Deep representations for similarity matching in Person Re-Identification

Francisco Gameiro Proença, *francisco.proenca@tecnico.ulisboa.pt*
Instituto Superior Técnico

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 paper 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.

***Index Terms*—Person Re-Identification; MobileNet; Deep Learning; Siamese Networks.**

I. Introduction

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 CCTV systems are being deployed across different places and countries, allowing for the identification and tracking of different people (e.g. a terrorist), actions (a robbery) and many other tasks of interest for society. Nowadays, the majority of these tasks rely on human work mostly consisting in identifying different people of interest and tracking them through the different cameras of the CCTV. However, as typical CCTV systems are composed by a large amount 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.

A. 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 divided into two tasks: (i) **Person detection** that corresponds to the detection of different people presented in the images. (ii) **Person Re-Identification** that 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 where the best matches will be at the top and the worst at the bottom.

The Re-Identification (Re-Id) problem poses several challenges. 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 are different viewpoints, changes in illumination, low-resolution images, occlusions or changing of clothes.

B. Objectives

The work of this paper aims to develop an efficient re-id pipeline close to state-of-the-art performance. We focus only on the person re-identification part and assuming a closed world scenario, i.e., that all queries are in the search gallery and single-shot (images) and multi-shot (video) datasets will be used. This system will face the challenges already presented in I-A through: **(i)** Development of a deep network that is able to extract good feature representations from different persons; **(ii)** Development of a good deep similarity matching network by comparing different ones trained with different losses; **(iii)** Study the deployment of a Re-Identification system to a scenario where it was not trained on (generalisation).

C. Outline

This paper is organised in six sections. In section 2 some background concepts on Deep learning for Re-Id are introduced and in section 3 the State of the Art is analysed. In section 4, the proposed approach is discussed followed by the implementation in section 5 and analysis of results in section 6. Finally, section 7 makes a brief conclusion of the work carried out and future improvements that can be made.

II. Background

A Siamese Neural Network is composed of two or more equal networks as they share the same configuration and weights. An example of it can be seen in Fig. 1. In this way, the same input produces the same output. This network goal 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, like: $d(x, y) = (\sum_{k=1}^{n} (y_k - x_k)^2)^{\frac{1}{2}}$ where x and y are the vectors, k their component index and n their length. If instead of using the Euclidean metric, one wants to learn a metric more suited to the dataset, losses like Contrastive Loss, as explained in II-A, and Triplet Loss, as explained in II-B, can be an option.

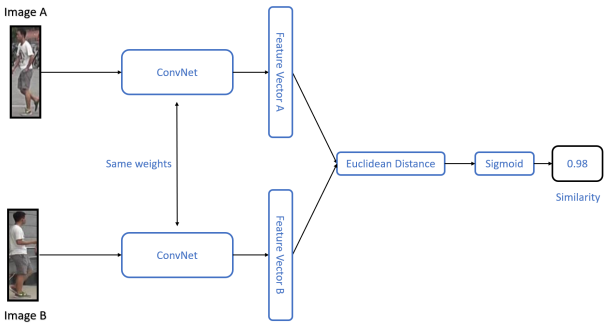

Fig. 1. Siamese Network example. The two Convolutional Network (ConvNet) are the same and have the same weight. The output will be the similarity between the input images.

A. Contrastive Loss

Contrastive Loss will receive two feature vectors as the input data. It trains the network aiming to obtain 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. In this case, we opted for an Euclidean distance metric. 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: $\mathcal{L} = Y * D_w^2 + (1 - Y) * \max(m - D_w, 0)^2$ - 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 to which different images must be pushed away. The $\max$ function chooses the largest number among 0 and the m minus D_w the distance. In Fig. 2, a demonstration of this contrastive loss can be seen.

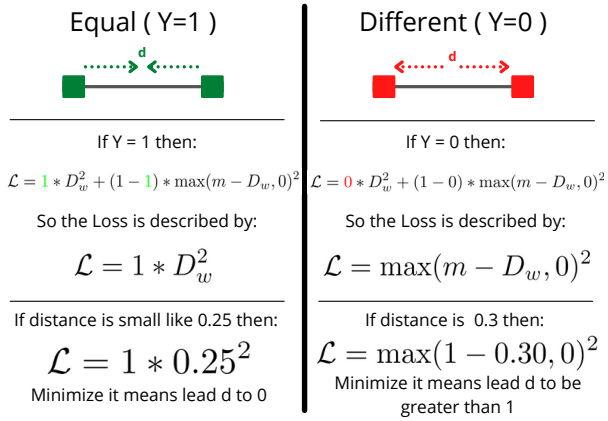

Fig. 2. Contrastive Loss explanation training in a Siamese Network.

B. Triplet Loss

Triplet Loss receives as input data three feature vectors distributed as: (i) an Anchor vector that is used as a point of comparison; (ii) a Positive vector that belongs to the same class as the Anchor; and (iii) a Negative vector that belongs to a different class than the Anchor. This Loss will have the objective of bringing the Anchor and Positive vector closer together while pushing away the Anchor and Negative vectors. In order to do this, the loss is formulated as: $L = \max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0)$, 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 that defines the distance to which different images must be pushed away. The $\max$ function chooses the largest number among two. Fig. 3 shows a simplified equation, where d_{AP} is the distance between Anchor and Positive Vectors and d_{AN} is the distance between Anchor and Negative Vectors. This brief graphical representation of triplet loss can be seen in Fig. 3.

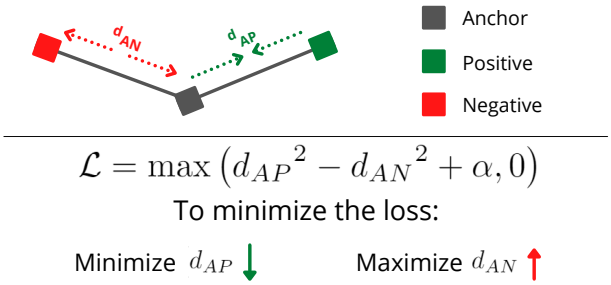

Fig. 3. Triplet Loss training explanation in a Siamese Network.

When one wants to train a network with triplet loss, the hardness of triplets is important. 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: $d_{AN} < d_{AP}$.

III. State of the Art

A. Hand Crafted Systems

In person feature extraction, the most used features are colour and texture. In [1], *Gray et al.* try to address the problem of the viewpoint of the camera as well as the pose of the person. Their algorithm uses different colour channels, texture histograms and several horizontal stripes that define a person allowing for the combination of simple features into a similarity function. That method is called Ensemble of Localised Feature (ELF). In [2] 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 [3], a person is separated into different parts (head, torso and legs) in which an HSV histogram is applied. Despite the good results presented by low level features, some work is being done in other areas, using attribute-based features[4]. Those are related to person characteristics often attributed by the human eye like gender or height, and are trained based on low level features. One of the approaches [4] consists on transfer learning, after learning different attributes in a photography dataset, into a re-id dataset. This type of features is gaining relevance and datasets related to it are also appearing.

However, all types of feature representation networks 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. The Euclidean distance is a particular case of it. This popular distance led to the Keep It Simple and Straightforward Metric (KISSME) [5] 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. Instead of focusing on distance metrics, there are some other works that try to learn subspaces. As an example, in [6], it is proposed that the model learns how to project into a low dimensional subspace with cross-view data solved in a similar manner to linear discriminant analysis. In this case, a distance metric is used in the resulting subspace. Finally, one of the methods presented in the literature dismisses metric learning, using techniques such as Support Vector Machine (SVM) [7] and Adaboost [1] instead, in order to correctly separate identities.

B. Deep Learning Systems

Deep Learning was successfully introduced in Re-Id by [8] and, since then, the number of publications in Re-Id using deep learning methods has been growing. There are two common techniques that are applied. The first method uses a CNN model for a classification purpose. Typically, this Convolutional Neural Network (CNN) is a state-of-the-art model that is already pre-trained on ImageNet [9] 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 where the objective is to put two images as an input and, in turn, the output will be the similarity between them. Different losses like Contrastive and Triplet loss can be implemented to improve results.

The first work that proposes to learn similarity metrics from the image pixels was presented by D Yi *et al.* [10]. 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). Different works were published trying to improve this Siamese network. Some tricks were added to the network itself, for instance, to improve the network by adding a gating function after each convolutional layer [11], or by adding an attention base model to retrieve better local features [12]. 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 [13], present the triplet loss training where the network has three images as an input. In that paper, the architecture is able to acquire both local and global features by using 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. One of the best performing systems using the Siamese Network is known as MuDeep [14] 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 [14], it combines both of these losses in order to train the network and uses both global and local features to classify each person. In [15], 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 [16], a similar method is followed. The author, firstly, trains the MobileNetV1 for a classification task and, after that, he takes the classification head in order to obtain the feature vector with the size of 1024×1. It then builds a similarity matching network that compares feature vectors and delivers the probability of being the same person.

In [17], 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, changing the position of some of the return results. In [18], 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.

C. Datasets

In order to train a Re-Identification system and then evaluate it, there are different public available datasets. In Table I, 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 Metrics used.

IV. Methodology

A. Overall Structure

The architecture chosen to address the problem is presented in Fig. 4 and will be explained in this section. In this system, it is worth mentioning three main processing blocks: **(i) Pre-Processing Block** 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 which contributes for increasing the amount of training data, since it creates new images from the already existing ones; **(ii) Feature Extractor**

TABLE I
RE-IDENTIFICATION DATASETS.

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

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; **(iii) Matching Network** whose task is to bring images of the same class closer together while pushing images from different classes further apart. Each block is described in this section.

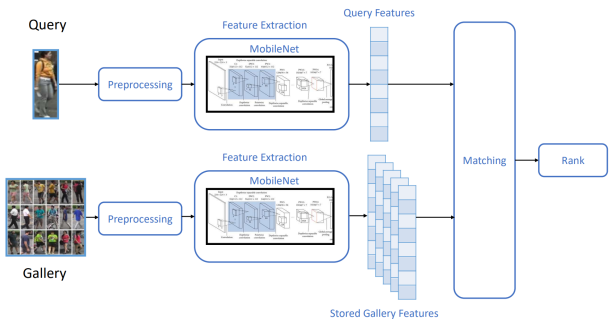

Fig. 4. Pipeline architecture of the Re-Identification system developed during this paper.

B. Pre-Processing

The Pre-Processing Block is the first one of the Re-identification system. This block is responsible for performing actions on the images, when loading them from different datasets. The actions performed are: **(i)** standardisation of images before they enter the CNN, which implies resizing all images via bi-linear interpolation to obtain a 128×128, and scaling the pixels between 1 and -1; **(ii)** standard data augmentation: Rotation, Zoom, Translation, Shear range, Horizontal flip and Brightness and **(iii)** the advanced data augmentation method: Random Erasing [18]. 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: 224×224 and 256×256.

C. 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. As a continuation of the work already carried out in [16], the MobileNetV1[19] was chosen as the CNN for the feature extractor, since it is a lightweight network which leads to less training and test time. On an attempt to improve [16], we tried to use MobileNetV2 [20] instead of MobileNetV1, on the expectation that the residual connections on MobileNetV2 could bring advantages. Since this was not the case, we opted to keep the MobileNetV1. These networks can receive different sizes, ranging from 96 to 256, so at the beginning 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). As was discussed in III-B , 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 [15]: 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. 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 dropout layer of the classification block were truncated, and the resulting head is shown in Fig. 5. This network is now used in a Siamese architecture to train the Re-Id task jointly with the matching network, to be presented in next section.

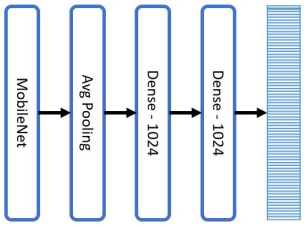

Fig. 5. The feature extractor starts with the MobileNet whose classification head was truncated and a new one was made. After MobileNet, an Average Pooling layer, two fully dense layers with 1024 neurons, a dropout layer established at 0.5 to produce the feature vector as shown.

D. Matching Network

The methods described in III-A 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 similarity matching networks, as referred to in III-B, that is, to use 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 based on a deep learning method to compute a similarity score between the two input patterns.

In order to define a baseline, this work starts by defining the similarity matching network as the simplest possible form: the Euclidean distance. A Siamese network was built, as shown in Fig. 6. 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. In this layer, the Euclidean 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 train this Siamese network based on the Contrastive loss [21] or Triplet Loss [22].

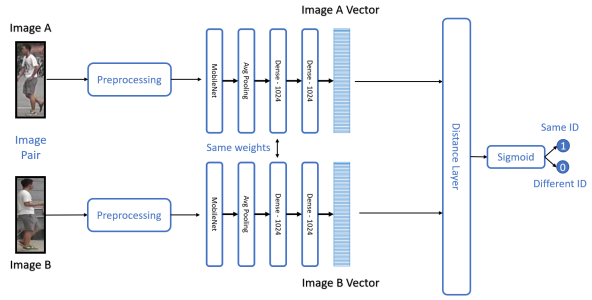

Fig. 6. An-end-to-end Re-Identification system is represented. It starts with two images at the network input being pre-processed. They then go through the feature extraction network where each one produces a vector. Finally, the vectors produced are compared with each other, using the Euclidean distance and going through a sigmoid. The result of 1 (belong to the same id) or 0 (do not belong to the same id) is shown.

Contrastive loss will allow the images to be distanced together or apart depending on their id, as explained in II-A. In this way, the metric learning can be trained, contrary to what was happening in Euclidean Distance (Fig. 6). The differences, in relation to Fig. 6, are a new batch normalisation layer and the loss. Triplet Loss will also allow to distance images together and apart, depending on their id, and at the same time, as explained in II-B. 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 consists in the normalisation of the feature vectors, calculating the distance between the anchor-positive and anchor-negative and consequently the loss itself. This will change some weight values that will allow to distance the classes apart and therefore improve the network.

E. Evaluation Metrics

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. When there is more than one image for a query presented in the gallery, the Rank-k accuracy is not the best metric to be used since it only reports the first appearance. In this way, mean Average Precision (mAP) should be used. This metric consists in calculating the mean Average Precision of all queries, as:

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

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 as:

$$AP = \frac{1}{m}\sum_{i=1}^{x} \left(Precision@i \times rel@i\right), \tag{2}$$

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. It is 1 if the sample is correct and 0 otherwise.

V. Implementation

A. 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 III-C, four were chosen due to their characteristics. These are: (i) **CUHK01 [23]** captured in the Chinese University of Hong Kong. It has a total size of 3884 images and 971 identities; (ii) **CUHK02 [24]** captured in the Chinese University of Hong Kong. It has 7264 images and 1816 identities; (iii) **Market-1501 [25]** captured in Tsinghua University. It has a total of 32668 images and 1501 identities; (iv) **HDA+ [26]** captured in a Portuguese University, Instituto Superior Técnico. HDA+ is a Multi-shot dataset that has a total of 16844 images from 66 different people. 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 for the classification task in each dataset can be seen in Fig. 7. For each dataset, the number of people is presented.

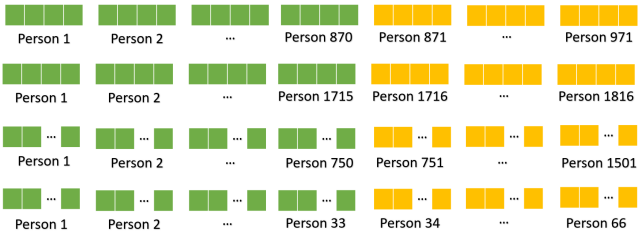

Fig. 7. Division of all datasets - CUHK01, CUHK02, Market-1501 and HDA+ respectively in figure from the top to the bottom - between training and testing. Each square represents an image. The green colour represents the training set. The yellow colour represents the test set.

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 as explained in IV-E.

To create the gallery and query sets the literature states that 1 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. That being the case, to evaluate this work, the sets are composed as follows: (i) **Gallery Set**: Select two images of each person present in the dataset and (ii) **Query Set**: Select 100 random people from the test dataset. In the case of HDA+, only 33 are selected.

Following these rules it will result in gallery sets with size of 1942 (CUHK01), 3632 (CUHK02), 3002 (Market-1501), 132 (HDA+). And query set with size of 100 (CUHK01), 100 (CUHK02), 100 (Market-1501) and 33 (HDA+). Each query will have two images of it present in the gallery, for a system to reach 100% mAP result the returned list ha6s to have the two images, of 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 [27] to verify if one method is better that another, with statistical significance.

B. Feature Extraction Analysis

In this part of the work, the focus will only be on developing a good and efficient feature representation network. 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 the ranking results. They consist in:

- **Baseline** (B) - In order to define the baseline for this work, the network previously developed in [16] 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 IV-C.
- **Baseline + 2048** (B+2048) - It consists on the same feature extractor as the baseline, as shown in Fig. 5, but instead of having dense layers with size of 1024 at the end of the network, it has layers with size of 2048.
- **Baseline + MobileNetV2** (B+V2) - It consists on the feature extractor structure discussed in IV-C but with a substitution of the backbone network to MobileNetV2.
- **Baseline + Padding** (B+P) - Instead of using linear interpolation, for resizing an image, the benefits of using a padding in images will be analysed.
- **Baseline + Data Augmentation** (B+DA) - Performing the first group of data augmentation techniques as stated in IV-B.
- **Baseline + Random Erasing** (B+RE)- Performing random erasing as stated in IV-B.

If the changes are good, a final feature extractor will be trained encompassing all techniques that improve the network. In addition, the resizing of the images will also be done where the 224×224 and 256×256 input image size will be tested.

Besides the division between training and testing, explained in V-A1, a division of the training partition between training and validation has to be done for training the feature extractor. In CUHK01 and CUHK02, this is very straightforward. For the four existing images of 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.

C. Deep Metric Learning

After the experiments in section V-B are completed, there will be a feature extraction network. In this way, with access to a good feature representation network, the construction of a similarity matching network, as explained in IV-D , can be done.

1) Contrastive Loss

In order to implement the contrastive loss, the procedure explained in II-A will be followed. This added contrastive block will be initialised with random weights and all the layers will be re-trained based on contrastive loss. In this case, pairs of images are used to train the network. 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 of this person can be created. For CUHK01 and CUHK02, some data augmentation will be performed to increase the number of images per person to a total of 8. As for Market and HDA+, some pairs will not be created as there is too many images per

person. In this way, CUHK01 has 44318 pairs, CUHK02 has 91626 pairs, Market has 345078 pairs and HDA+ has 164688 pairs.

2) 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 loss calculation, as explained in section II-B. 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. To train the network, only hard and semi-hard triplets where made in an offline manner (not during training). 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, as there are a lot of images per id and a total of approximately 11000 images, doing this procedure would imply a huge amount of time spent. Considering this, it was opted to choose a maximum of 8 images per id and the same procedure used for other datasets was replicated in this condition. For this implementation, data augmentation was not used for any dataset.

VI. Results

A. Feature Extraction

In this section, all the results related to the feature extractor experiments, discussed in V-B, will be shown and analysed. To obtain the results of this paper, a GPU GeForce GTX 1080 Ti was used. In order to obtain the baseline results (as discussed in V-B), the MobileNetV1 was fine-tuned for each dataset in 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 [16]. The baseline results expressed in Rank-k accuracy, k = 1, 5, 10, and mAP, can be seen in Table II, at the first row for each dataset. In this table, three fields can be seen: (i) Value field shows the mean value of the results for the 10 different queries and galleries; (ii) SD (Standard Deviation) field is the distribution around the mean of all 10 results obtained; (iii) difference in relation to the baseline, where the positive difference is represented in green, while the negative one is shown in red. In addition, * shows that the null hypothesis that the system (analysed in terms of mAP) is not better than the baseline is negative and can be rejected at a confidence level of 5% in each table.

After analysing all 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. Therefore, these additions are discarded. In contrast, the data augmentation techniques show an improvement in comparison to the baseline, so they will encompass the final feature extractor.

TABLE II
Ranking results for the Feature Extraction Network in all datasets.

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
B	54.70	2.49	-	72.10	2.12	-	80.90	1.51	-	85.70	1.35	-	42.41	1.83	-
B+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
B+V2	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
B+P	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
B+DA *	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
B+RE *	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
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
B	38.80	3.16	-	59.30	2.79	-	67.00	2.19	-	77.00	2.05	-	27.57	1.73	-
B+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
B+V2	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
B+P	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
B+DA *	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
B+RE *	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
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
B	45.80	5.42	-	70.00	2.90	-	78.00	2.14	-	86.40	2.06	-	41.85	3.14	-
B+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
B+V2	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
B+P	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
B+DA *	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
B+RE *	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
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
B	56.97	3.78	-	67.01	4.15	-	75.46	5.50	-	79.70	6.50	-	53.66	2.76	-
B+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
B+V2	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
B+P	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
B+DA *	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
B+RE *	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

Changing the image size could have some impact in the results and this experiment is reported in Table III. Three different sizes were tested: the 128×128 is the Improved Baseline, 224×224 and 256×256 were the newly obtained results. In this table, $^{(1)(2)(3)}$ correspond to the position of the system among the three shown (analysed in terms of mAP), where 1 corresponds to the best and 3 to the worst. The comparison is made using the Wilcoxon test.

Overall, increasing the size of the input images means better results. For CUHK01 and CUHK02, the 224×224 is clearly the best network presented among the three. As for rank-1 accuracy and mAP, the improvement is as much as 9%. 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 the original 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 loss of information when resizing them, and therefore, worst results.

At this point, all the desired experiments are concluded for each dataset and a feature extractor is obtained. For CUHK01 and CUHK02, the best size for the feature extractor is undoubtedly the one presented in Table III 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.

TABLE III
BASELINE IMPROVEMENT BASED ON SIZE.

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

B. Matching Network

In this section, the results of adding a similarity matching network and re-training the whole network with Contrastive or Triplet loss are shown and discussed. The results followed the same procedure as mentioned in the previous section V-C.

1) Contrastive Loss

Following the already explained procedure in V-C, each network was entirely re-trained until achieving convergence. The loss used was the Contrastive Loss, the optimiser was Stochastic Gradient Descent (SGD) with batch size of 32, the learning rate decay of 0.1 every 10000 batches and the learning rate depends on each dataset. For the CUHK01 dataset, 44318 pairs were created and the network was trained which took approximately 6 hours and 200 epochs with a learning rate of 10^{-4}. For CUHK02, the train was similar: 91626 pairs were created and the network took approximately 8 hours and 300 epochs to train with a learning rate of 10^{-3}. As for Market-1501, 345078 pairs were created and the model took 24h and 40 epochs to train until convergence with a learning rate of 10^{-3}. For HDA+, 164688 pairs were created and the model took 8h and 100 epochs to train until convergence with a learning rate of 10^{-4}.

As it can be seen in Table IV, the Matching Network shows some improvements. Although for CUHK01 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 since for this metric it is important to identify and put at the top of the list all images from the query, as explained in section IV-E. In this way, one can say that the Matching Network is better than the Feature Extractor confirmed by the Wilcoxon test for the mAP. As for CUHK02, the addition of the matching network implies an increase of almost 5% in the majority of the fields of interest. The rank-1 accuracy improves 5.60% with this addition. Regarding the Market-1501 dataset, the results achieved are promising, since there is a big increase

TABLE IV
RESULTS OF THE RE-IDENTIFICATION SYSTEM FOR ALL DATASETS TRAINED WITH CONTRASTIVE LOSS.

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	-
Matching Network *	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
CUHK02															
Euclidean Distance	63.90	2.07	-	85.20	2.23	-	90.70	1.49	-	93.70	1.10	-	55.37	1.65	-
Matching Network *	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
Market-1501															
Euclidean Distance	68.70	3.95	-	89.10	3.41	-	95.00	2.76	-	96.63	1.92	-	63.73	2.69	-
Matching Network *	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
HDA+															
Euclidean Distance	62.12	4.12	-	70.61	5.76	-	74.55	5.62	-	81.82	4.69	-	58.07	2.51	-
Matching Network *	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

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. The HDA+ is the dataset that shows the highest growth by reaching 10% in rank-1 accuracy and 5% in mAP. In general, all datasets show good improvements when this matching network is added and when the network is retrained with Contrastive Loss.

2) Triplet Loss

Following the already explained procedure in V-C, each network was entirely re-trained until achieving convergence. The loss used was the Triplet Loss, the optimiser was Adam with batch size of 16 and both learning rate and triplet margin depends on each dataset. For CUHK01 dataset, 992077 triplets were created and the network was trained which took approximately 1.5 hours and 50 epochs with a learning rate of 10^{-6} and a margin parameter of 0.5. For CUHK02, the train was similar: 1324176 triplets were created and the network took approximately 2 hours and 20 epochs to train with a learning rate of 10^{-7} with a margin parameter of 0.4. As for Market-1501, 8141090 were created and the model took 3h and 20 epochs to train until convergence with a learning rate of 10^{-7} and a margin parameter of 0.4. For HDA+, 151474 were created and the model took 1.5h and 20 epochs to train until convergence with a learning rate of 10^{-6} and a margin parameter of 0.5.

TABLE V
RESULTS OF THE RE-IDENTIFICATION SYSTEM FOR ALL DATASETS WITH TRIPLET LOSS.

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
CUHK02															
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
Market-1501															
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
HDA+															
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

As it can be seen in Table V, and for the majority of the datasets, the triplet loss results in an increased value for all fields evaluated. The major exception is CUHK01, whose results do not increase, probably due to fewer training data, in a similarl way to what happened for the Contrastive loss. As for other datasets, all of them show an improvement in relation to the baseline extractor and some fields, particularly rank-1 and mAP, show an improvement in relation to Contrastive Loss. However, for Market-1501, the triplet loss is not better than the Contrastive Loss. This can be due to not having made all triplets in a exhaustive manner, as it was done for other datasets, due to the lack of time. However, triplet loss training can improve the overall results in all datasets in respect to the baseline and, in some cases, it also proves to be better than contrastive loss.

C. Comparison with State-of-the-art

In this section, the final results of this paper 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, there are not many state-of-the-art papers that evaluate the performance in rank accuracy and also, the mAP value is not referred to. In this sense, Table VI, do not present any value for mAP, with the exception of the proposed model. In Table VI, the results of the rank accuracy can be seen for different state-of-the-art systems. FPNN and mFilter are the only methods that are not based on deep-learning and, therefore, they present a much inferior performance. As regarding to 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.

TABLE VI
COMPARISON OF STATE-OF-THE-ART MODEL AGAINST THE PROPOSED MODEL FOR THE CUHK01 DATASET.

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)	27.87	64.00	75.00	87.00	-
mFilter (2014)	34.30	55.00	65.30	-	-
MTDnet (2016)	78.50	96.50	97.50	-	-
PersonNet (2016)	71.14	90.07	95.00	98.06	-
JLML (2017)	87.00	**97.20**	**98.60**	**99.40**	-
GOG-NFST_exp (2019)	55.60	77.70	84.80	-	-
MuDeep (2019)	**87.55**	96.63	98.38	-	-

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 VII, the Proposed Model is better when compared to the one presented in [28] even though it has fewer parameters in the network.

TABLE VII
COMPARISON OF STATE-OF-THE-ART MODEL AGAINST THE PROPOSED MODEL FOR THE CUHK02 DATASET.

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)	57.90	79.30	85.70	-	-

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 VIII are proposed as a baseline for this dataset.

TABLE VIII
COMPARISON OF STATE-OF-THE-ART MODEL AGAINST THE PROPOSED MODEL FOR THE HDA+ 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

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

TABLE IX
COMPARISON OF STATE-OF-THE-ART MODEL AGAINST THE PROPOSED MODEL FOR THE MARKET-1501 DATASET.

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)	84.92	94.21	-	-	69.14	ResNet-50
JLML (2017)	85.10	-	-	-	65.50	JLML-ResNet39
PCB (2018)	92.30	97.20	**98.20**	-	77.40	ResNet-50
SGGNN (2018)	92.30	96.10	97.40	-	82.80	ResNet-50
MG-CAM (2018)	83.30	-	-	-	74.30	ResNet-50
LocalCNN (MG) (2018)	95.90	-	-	-	91.50	ResNet-152
BoT Baseline (2019)	95.43	-	-	-	85.90	ResNet-50
VA-ReID (2019)	**96.23**	98.69	-	-	**91.70**	SEResNext
Pyramid (2020)	96.10	**98.70**	-	-	89.00	ResNet-50

When analysing Table IX, the best model is VA-ReID [29] 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. 8, 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.

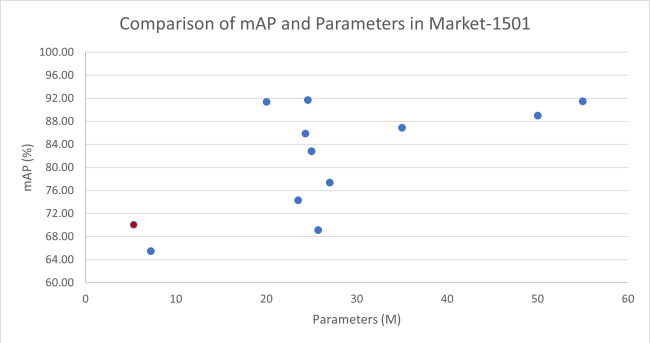

Fig. 8. Comparison of mAP results against the number of parameters in the systems presented in Table IX. In red, it can be seen the proposed model position. The number of parameters for some of the networks was approximated.

VII. Conclusion

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. Several alternative design choices were evaluated to achieve a model that is competitive with the state-of-the-art with a much smaller number of parameters, being thus suited for real-time applications.

Some techniques, to future work to be developed, 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. 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.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 152–159.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

[18] 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.

[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] 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.

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

[23] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *Asian conference on computer vision*. Springer, 2012, pp. 31–44.

[24] 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.

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.