments of this work. The first list (figure B.1) is from the final feature extractor presented in section 5.2. On the other hand, the second list (figure B.2) is from the system trained with triplet loss. The query and the gallery are the same in both figures and this experiment was performed for the CUHK02 dataset.

After analysing both figures, one may conclude that the triplet loss training improves the positions of the corresponding images of the query. As for the first system, the query images were returned in positions 4 and 6, while, in the second system, the query images are at the first and third position, which shows an improvement in relation to the feature extractor.

Finally, Figures B.3 and B.4 allow to visualise feature vectors in a high-dimensional vector space. Both representations were performed using t-Distributed Stochastic Neighbour Embedding (t-SNE) and the Market-1501 dataset.

Figure B.1: Re-Identification system results for the Euclidean Distance system. Here, it is presented a ranked list returned by the system and searching for a query (as shown on the left side). The list is ranked and the images corresponding to the query have a green frame.

Figure B.2: Re-Identification system results after the triplet loss training. Here, it is presented a ranked list returned by the system and searching for a query (as shown on the left side). The list is ranked and the images corresponding to the query have a green frame.

In Figure B.3, 50 ids of the training part of the Market-1501 dataset were chosen and were represented in 4 different phases of the project: (i) Without any training, (ii) using Euclidean Distance, (iii) using Contrastive Loss and (iv) using Triplet Loss. In figure B.4, 50 ids of the test part of the Market-1501 dataset were chosen and were represented in 4 different phases of the project: (i) Without any training, (ii) using Euclidean Distance, (iii) using Contrastive Loss and (iv) using Triplet Loss.

Analysing both figures, one may conclude that those are good representations since, at the bottom

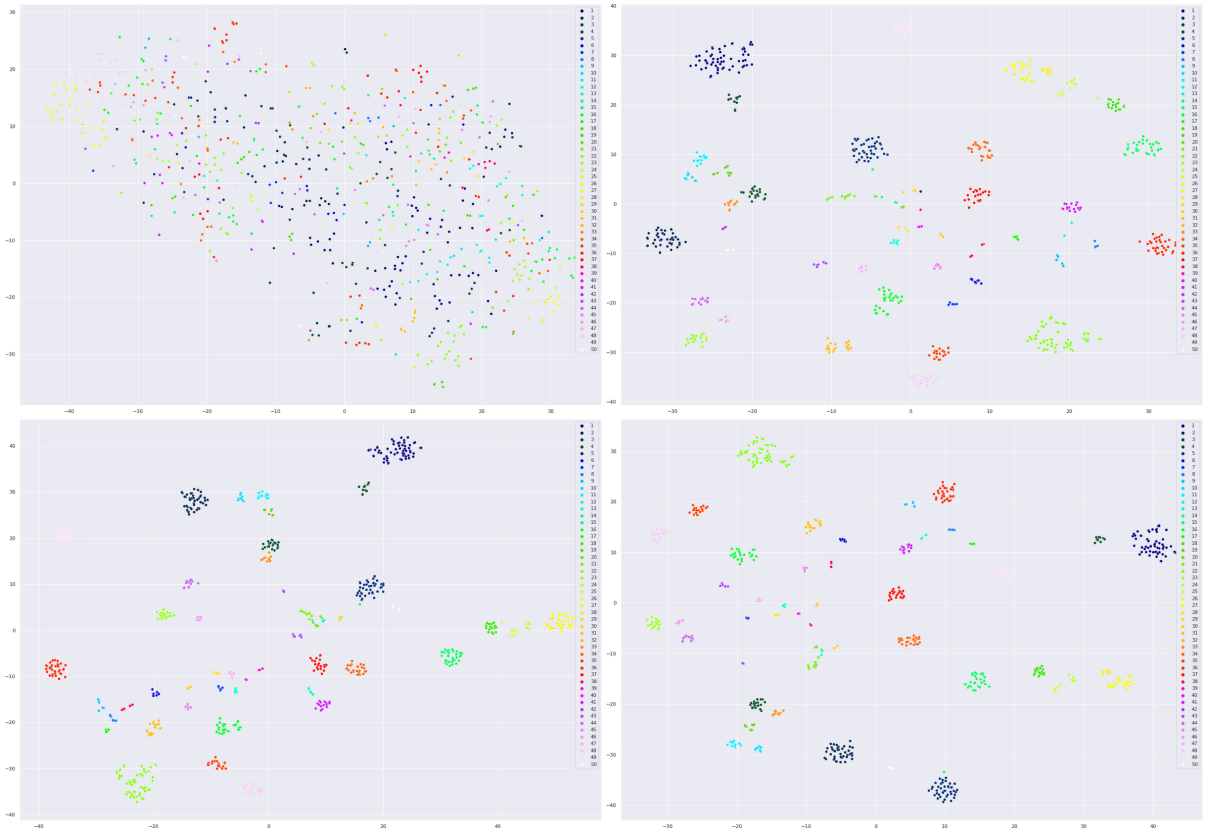


Figure B.3: Feature Vector representation based on t-SNE for 50 ids of the training Market-1501 dataset. Four different phases of the project are represented: (i) Without any training - Top Left, (ii) using Euclidean Distance - Top Right, (iii) using Contrastive Loss - Bottom Left and (iv) using Triplet Loss - Bottom Right.

part of each figure, different groups of classes can be seen. This means that the system was able to split the images of different classes and to bring closer the images of the same class. As expected, the representations results for the training are better than for the testing as better clusters were formed because, during the training part, the system had a direct contact with the images, and was able to learn how to distance one from another.

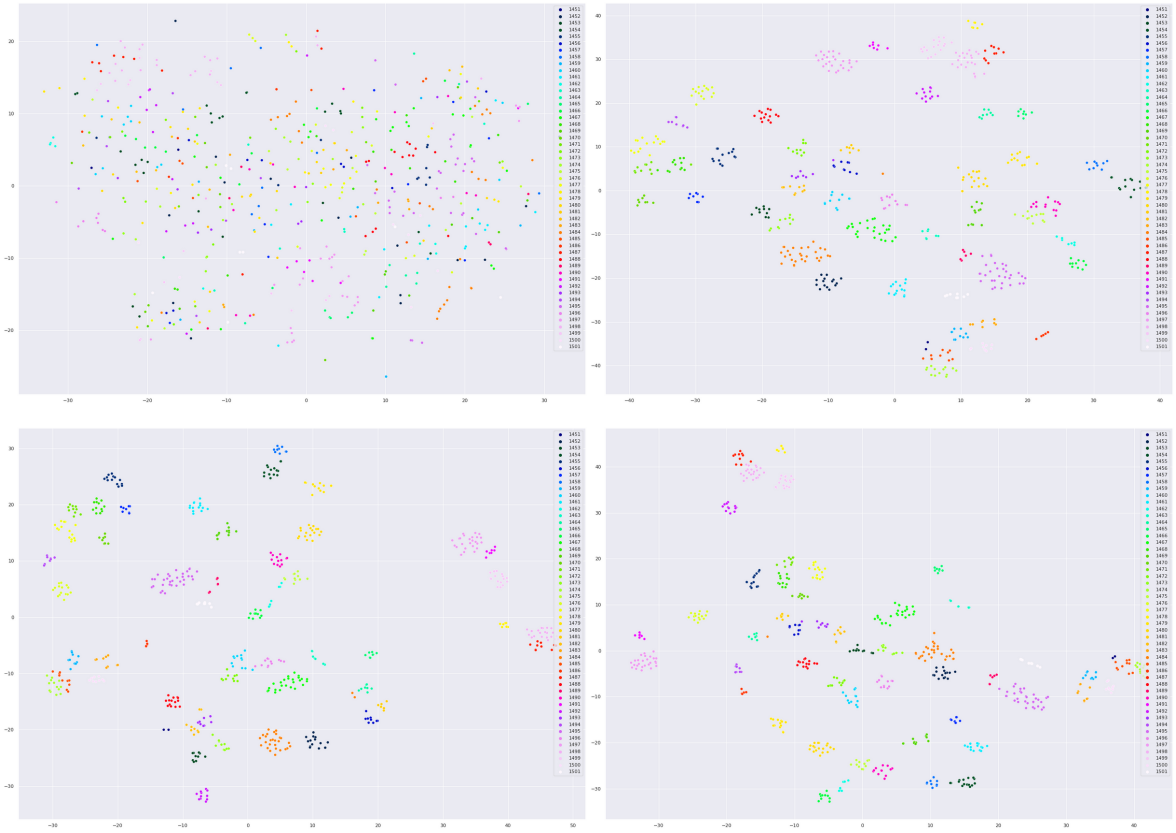


Figure B.4: Feature Vector representation based on t-SNE for 50 ids of the test Market-1501 dataset. Four different phases of the project are represented: (i) Without any training - Top Left, (ii) using Euclidean Distance - Top Right, (iii) using Contrastive Loss - Bottom Left and (iv) using Triplet Loss - Bottom Right.