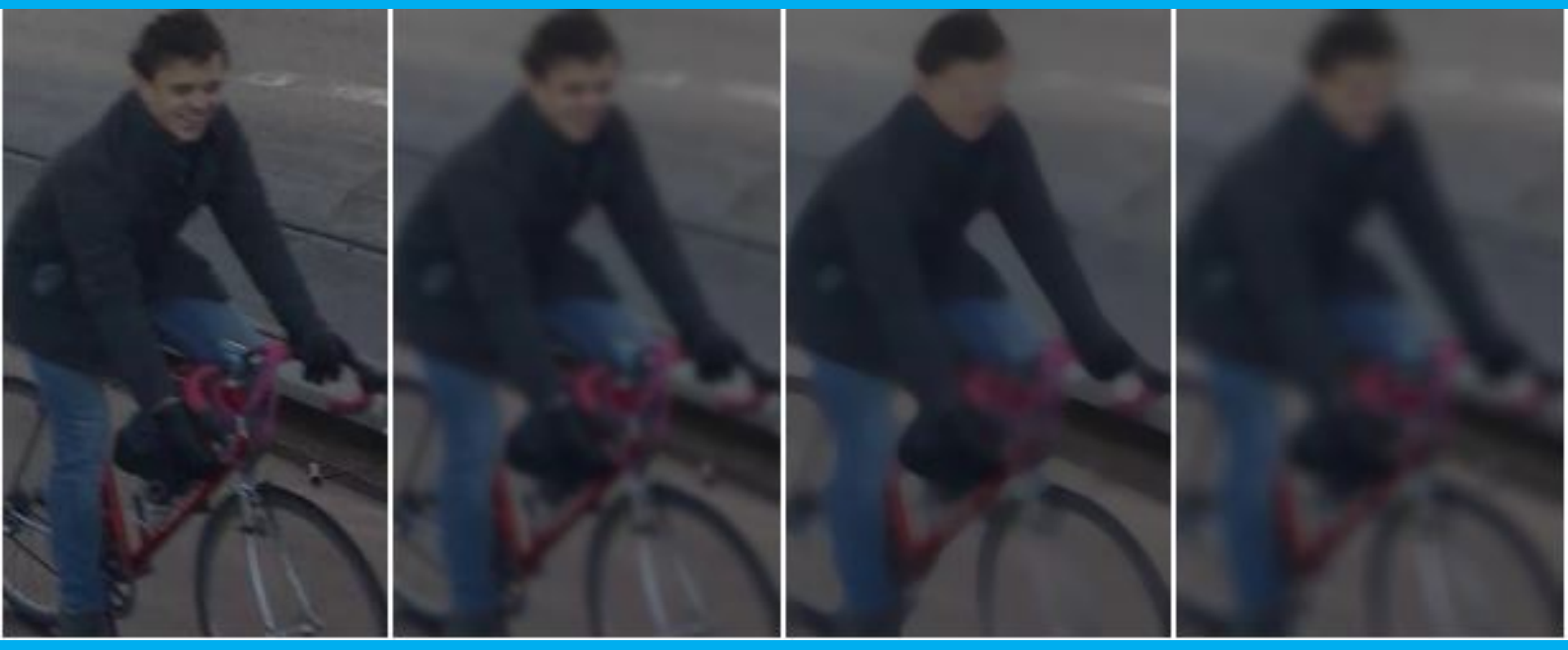
Anonymous Open-World Cyclist Re-Identification

M. Schoustra

Supervisors:
M. Dubbeldam
J.C.F. de Winter
Z. Xia
16-07-2020



TU Delft

Technolution

Anonymous Open-World Cyclist Re-Identification

by

M. Schoustra

to obtain the degree of Master of Science, in **Mechanical Engineering**,
at the Delft University of Technology,
to be defended publicly on Thursday July 16th, 2020 at 11:00 AM.

Student number:	4308611	
Project duration:	September 1, 2019 – July 16, 2020	
Thesis committee:	ir. M. Dubbeldam,	Technolution, Daily Supervisor
	dr. ir. J.C.F. de Winter ,	TU Delft, Supervisor
	ir. Z. Xia,	TU Delft, Supervisor
	dr. ir. J.F.P. Kooij	TU Delft, comittee member

TU Delft Delft University of Technology

Technolution

Preface

This thesis is the final part of my masters degree in Mechanical engineering at the Technical University of Delft. It serves as an overview of my research, performed the past year during my graduation internship at Technolution, which was from September 2019 until July 2020. In this research the possibilities of anonymous re-identification, and re-identification for a real world situation are explored.

I have always been a huge fan of artificial intelligence concepts applied to our everyday life and I am glad that I was able to work on a topic matching my interests. It was a true learning experience and i had fun exploring new concepts. A lot of hours went into this work, and I am proud to say that it feels like the 'kers op de taart'(Dutch saying for finishing touch) of my studying period.

For their help, i would like to thank the following people in particular: Michael Dubbeldam, thesis supervisor and senior architect at Technolution, for his advice and feedback. Also he supplied me with the topic and the possibility of carrying out my research at Technolution. I also would like to thank Zimin Xia and Joost de Winter, thesis supervisors, for their guidance during my thesis research. Also i would like to thank Technolution, for the possibility of working on my research project at their office and the free coffee.

Finally i would like to thank my girlfriend for her mental support and my friends for the many discussions on the artificial intelligence topic.

M. Schoustra
Delft, July 16, 2020

Contents

1 Introduction

Cycling is extremely popular in the Netherlands, according to an article from the Dutch Central Burea of Statistics(CBS), the Dutch citizens cycled a total of 14.75 billion kilometers[2], that could take us around the earth 369,000 times. Needless to say cycling is very popular in the Netherlands, however very few data exist on the flow and route choices of cyclist, especially when we compare it to regular vehicle traffic. That is where the FlowCube[29] comes into play.

The FlowCube is a product being developed by Technolution. The FlowCube is a single-box traffic sensor that aims to replace different conventional traffic monitoring systems (e.g. traffic radars, floating car data, induction loops), based on computer vision and edge AI. One of the initial use cases for FlowCube is measuring travel times and route selection of cyclists. There is currently no effective solution for measuring this. There is also a strong societal urge to stimulate the usage of bicycles (Climate change, less pollution). One of the purposes of this product is to map the route choices made by cyclists on the road. This information can be extracted by automatically matching the cyclist in different video streams. The task of finding the same person in multiple images or video resources is known as person re-identification.

This work is focused on improving the achieved cyclist re-identification score. To achieve this, first a dataset was created. Next the effect of anonymization on the re-identification performance was evaluated. This anonymization was subsequently applied to our own dataset and next several benchmark models were built and applied to our new dataset. Next some heuristics and a loss function from other deep learning fields were evaluated for person re-identification. These steps are described in the scientific paper. Some general background information is given in the deep learning section, the literature review is also included for the interested readers.

2

Scientific Paper

Anonymous Open-World Cyclist Re-Identification

Michael Schoustra
Technical University of Delft
m.schoustra@student.tudelft.nl

***Abstract*— This paper explores the topic of anonymous open-world cyclist re-identification. Person re-identification (re-ID) with deep neural networks has made progress and achieved high performance in recent years. However, most existing re-ID works are designed for closed-world scenarios rather than realistic open-world settings, which limits the practical application of the re-ID technique. Currently, no dataset of cyclists exists. Directly applying a trained re-ID network on another dataset does not yield good results. Therefore, a new dataset of cyclists is introduced in this paper. Our dataset is different than most existing benchmark datasets as every person in our dataset has been blurred to respect their privacy. In this paper the effect of blurring on re-identification performance is evaluated. To evaluate the impact of blurring on the re-identification performance we first tested it on the Market1501 dataset. Here, the performance of the blurred version could easily be compared to the original version blurring. The experiments show that blurring the data only impacts the rank-1, and mAP score by 1-4% for the Market1501 dataset. This impact depends on the size of the blurring window that is used. Several state of the art performing re-identification models were rebuilt and evaluated on our new dataset and their performance was compared. Furthermore, different backbone architectures were evaluated, we found that EfficientNetB0 outperforms the standard ResNet50 backbone architecture for re-identification, while using fewer parameters. Next the effects of RandAugment and Cosine learning rate decay were evaluated for re-identification. It was found that including RandAugment increases the rank-1 and mAP scores achieved on our dataset by up to 3%, and that using cosine decay further improves the achieved score. The final scores achieved on our dataset are 89.8% rank-1 accuracy and a mAP of 81.4%. Next we show that the batch hard pairwise loss function increases the F1-score by 7% for open-world re-identification. It was concluded that combining the embeddings is necessary to achieve good performance for open-world re-identification.**

I. Introduction

In recent years, person re-identification has become increasingly popular in the research community due to its application and significance. It is also a controversial topic together with facial recognition; an article published in the Financial times[1] showed that the datasets used for re-identification and facial recognition are often collected without the consent of the clearly recognizable people in the images. Even your face or mine could be in one of the datasets. As a result of this article, the Duke MTMC Re-ID[2] dataset was discontinued.

The re-identification task can be summarized as matching a pedestrian from an input image to a gallery set which has been captured by different cameras. Some challenges seen in the person re-identification task are the variations in lighting, view angle, pose of individuals, low resolution and (partial) occlusion of individuals[3]. These challenges are shown in Figure 1.

Figure 1: Challenges seen in re-identification. Images from the Market1501[4]. From left to right we have bad image quality, occlusion, change of viewpoint angle, change of person appearance, illumination changes and similar-looking individuals.[5]

Recent models have learned to deal with these challenges and are able to outperform humans on certain benchmark datasets. For example, on the Market1501 dataset[4], the state of the art achieves a rank-1 accuracy of 94.5% while the human level accuracy was tested at 93.5%[3]. This implies that the re-identification models are ready to take a step towards the more challenging issue of open-world re-identification. In an open-world situation, captured images from pedestrians do not have to be in the gallery set and the gallery set should contain lots of irrelevant images. This means the model cannot simply return the best match from the gallery; it must also have the ability to return that no match was found. Current benchmark datasets[2, 4, 6, 7] do not provide a benchmark method to evaluate the effectiveness of an open-world model. Also, the standard metrics used for person re-ID, the cumulative matching curve (CMC), and the mean average precision (mAP) cannot be used for an open-world evaluation as not every input has a match in the gallery set.

There are only very few researchers who reported their open-world scores[8, 9]. The concept was first introduced by Gong et al.[10]. The researchers created a gallery set consisting of targets and used the other images as query images; next, they evaluated the amount of true targets rates (TTR) at certain false target rates (FTR). For the Market1501[4] they use 15 identities as target people and use 2 images per person as the gallery set. In this work, we further investigate the concept of open-world

re-identification.

Re-identification can be used for many more purposes than just surveillance. In this work, we show that re-identification can also be used to measure cyclist flow and route mappings accurately. This data can subsequently be used by municipalities that are interested in optimizing their cyclist flows. In order to train a model that works well on cyclists, a dataset of cyclists is required. Therefore we create our own dataset to train our model. This dataset is created from multiple cameras set up in multiple cities (Groningen, Rotterdam and Copenhagen). To comply with the privacy laws of the different countries, all of the images of cyclists have been anonymized. In this paper the effect of this anonymization on the re-identification model is investigated. Current works in re-identification mainly use ResNet50 as a backbone, however this network is not ideal to use in an embedded manner. Therefore different mobile models are evaluated. Also current loss functions for re-identification do not actively enforce a certain distance between positive and negative matches. This is problematic for open-world scenarios where a certain threshold is used to classify if there is a match or not. To tackle this problem we introduce a new loss function for re-identification.

The rest of the paper is organized as follows. Section 2 provides an overview of related work in the person re-identification field. Section 3 describes techniques applied during the creation of our dataset, and during the training of our models. Section 4 describes current benchmark datasets in re-identification. Section 5 describes the experiments and their results. Finally, Section 6 describes our main findings. research.

II. Related Work

Person re-identification can be divided into classification-based learning and metric-based learning. This section will be focused on person re-identification in images. Since the work of Gheissari et al.[11] in 2006, person re-identification has mostly been applied to single images. Gheissari et al.[11] used invariant signatures for each image, which are generated by combining normalized color and salient edges. These are hand-crafted features, which performed quite well on their own small scale dataset, but they are not suitable for current benchmark datasets. The state of the art models are all learned models[3]. The best performing model that uses handcrafted features is provided by Zheng et al.[12], they learn a discriminative null space for their features; they achieve a rank-1 accuracy of 69.9%. To achieve good scores for re-identification, it is required that the feature extractor deals well with the challenges mentioned in the previous section. Mainly deep network-based approaches deal well with these challenges. Therefore they are the main focus in this section. For a more elaborate view of person re-identification, we refer to [3, 13] which provides recent surveys covering the entire re-identification field. Generally speaking, two types of deep approaches can be done; one is treating person re-identification as a classification problem; the other is using metric learning based approaches. These approaches can be distinguished by the way the network is optimized. Classification based models are optimized using cross-entropy loss. Models based on metric learning use a distance metric based loss function like the triplet loss to optimize their network.

A. Models based on classification

A model based on classification creates a class prediction for an input image. Here each class represents a unique individual; this means there could be millions of classes for real-world systems.

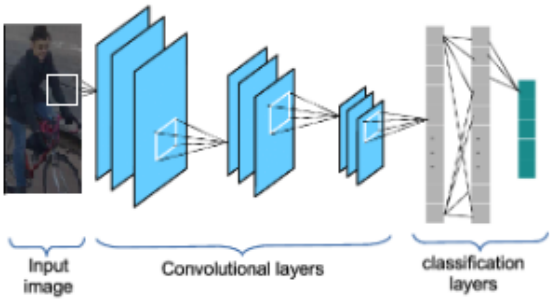


Figure 2: An input image is passed through a convolutional neural network (CNN), and the output is a probability value from 0 to 1 for each of the classification labels the model is trying to predict.

A standard classification model is shown in Figure 2. The loss can be calculated for each image in the training set by calculating the cross-entropy loss between label prediction and ground truth labels. Since often the persons in the training set are not part of the test set, the final prediction layer cannot be used during testing. During testing, the layer of the neural network, which is before the class predictions, is used as an embedding. An important aspect of classification models are the feature extraction networks or backbone networks. In early re-identification works, authors[14]–[19] used shallow CNN's with less than 10 layers as their backbone networks. In later works pre-trained networks from the ImageNet[20] challenge are used as backbone network. These networks are pre-trained because of the small dataset size for re-identification. Mainly ResNet50[21] is a popular choice of backbone network for re-identification.

B. Models based on metric learning

Metric learning approaches generally use Siamese neural networks, which contain two or more identical sub-networks. These sub-networks share the same architecture, weight and parameters. In re-identification, we typically see two or three branch networks, respectively pairwise and triplet networks. **Pairwise models** are used by some researchers[16, 17, 22, 23]. A pair of images is used as an input into Siamese convolutional neural networks(SCNN). First a convolutional model is used to create a feature vector for each image. The similarity between the vectors could be calculated using the cosine distance[14, 16] and

subsequently optimized for positive and negative pairs. The pairwise models have lost their popularity in recent years, because the performance is inferior to the triplet networks. **Triplet networks** use three images as input, one anchor, a positive match and a negative match. Triplet networks are popular for re-identification [24]–[27], and have proven to be successful. The goal of these networks is to pull the positive match closer to the anchor while pushing the negative match further away. This is done by means of optimizing a loss function. The triplet loss can be calculated in different ways, so Hermans et al.[26] compared different formulations of the triplet loss function. They conclude that the batch hard triplet loss yields the best results for person re-identification. The batch hard triplet loss function for each anchor within a batch finds the hardest positive and hardest negative match. In euclidean space, the hardest positive is the positive with the largest distance to the anchor, and the hardest negative is the negative with the smallest distance to the anchor. Recent works combined classification-based models with the models based on Siamese networks, here the loss functions for classification loss and triple triplet loss are combined with certain weights[28]–[31].

C. Global vs. part-based methods

In literature, a distinction can be made between part-based and global methods. The key difference between the methods is that global feature methods use an entire image as input and extract one feature vector from this image, while part-based methods split the input image into parts, and for each of these parts a feature vector is created. Next the feature vectors of the parts are combined into a single feature vector. Splitting the image into parts can be done in many ways. Like using manually designed horizontal windows[32], or extracting the locations of body parts from the image and using these as features[15, 33]. Others divide the image into parts before the pooling layer[34], or using attention-based models to find the most important features of each part of the feature map[35]. Even though part-based models have proven to be very successful at solving the re-identification task, they still rely heavily on the quality of the bounding box[36]. In most challenging cases, current detection performance is not sufficient enough to guarantee a good bounding box. Therefore in this work we emphasize the use of global feature models.

D. Training strategy

An often overlooked part of person re-identification is the method applied during training. For example, adding augmentation to training data, like random erasing data augmentation[25] where random boxes of random images were replaced by random pixel values. This was done to tackle occlusion. Another possibility are methods that allow the model to converge faster, like adding a batch normalization layer or warming up the learning rate[37]. For warm-up, instead of starting at a certain predefined learning rate, the learning rate is linearly increased from 0 towards this learning rate. This is a method used to reduce the effect of early training examples. It was first used for re-identification by Fan et al.[31] and later adapted by many other researchers[27, 29, 38]–[40] in the re-identification field. Another method to increase the score is changing the last stride of the network from 2 to 1[29, 34]. Changing the last stride from 2 to 1 increases the size of the feature map. Higher spatial resolution enriches the granularity of features and can increase the achieved accuracy by a few percent. Another approach seen is adding external information, for example Wang et al.[40] added spatial-temporal information to the images and Lin et al.[41] added 27 attribute labels to each of the images. Adding external information during training is not plausible for an open-world system, so it is outside of the scope for this work. Often these heuristics or methods are adapted from other deep learning tasks, like the ImageNet[20] challenge. Another popular tool is re-ranking (RR)[42], which is applied after the distance matrix has been computed, thus it is considered as post-processing. Here the calculated distance matrix is re-ranked by encoding the nearest neighbors into a vector and using these vectors to re-rank the images based on the Jaccard distance. Using RR can add 10% to the achieved mean average precision (mAP) score and can be defined as post-processing (it is applied after the distances are calculated). It is unfair to compare researchers that applied RR to those who did not. Therefore, often researchers report both scores with and without RR. The scores reported in this work are without RR unless it is explicitly mentioned.

E. Benchmark datasets

To get an idea of the datasets used in person re-identification we compare the CUHK01[43], CUHK03[44], Duke MTMC[2], ViPer[7], and Market1501[4] datasets.

Dataset	BBoxes	Identities	Detection method	# cam
CUHK03[44]	28,192	1467	DPM, hand	2
CUHK01[43]	3884	971	hand	10
Duke MTMC[2]	36,411	1812	hand	8
ViPer[7]	1264	632	hand	2
Market1501[4]	32,364	1501	DPM, hand	6

TABLE I: Re-identification benchmark datasets. Detection is done by hand or with the deformable parts model(DPM) as described in [45]

A comparison of available benchmark datasets can be seen in Table I. The ViPer[7] dataset was one of the first datasets available for re-identification and is still considered one of the most challenging ones[10], this is due to the fact that there are only two images of each person to train on. The CUHK01[43] and CUHK03[44] datasets were both recorded on the campus of the Chinese University of Hong Kong. While the CUHK01 dataset is considered too easy the CUHK03 is certainly not, however the CUHK03 only uses two cameras and researchers prefer the slightly larger Duke MTMC[2] dataset. The Duke MTMC was recorded on the campus of Duke university and was later discontinued because of the breach in privacy of the students and the applications of the dataset[1].

The Market1501[4] is a popular benchmark dataset for re-identification, it contains 32,634 images of 1501 identities. It is used extensively in re-identification research. This dataset is split into 12,936 images of 750 identities for training and 19,271 images of 751 identities as the test set. This test set is then split into a query and gallery set, respectively containing 3360 and 15,913 images. The authors also provided 500,000 images which can be used as distractors in the gallery set. These images however are a collection of bad bounding boxes and do not contain any full persons. All identities in the query set are also present in the gallery set. This dataset will be used to validate the effectiveness of the experiments.

F. Results on benchmark datasets

The scores of the papers submitted to CVPR 2016-2019 are shown in Figures 3 and 4, which are the scores on the Market1501 and Duke dataset. To fairly compare achieved scores, models using re-ranking[42] have not been included in the graph. The scores are given in mean average precision(mAP) and rank-1 accuracy. The scoring metrics will be further exaplained in the third section. Currently the best performing model on the Market1501 benchmark dataset is the one created by Luo et al.[29]. They collected different loss functions and training strategies and applied these for person re-identification, yielding them a 94.5% rank-1 accuracy and a mAP of 85.9%. They combined the triplet loss with center loss and the cross-entropy loss. A Resnet50[21] architecture was used as backbone feature extractor, they changed the last stride from 2 to 1 giving them a larger feature vector output.

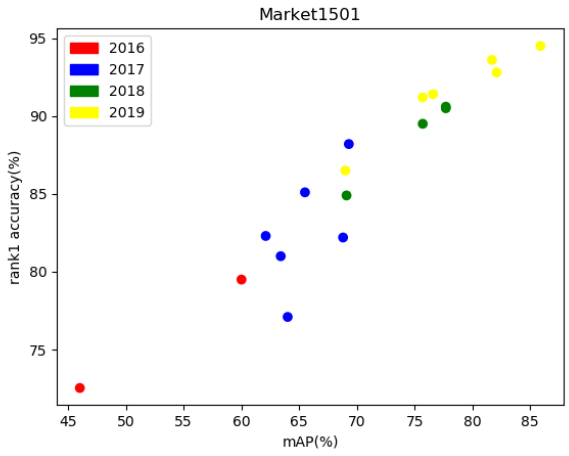


Figure 3: Results on the Market1501[4] dataset. The almost linear increase in scores over the years can be seen in this graph.

The Duke dataset is considered to be slightly more challenging than the Market1501 dataset and this can also be seen in the achieved scores. The results on the Duke dataset are very comparable to the results on the Market1501 dataset, but they are about 10% lower on average, they are shown in Figure 4.

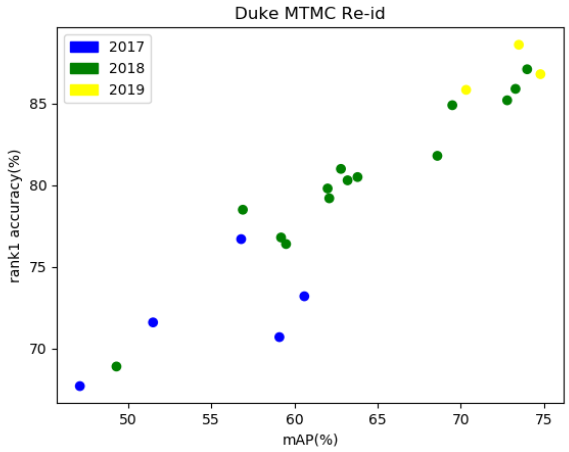


Figure 4: Results on the Duke Re-id[2] dataset. The almost linear increase in scores over the years can be seen in this graph.

For both benchmark datasets we see that the scores are increasing each year showcasing that this is a subject where a lot of improvement is possible. In this work the Market1501[4] dataset will be used to validate the rebuilt models.

III. Methodology

In this section we will discuss the methods applied during our research, first the methods for the creation of the dataset will be discussed, then the methods for re-identification will be explained. Finally the evaluation metrics for re-identification will be explained.

A. Dataset creation

Cross domain performance is generally not consistent with the performance on the original dataset[29]. So to get good performance for cyclist re-identification our own dataset was required. First video footage was collected from different locations, then object detection was applied to all of the footage and finally the tracklets were extracted from the video data.

1) Camera setup: We set up our cameras in three different cities: Copenhagen, Groningen and Rotterdam. An example of the camera setup is shown in Figure 5, and with permission of the different municipalities over 50 hours of video data was collected. The video footage was directly blurred to respect the privacy of the individuals in the footage.

Our main interest in the video footage are the cyclists that pass through one or multiple camera views. From this video data we want images of each unique cyclist to be saved under a new identity. To extract the images of each cyclist from the video footage, first the cyclists must be detected. A region of interest(ROI) was formulated for each camera and location.

An example of this ROI is shown in Figure 6, here the ROI is defined as the red polygon. The ROI was implemented to ignore pedestrians and walking cyclists. For each bounding box we required the center of the bottom axis of the bounding

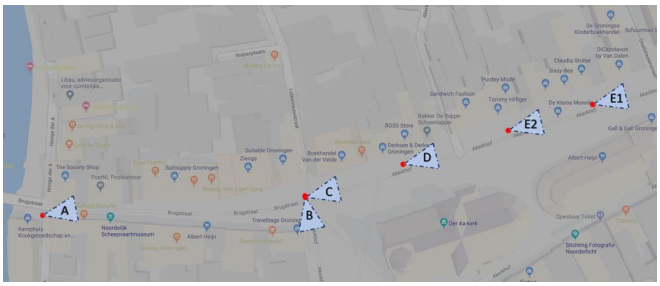


Figure 5: An example of the Camera setup in Groningen. Here five camera's were set up along one of the busiest cycling streets in Groningen.

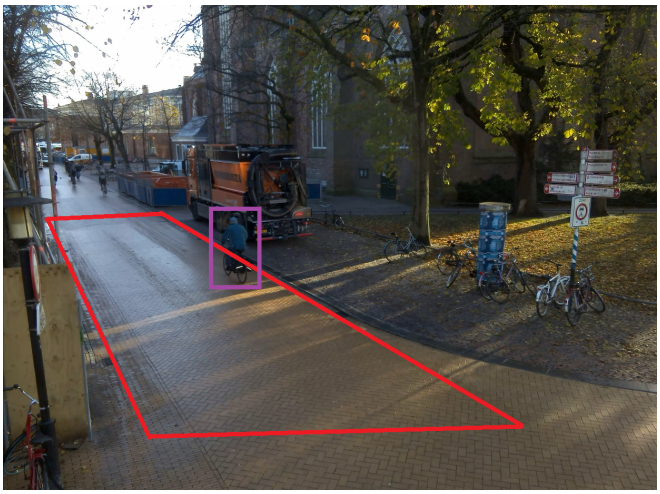


Figure 6: Camera B view in Groningen with ROI in red and bounding box of cyclist in purple

box to be within the ROI. After the ROIs were defined for all cameras and locations, the object detection was applied.

2) Object detection: Object detection is applied to each frame. An object detection model returns the bounding boxes of each frame, and for each bounding box the score, location and class are stored. In a recent survey on object detection[46] the performance of object detection models on the COCO[44] dataset was compared. The COCO[44] dataset is a large benchmark dataset for object detection with over 1.5 million object instances and 80 object categories. The object detection can be applied offline, therefore inference time is disregarded and the model with the best performance is chosen. The updated version of the faster R-CNN[47] is the current state of the art. The faster R-CNN[47] was run over all of the video footage, generating the bounding boxes for each frame. This returned over 10 million bounding boxes for 2 million frames. From all of these bounding boxes the dataset must be created, in this dataset bounding boxes belonging to the same person must be assigned the same identity. To achieve this we must must create tracklets of each cyclist by tracking them through each camera's view.

3) Tracklet creation: To create tracklets the cyclists are tracked through the camera view. This is known as object tracking and is an entire different field of deep learning. For our dataset creation, objects are tracked by using Intersection over Union(IOU) and a Kalman filter[48]. First the bounding boxes which do not meet the requirements (Classification score, Aspect ratio, Horizontal size) are filtered out, the requirements are shown in Table II. The score is the classification probability which is output by the object detection model. The aspect ratio was added to filter out boxes which only contained partial cyclists. The horizontal size requirement was added to filter out small bounding boxes.

After the bounding boxes that did not meet the requirements have been filtered out, the tracklet creation process can start. The tracklet creator is run over successive frames. Given a certain frame's bounding boxes, the IOU of the boxes is calculated with the boxes of the next frame. Boxes with high IOU are potentially the same cyclist while boxes with an IOU of 0 represents bounding boxes without any overlap. The formula for intersection over union is given in Equation 1.

$$IOU_{ij} = \frac{Box_i \cap Box_j}{Box_i \cup Box_j} \tag{1}$$

In an ideal case the IOU would be sufficient to extract the track of a cyclist that has cycled through the camera view, however often cyclists are close together. Thus only using IOU is not an effective solution and gives many incomplete/incorrect tracklets. Many other object tracking models[49] add a re-identification algorithm to their object trackers. Here, bounding boxes are compared by extracting feature vectors and comparing them to find the same person.

Using re-identification to create a dataset for re-identification can be problematic, an example is occlussion of individuals. A well performing re-identification model should be able to re-identify partially occluded individuals, however if these examples are not included in the training set the model will not perform well for occlusion. Therefore we decided not to include re-identification in our tracklet creation process. Instead a Kalman filter[48] was added to predict the location of the bounding box in the next frame, given the previous boxes. Next the IOU of the predicted box with the actual box is calculated. The entire algorithm for the tracklet creation can be found in the Appendix, and the values of the parameters are shown in Table II.

Attribute	Req	Attribute	Req
Classification Score	0.7	Min pickup time (s)	0.3
Aspect Ratio	0.6	IOU min	0.3
Horizontal size	60	IOU max	0.9
Expiration time (s)	0.5	IOU Kalman	0.5

TABLE II: Tracklet creation variables. These variables were were chosen through trial and error.

B. Methods in re-identification

In this section the methods used for person re-identification are explained. The methods for re-identification are divided into five parts: loss functions, backbone networks, data augmentation, training details and inference details.

1) Loss functions: During training a function is defined which should be minimized, this is known as the loss function. The performance of the model will greatly depend on the choice of the loss function. These loss functions differ greatly in their complexity. An optimizer is used to minimize the loss functions and they exist in many different forms. The choice of optimizer is beyond the scope of this paper, so the most common one for re-identification is used, which is the ADAM optimizer[50]. In the person re-identification field four types of loss functions are used: cross-entropy loss, triplet loss, pairwise loss and center loss.

Cross-entropy loss uses the probability predictions and compares these to the true label. The logits are defined as the output of the layer before the classification layer. The logits are passed through a softmax activation function and this turns them into probabilities. The labels are one-hot encoded vectors, containing all 0s and a single 1. Sometimes researchers[27, 31, 39] add noise to the label vectors, this is known as label smoothing. The natural way to measure the distance between two probability vectors is by calculating the cross-entropy loss. The cross-entropy loss is calculated by taking the log of the predicted probability and multiplying this with the probability of the ground truth label. If the labels are one-hot encoded this only requires one calculation per embedding and thus is quite efficient. The formula for the cross-entropy loss is given in Equation 2.

$$L_{identity} = -\frac{1}{N}\sum_{i=0}^{i=N} Label * \log(Pred) \tag{2}$$

Here N represents the batch size. During inference the classification layer is discarded and the logits are used as embedding for comparison. This is done because it is unknown how many identities are included during inference time and the classes differ from the training set thus it is not effective to use the classification layer as embeddings.

Triplet loss requires three inputs, an anchor, a positive match and a negative match. It uses the logits of the deep model as an embedding. The euclidean distance between each of the embeddings is calculated, and with these distances the triplet loss can be calculated. The triplet loss function knows many different formulations, the standard formulation is given in Equation 3.

$$L_{triplet} = \sum_{a,p,n} [m + D_{a,p} - D_{a,n}]_+ \tag{3}$$

Here D represents the distance, a are the anchors, p are the positive matches, n are the negative matches and m is the margin. So $D_{a,p}$ is the distance between the anchor and positive match. $D_{a,n}$ is the distance between the anchor and the negative. The margin is the distance which is enforced between positive and negative matches. If the value for a certain triplet becomes negative, it is replaced by a zero. In this implementation a certain set of B triplets is chosen and their images are stacked into batches of size $3B$, because each triplet consists of 3 images. Each batch of 3B images has B terms which contribute to the triplet loss, given the fact that there are up to $6B^2 - 4B$ possible combinations of these $3B$ images that are valid triplets, using only B of them seems wasteful. Hermans et al.[26] introduced a solution for this, they create batches by randomly sampling P classes (person identities), and then randomly sampling K images of each class, resulting in a batch of PK images. Within a batch of size PK there are different ways of formulating the triplet loss for the batch. Hermans et al.[26] introduce the batch hard triplet loss and the batch all triplet loss. The formulas are given in Equations 4, and 5 respectively.

$$L_{triplet-BH} = \underbrace{\sum_{a=1}^{P}\sum_{i=1}^{K}}_{anchors}[m + max(D_{a_i,p}) - min(D_{a_i,n})]_+ \tag{4}$$

$$L_{triplet-BA} = \underbrace{\sum_{a=1}^{P}\sum_{i=1}^{K}}_{anchors}\underbrace{\sum_{p=1}^{K}}_{pos}\underbrace{\sum_{n=1}^{K}}_{neg}[m + D_{a_i,p} - D_{a_i,n}]_+ \tag{5}$$

The difference between the batch all triplet loss and the batch hard triplet loss is that the batch hard triplet loss only takes into account the hardest positive and hardest negative relative to the anchor. This means, within each batch, only the positive match with the largest distance to the anchor, and the negative match with the smallest distance to the anchor are used to calculate the batch hard triplet loss. For the batch all triplet loss, all positive and negative matches for each anchor within the batch are used to calculate the loss.

A disadvantage of the triplet losses is that they only consider the difference between the positive and negative distance and ignore the absolute values of them. For example with m = 0.3, if the distance with the positive is 0.4 and the distance with the negative is 0.6 the loss for that triplet is 0.1. In another case, the positive distance is 1.4 and the distance with the negative is 1.6, here the loss for the triplet is also 0.1. Since in an open-world scenario we cannot simply prescribe the closest image as a match, a threshold must be set where images are assigned a distractor label. This means the triplet loss might not be suitable for an open-world scenario. Therefore it might be useful to include this margin in the loss function. Therefore we introduce the batch hard pairwise loss.

Batch hard pairwise loss is quite similar to the batch hard triplet loss. For each anchor the hardest positive and hardest negative within the batch are used to calculate the loss. The formula is given in Equation 6.

$$L_{pair-BH} = \underbrace{\sum_{i=1}^{P}\sum_{a=1}^{K}}_{anchors}[max(D_{a,p}) - m_p]_+ + [m_n - min(D_{a,n})]_+ \tag{6}$$

Here m_p is the positive margin and m_n is the negative margin. The key difference between the batch hard triplet loss, and the batch hard pairwise loss, is that the margin is enforced separately for each positive and negative pair.

This gives us more control over the distance that a positive/negative match should have.

Center loss is often added to maximize intra-class compactness, it is mainly popular in the facial recognition scene[51]. The formula for center loss is:

$$L_{center} = \frac{1}{2}\sum_{j=1}^{B}|L_j - C_j|^2 \tag{7}$$

L represents the embedding of a certain input image, C is the center of all embeddings of the matching identity and B represents the batch size. This function learns a center for each class and calculates the Euclidean distance of the sample to its center. The sum of these distances is then multiplied by 0.5. Minimizing center loss will increase intra-class compactness, meaning positive pairs are pulled closer together. The center loss can be combined with the triplet loss to make the result more robust. A disadvantage of the center loss is that it is sensitive to outliers.

2) Backbone networks: Backbone networks form an important part of re-identification models. Given an input image, the backbone network returns a feature vector of N features. During training these features are used to calculate the loss and train the network. The size of the feature vector N depends on the size of the final layers of the network. For Resnet50[21] the layer before the classification layer has 2048 nodes, most researchers use these 2048 nodes as their feature vector. Resnet50[21] is the most popular backbone network for re-identification at this moment, however there are networks which outperform Resnet50 in a number of other deep learning tasks. A few networks which could outperform Resnet50 for re-identification were chosen: EfficientNets[52], MobileNets[53] and Xception[54].

ResNets were introduced in 2015 by He et al.[21], they introduced residual connections which made it possible to train deeper networks. The ResNet architectures were originally created for use on the ImageNet[20] challenge and were later adapted for many other deep learning tasks among which person re-identification. ResNet50 is used by a significant amount of person re-identification researchers[3], however there are many other potentially effective backbone networks. In this work we compare the effectiveness of several state of the art architectures from other deep learning tasks for person re-identification. We compare the standard ResNet50 backbone to EfficientNets[52], MobileNets[55] and Xception[54].

EfficientNets were first introduced by Tan et al.[52], they showed that instead of just scaling networks in depth it is also useful to scale the network in width and resolution. They created different scales of models ranging from B0 to B7 and from 5.3m parameters to 66m respectively. They show that their network is able to outperform ResNet50 on a number of classification tasks.

MobileNets currently have three versions: MobileNet[53], MobileNetV2[56] and MobileNetV3[55]. For each version the achieved accuracy on ImageNet was improved and they use roughly the same amount of parameters (4m). For re-identification to be successful we believe it must be deployed in an embedded way. MobileNets have a small number of parameters and they have been optimized for use on mobile devices. This makes them ideal candidates for re-identification.

Xception was introduced by Chollet et al.[54] shortly after ResNets were invented. It outperformed Resnet on many challenges but has not been used for re-identification. The architecture is heavily inspired by the Inception architecture[57] but the Inception modules are replaced by depth wise separable convolutions. In this work both Inception and Xception are evaluated for re-identification.

3) Data augmentation: In this work different types of data augmentation are applied. The effect of these augmentations on the performance is evaluated. Random Erasing Augmentation[25], which is very popular in the re-identification field, will be applied. Also the effect of RandAugment[58] which is used by the top performing models on the ImageNet challenge is evaluated for re-identification.

Random Erasing Augmentation has proven to be very successful for person re-identification. Occlusion is a common challenge in person re-identification. To address the occlusion problem and improve the generalization ability of re-identification models, Zhong et al.[25] introduced random erasing augmentation. Within an image they select a random region W x H and replace the pixel values in this region by pixels of random values. Random erasing is not applied to all images, it is applied with certain probability P_{erase}. In this work, we use the same hyperparameters as described in the original paper[25]. An example of random erasing is shown in Figure 7, here the same image was sampled independently eleven times through the random erasing procedure with $P_{erase} = 1.0$.

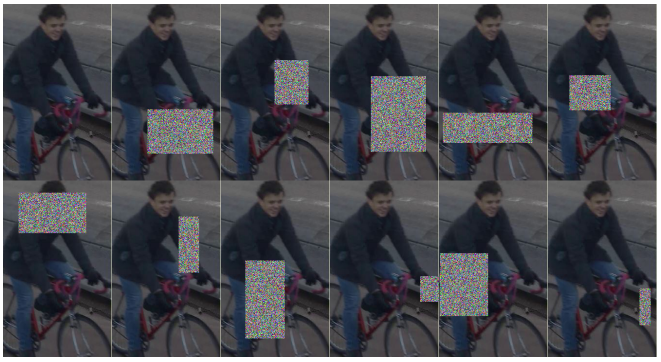


Figure 7: Random Erasing Augmentation[25] applied to the image in the top left, 11 times with p_{erase} = 1.0

RandAugment is a data augmentation technique, introduced by Cubuk et al.[58]. In this work this technique of augmentation is applied for person re-identification. RandAugment[58] applies a random number of augmentations from a list of possible augmentations:

- identity
- autoContrast
- Equalize
- Rotate
- Solarize
- Color
- Posterize
- Contrast
- Brightness
- Sharpness
- Shear-x
- Shear-y
- translate-x
- translate-y

Both the number of augmentations that are applied, and the magnitude of these augmentations, are hyperparameters. Cubuk et al.[58] state that the values for these hyperparameters depend on network size, and the size of the training set. For Resnet50 they use magnitude M = 9 and number of transforms $N_{transforms}$ = 8. An example of RandAugment applied to images is shown in Figure 8.

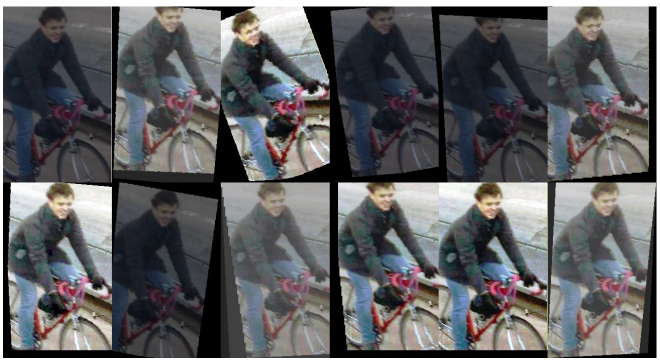


Figure 8: RandAugment[58] applied to the image in the top left, 11 times with $N_{transforms}$ = 5 and M = 5

4) Training details: After the augmentations have been applied to the training data, the images are pre-processed. After pre-processing the images the training process can start.

Pre-processing is applied to the training data before training our network. We follow the pre-processing steps as described by Luo et al.[29]. First each image is resized to the input size used by the different networks, for example 256x128 pixels. Following the ImageNet pre-processing steps each image is decoded into 32-bit floating point raw pixel values in [0, 1]. The RGB channels are normalized by subtracting 0.485, 0.456, 0.406 and dividing by 0.229, 0.224, 0.225, which are the mean and the standard deviation of the images used for ImageNet. These values are used since the backbone models are often pre-trained on ImageNet. Next a 1xN dimensional fully connected layer is added to the network which is used as logits for the triplet losses. If a classification loss is used then another fully connected layer, with softmax activation function is added. This final layer can be used for ID predictions.

For each epoch the dataset is divided into batches of P identities and K images per identity, our standard batch size is 64 images with P = 16 and K = 4. Each identity is only sampled once for each epoch. If an identity has less than K images available a random copy of one of the available images is added for that identity. In available benchmark datasets persons that appeared in multiple cameras subsequently are manually labeled as the same person. This is one of the factors why person re-identification datasets are relatively small as this is a lot of work. In our dataset these persons are not saved under the same identity since this would results in an immense workload. To prevent the same person from appearing under a different identity code in the same batch, we sample our batches from a single camera. Next the loss is calculated for each batch. The loss is defined as one of the described loss functions or a combination thereof. Once the loss function has been defined, an optimizer can be deployed to minimize the loss function and find a global minimum of the defined loss function. In this work, the ADAM[50] optimizer is used. The optimizer takes the learning rate as an input. The learning rate weighs how heavily the calculated gradient is used to update the existing values. A large learning rate results in fast initial convergence but it might fail to converge to global minimum. Using a small learning rate will result in slow convergence and is therefore also not optimal. To get fast convergence and find a global minimum both high and low learning rates are needed and therefore learning rate schedules are created. The learning rate schedules used in this work will now be explained. **Learning rate decay** is the practice of decaying the learning rate at certain epochs by a certain factor and this helps convergence towards a global optimum of the loss function. This is known as step decay. Instead of starting at a certain learning rate, the learning rate is warmed up as done by Fan et al.[31]. The Learning rate is increased from 0 to the learning rate used as input in 20 epochs. **Cosine Learning rate decay** is an alternative learning rate schedule. After the warm-up stage described earlier, the cosine decay decreases the learning rate slowly at the beginning. Around the middle the learning rate decreases almost linearly, and the decrease slows down again around the end. The idea of cosine decay was introduced for other deep learning tasks by Loshcilov et al.[59]. In this work it is adapted for person re-identification. An example of the different learning rate schedules is shown in Figure 9.

$$LR_{step} = \begin{cases} e * \frac{LR}{20} & e \leq 20 \\ LR & 20 < e \leq 100 \\ LR * 10^{-1} & 100 < e \leq 150 \\ LR * 10^{-2} & 150 < e \leq 200 \\ LR * 10^{-3} & 200 < e \leq 250 \end{cases} \tag{8}$$

$$LR_{cosine} = \begin{cases} e * \frac{LR}{20} & e \leq 20 \\ 10^{-4.5+1.5*\cos\left(\frac{e-40}{65}\right)} & 20 < e \leq 250 \end{cases} \tag{9}$$

The formulas of the learning rate schedules are given in the equations above. Here e represents the current epoch and LR is the learning rate for the optimizer. The formula for step decay without warm-up is the same as the formula for the step decay with warm-up with exception of the first step where the learning rate starts from the constant value.

5) Inference details: Inference/testing is different from training for re-identification. During inference the amount of

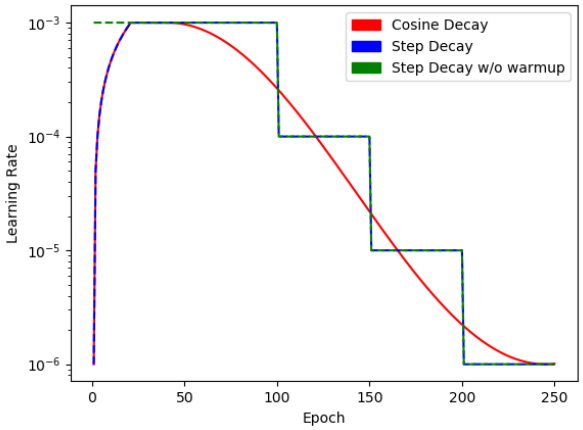


Figure 9: Step decay and Cosine decay scehdules. Here 10^{-3} was used as learning rate and the schedules were created for 250 epochs.

classes is unknown. Also since a class represents a person the details of each class are still unknown, thus a standard classifier cannot be applied to classify an input image. In benchmark datasets the test set is split into two parts, a query set and a gallery set. The goal is to match each person from the query set to the correct identity in the gallery set. Multiple images of an identity might exist in the gallery/query. In most cases Euclidean distance is used as a similarity measure, the smaller the distance between two feature vectors, the more similar the images should be. During the calculation of the scores images taken from the same camera are not compared. This is done to simulate the objective of re-identification, which is to find an individual in a different camera view. In a closed-world situation a ranking is created based on Euclidean distance. Given an image from the query set, the images from the gallery set are ranked based on Euclidean distance. In an open-world situation a threshold is used to describe if an image is a match or not. Therefore different scoring metrics must be used.

C. Evaluation metrics

To gain understanding of the model's performance, first the underlying metrics must be explained. A distinction is made between closed-world and open-world scoring metrics.

1) Closed-world: The cumulative matching curve(CMC) and the mean average precision(mAP) are used as the benchmark scoring metrics in person re-identification. The CMC measures the retrieval precision, given a list of closest matches from the gallery, the rank-x indicates the probability that the matching image is among the first x positions. The rank-1 accuracy gives the probability that the top ranked image is a match. In early research in person re-identification only the CMC scores were reported, but with growing datasets this score on its own is not a good representation of the model's performance anymore. The mAP was added to take into account the position of all positive matches from the gallery. The mAP is calculated by first calculating the Average Precision (AP), subsequently the mean of all the average precisions is calculated. In Figure 10 a calculation example of these scoring metrics is given. The mAP gives a better indication of the performance since it takes the position of each matching person into account.

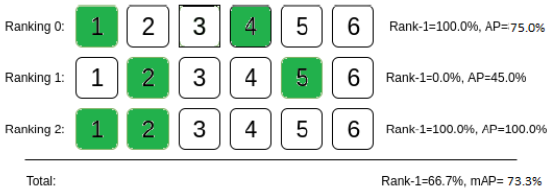


Figure 10: Example scores for rank-1 and average precision. Only using the rank-1 accuracy does not give a good indication of the performance if a large gallery set with multiple matches is used.

2) Open-world: In the real-world application the gallery size may vary, and not every query image has a match in the gallery set. The scoring metrics used for closed-world re-identification would not give a good indication of the model's performance. Therefore the open-world re-identification problem is viewed as a binary classification problem. For each gallery image the model must classify whether it matches the query image or not. The F1-score can be used as a scoring metric for open-world re-identification. First the precision and recall are calculated, next these can be used to calculate the F1-score. The precision and recall are defined in Equation 10.

$$Precision = \frac{TP}{TP + FP} \text{ and } Recall = \frac{TP}{TP + FN} \quad (10)$$

With TP = True positives, FP = False positives, and FN = False negatives. The formula for the F1-score is given in the equation below.

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (11)$$

The F1-score helps find a balance between the precision and recall, a value of 1.0 portrays a perfectly working system.

IV. Dataset

In this section the created dataset is discussed, and some comparison with benchmark datasets is made.

A. Cyclist Re-id

In this paper a new dataset named 'cyclist Re-id' is introduced. It was created because no dataset of cyclists for re-identification exists. Another issue in re-identification datasets compared to datasets in other deep learning fields is that they lack in size. Also our dataset is the first fully anonymized re-identification dataset. Following the methods described in the previous section, this dataset was extracted automatically from the 52 hours of video footage. Given video footage the algorithm returns the tracklets of unique identities that cycled through the camera views. An overview of the compositions of the training data is given in Table III.

Location	Cam	Hours	# Crops	# IDs
Groningen	5	40	188,571	9584
	A	8	53,105	3120
	B	8	65,369	2674
	C	8	16,079	914
	D	8	17,221	1303
	E	8	36,797	1573
Rotterdam	2	8	51,863	1296
	A	4	28,390	706
	B	4	23,473	590
Copenhagen	2	4	20,347	789
	A	2	8,121	264
	B	2	12,226	525
Total		52	260,781	11,669

TABLE III: Composition of training data from different locations and cameras

This gives us a total of 260,781 images of 11,669 non-unique identities. These identities are defined as non-unique as they have not been labeled for cross camera re-identification. The test set must consist of identities which have been labeled across cameras so that we can assess the cross camera re-identification performance. To collect test data the videos were first split into training/validation parts. For each hour of footage, 50 minutes were reserved for training data and 10 minutes for validation data. Next the tracklet extractor was also applied to the validation videos, and the cross camera labeling was done manually. The test set contains identities from all of the recorded locations, an overview can be seen in Table IV.

Location	# IDs	# Images
Groningen	110	7856
Rotterdam	91	7345
Copenhagen	60	120
Total	261	15161

TABLE IV: Composition of test data from different locations

In total there are 15,161 images of 261 identities split over up to 4 cameras per location. These images are saved in their original size. Instead of splitting the data into a query and gallery set, we sample each image as a query and compare it to the left over images. So the gallery set is the test set without the current image. Before calculating the score, we remove the images which belong to the same identity from the gallery set.

V. Experiments Discussion

In this section five different experiments are discussed, in the first experiment the effect of different types of blurring on the re-identification performance were evaluated. This experiment was done to test the feasibility of using blurred/anonymized images (as required by municipalities) for re-identification. In the second experiment, four models that perform well on benchmark datasets are rebuilt and tested on our own data. In the third experiment lightweight backbone networks which can be used on embedded platforms are evaluated for re-identification. In the fourth experiment we evaluate cosine learning rate decay and RandAugment for re-identification. In the fifth experiment the introduced loss function, batch hard pairwise loss, is compared with other triplet losses for closed-world and open-world re-identification.

A. Experiment 1: Influence of blurring on re-identification performance

It was agreed with the municipalities, where the video footage was taken, that the stored data must not contain any recognizable persons. To achieve this, blurring must be applied to the created training set. This experiment was set up to evaluate which type of blurring has the least effect on the re-identification performance. Different blurring strategies were tested: average blurring; gaussian blurring; and median blurring from the OpenCV library[60] were applied to the entire image dataset. Another blurring strategy, YoloFace[61] was also included. YoloFace[61] finds bounding boxes of the faces in each crop, and applies a median blur to these bounding boxes.

The blurring was applied to the training set of the Market1501[4] dataset while varying the blurring window. Since the image size in the Market1501[4] data is the same for all images, fixed values of 5x5, 7x7 and 9x9 pixels were chosen for the blurring window size. From a visual test it was concluded that using 5x5 pixels as blurring windows was sufficient to anonymize. The larger windows were added to test the effect of these larger windows on the achieved accuracy. The visual effect of the different blurring types is shown in Table V.

Here we see that the Average Blurring technique has trouble anonymizing the crop. The larger the window used the more recognizable features disappear. YoloFace yields the best visual results.

The TriNet[26] model is commonly used as a baseline due to the simple structure. For this experiment we will also use TriNet as a baseline model. In their paper they achieve a rank-1 score of 83.3% and mAP of 64.3% while using a margin of 0.2 for the triplet loss. Resnet50[21] is used as a feature extractor and it is trained for 150 epochs. The batch all triplet loss is used as a loss function, and the standard learning rate decay without warm-up is used. The re-created baseline achieves a rank-1 score of 83.7% and mAP of 66.0% on the market1501 dataset.

This network was now trained on the different blurred training sets, and evaluated on the original test set. The rank-1 and mAP scores have been summarized in Table VI.

Window size	5	7	9
	rank-1 / mAP	rank-1 / mAP	rank-1 / mAP
Gaussian	79.1% / 61.3%	76.6% / 56.2%	73.2% / 51.4%
Average	77.5% / 56.1%	71.7% / 49.3%	64.5% / 39.8%
Median	79.3% / 61.7%	73.2% / 51.0%	65.6% / 42.7%
YoloFace	82.8% / 65.3%	**Original**	83.7% / 66.0%

TABLE VI: Rank-1 and mAP performance on the market1501 dataset with different anonymization strategies.

The visual effect of the larger blurring windows is minimal, however the effect on the performance of the larger window sizes is significant. The effect of the YoloFace

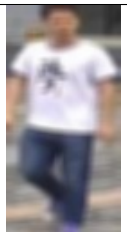
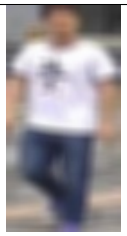
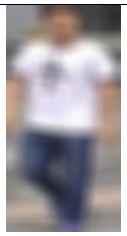
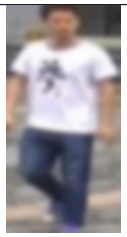
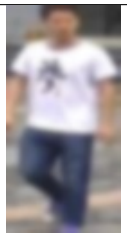
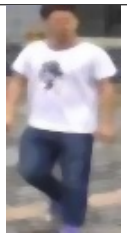
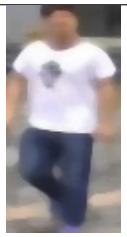
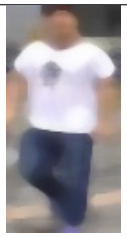
Blur window	5	7	9
Gaussian			
Average			
Median			
Original		**YoloFace**	

TABLE V: Visual effect of different types of blurring, and the different blurring windows on the same image from the Market1501[4] dataset. The original image can be found in the bottom left.

blurring strategy on the re-identification performance is significantly less than standard blurring strategies, however the Yoloface strategy adds an uncertainty. The YoloFace algorithm at this time cannot guarantee to find the face in each of the crops. Given the responsibility of anonymizing the dataset it was chosen to continue with the median blur.

B. Experiment 2: Performance on our own dataset

In this experiment several state of the art models were rebuilt and evaluated on our own dataset. The first model used is the model designed by Zheng et al.[62] which in literature is often described as a baseline for ID based re-identification. The cross-entropy loss function described in the previous section is used. The next model used is the TriNet model which was introduced by Hermans et al.[26]. This model uses different representations of the triplet loss, we will evaluate this model with the batch all triplet loss and the batch hard triplet loss. Next Random Erasing augmentation was added to the TriNet model as demonstrated by Zhong et al.[25]. The final model used is the Bag of Tricks model designed by Luo et al.[29]. They combine the triplet loss with the cross-entropy loss, and the center loss. Also Random Erasing and a learning rate schedule were used. A short overview of the models can be found in Table VII. To verify the models were rebuilt correctly we trained/evaluated them ourselves on the Market1501 dataset. The scores in the table represent the scores that our rebuilt models achieved, not the scores originally reported in the papers.

Model	**Loss**	**Heuristics**	**rank-1 / mAP**
DCNN[62]	Cross-entropy	-	78.04% / 58.89%
TriNet[26]	Triplet	-	82.60% / 65.79%
Random Erasing[25]	Triplet	RE	83.94% / 68.67%
Bag of Tricks[29]	Combined	RE, WL	91.01% / 80.43%

TABLE VII: State of the art models for Market1501 dataset (RE = Random erasing augmentation, WL = warm-up learning rate).

These models use Resnet50 as backbone network with the learning rate set to $3.5 * 10^{-4}$ and decayed by a factor 10 after 100, 150, and 200 epochs. The margin for the triplet losses is set to 0.3 for all models. The Bag of Tricks model uses the warm-up learning rate schedule. The batch size is set to 64, and the pre-processing steps described in the training details section are applied. The models were trained for 250 epochs, each epoch containing 400 batches of images. The scores that the different models achieve on the combined test set are shown in Table VIII.

Model	**rank-1 / mAP**
TriNet (batch all)[26]	75.8% / 63.1%
TriNet (batch hard)[26]	74.2% / 60.8%
TriNet[26] + Random erasing[25]	78.8% / 67.8%
DCNN[62]	70.3% / 57.1%
DCNN[62] + Random erasing[25]	75.5% / 57.6%
BoT (batch all)[29]	84.7% / 75.2%
BoT (batch hard)[29]	84.0% / 73.8%

TABLE VIII: Scores of different benchmark models on the cyclist dataset.

As expected the model with the best performance on the Market1501 dataset also has the best performance on our dataset. Furthermore, it can be seen that the random erasing heuristic increases the accuracy, and using the batch hard triplet loss instead of the batch all variant seems to have a negative effect on the performance for our dataset, while on the Market1501 dataset it increased performance. This might be caused due to errors in the training set. The batch hard triplet loss finds the most difficult positive and negative match. If one of these contains a wrong identity the model will perform well. The Bag of Tricks model was chosen as baseline for the next experiments.

C. Experiment 3: Backbone network

Resnet50[21] is the most popular backbone network in person re-identification. Resnet50 used to have state of the art performance on the ImageNet challenge. However, in recent years many networks have outperformed the Resnet50[21] network on the ImageNet challenge. Another issue encountered while using Resnet50 for re-identification, is that it does not run well on embedded platforms due to

its large size. In this experiment we focus on networks that achieve similar performance on the ImageNet challenge but use fewer parameters and thus should have faster inference time on an embedded platform. An overview of the networks that will be evaluated in this experiment are shown in Table IX.

Model	rank-1(%)	# Parameters	Year
Resnet50[21]	75.9	26m	2016
MobileNetV1[53]	70.6	4.2m	2017
MobileNetV2[56]	72.0	3.4m	2019
MobileNetV3[55]	73.3	4.0m	2020
EfficientnetB0[52]	77.3	5.3m	2019
EfficientnetB1[52]	79.2	7.8m	2019
NasNetMobile[63]	74.0	5.3m	2018
InceptionV3[57]	78.8	24m	2017
Xception[54]	79.0	23m	2017

TABLE IX: rank-1 accuracy of different backbone networks on the ImageNet[20] challenge

The Bag of Tricks model of Luo et al.[29], which had the best performance on our dataset in the previous experiment is used as a baseline. The Resnet50 backbone they used is replaced by the networks shown in Table IX. These networks are available through the Keras[64] Python library. Next the different backbone networks are trained on our dataset. The training steps are similar to the previous experiment. The results have been summarized in Table X.

Backbone network	rank-1 / mAP
Resnet50[21] (baseline)	84.7% / 75.2%
MobileNetV1[53]	76.4% / 65.0%
MobileNetV2[56]	81.4% / 69.3%
MobileNetV3[55]	82.8% / 73.8%
EfficientNetB0[52]	86.2% / 78.4%
EfficientNetB1[52]	86.6% / 79.0%
NasNetMobile[57]	82.8% / 73.9%
InceptionV3[51]	83.2% / 74.0%
Xception[54]	81.8% / 73.2%

TABLE X: Scores of different backbone networks using the Bag of Tricks[29] model as a baseline.

Strong correlation can be seen between the results achieved on the ImageNet challenge, and the results achieved by these backbone architectures when applied to our dataset. The Inception and Xception backbone networks do not outperform Resnet50. EfficientNetB0 has fewer parameters than ResNet50, and achieves a better score. EfficientNetB1 only gives a slight increase in performance over EfficientNetB0, while using significantly more parameters. Given the results of this experiment we conclude that EfficientNetB0 is the best alternative network.

D. Experiment 4: Evaluating Heuristics

Heuristics applied during training have great influence on final model performance. Some heuristics are already used by the Bag of Tricks model such as warm-up and random erasing. In this experiment two heuristics are introduced for re-identification: RandAugment[58] and Cosine learning rate decay[59].

1) Randaugment: Will be evaluated using different values for the amount of transforms (N) used, and the magnitude (M) of the transforms. The optimal values of N and M depend on the size of the dataset and the backbone architecture that is used[58]. In this experiment the amount of transforms will be varied from 1-16, and the magnitude is varied from 1-30. These will be evaluated using the EfficientNetB0 setup from the previous experiment as a baseline. This baseline scored a rank-1 accuracy of 86.2%, and a mAP of 78.4%. Cubuk et al.[58] found that the optimal value of the Magnitude was 9 while using a ResNet50 backbone network. Using this value for the magnitude we first vary the number of transforms, the corresponding scores are shown in Figure 11.

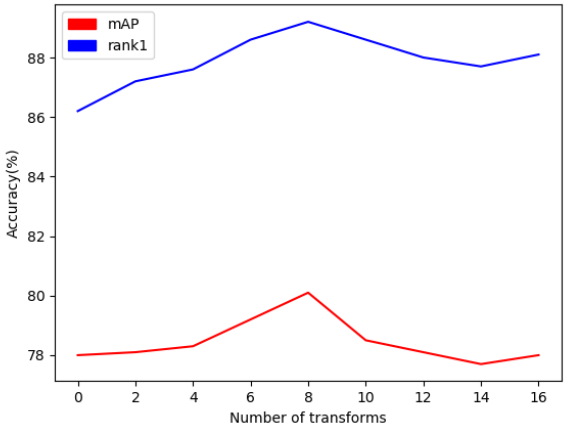


Figure 11: Randaugment with the magnitude set to 9, and the number of transforms varied from 0-16

Here we find that the optimal amount of transforms is 8. For the next part the amount of transforms N is set to 8, and the experiment is repeated but now the number of transforms is kept constant. The magnitude is varied from 0-30. The achieved accuracy's are shown in Figure 12.

Here we find that the optimal value for the magnitude is 8. Furthermore we conclude that the magnitude has greater influence on the performance than the number of transforms. For EfficientNetB0 and our dataset the optimal values are $N = 8$ and $M = 8$. After adding RandAugment the rank-1 accuracy is further increased to 89.1%, and the mAP is increased to 80.2%.

2) Cosine Learning rate decay: Using cosine decay instead of step decay could further increase the convergence towards a global minimum. In this part the performance of networks trained with different learning rate schedules is compared. This experiment is repeated 5 times in order to calculate the mean and the standard deviation of the achieved scores. The mean results and the standard deviations are presented in Table XI.

We see the importance of warming up the learning rate, also we see that using a cosine decay increases the final performance slightly while also decreasing the variance in the results.

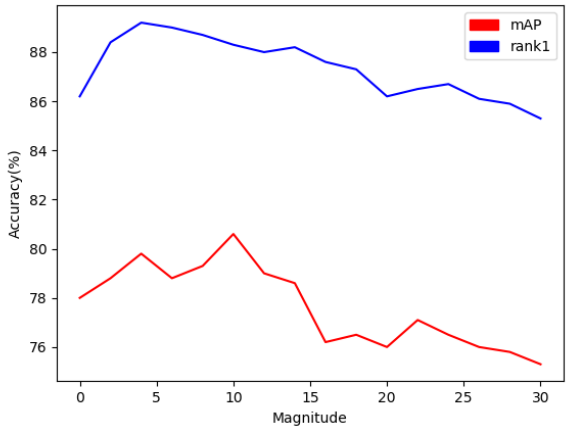


Figure 12: Randaugment with number of transforms set to 8 and the Magnitude varied from 0-30.

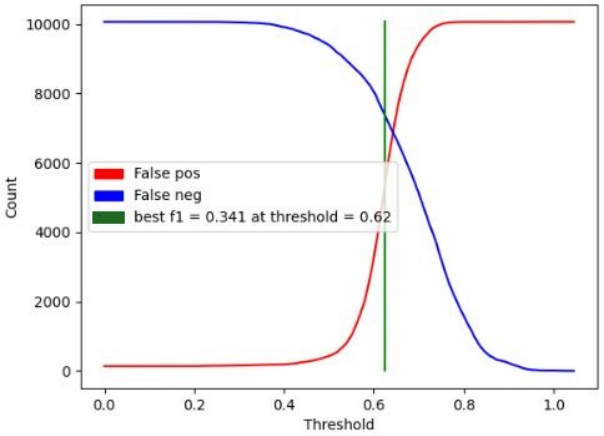


Figure 13: False positives/false negative curve using the batch hard pairwise loss

Learning Rate Schedule	rank-1(%)	mAP(%)
Step decay	87.2±1.7	77.1±1.3
Step decay + warmup	89.0±0.7	80.2±0.5
Cosine decay + warmup	89.6±0.3	81.1±0.4

TABLE XI: Influence of learning rate schedules on final model performance. The mean performance is shown with the standard deviation.

E. Experiment 5: Batch hard pairwise loss

In this experiment the effectiveness of the batch hard pairwise loss function is evaluated for open-world re-identification, and closed-world re-identification. We will use the setup from the previous experiment, and replace the batch all triplet loss with the batch hard pairwise loss.

1) Closed-world: First the loss functions are compared in a closed-world setting. The setup from the previous experiment is used as a baseline, this baselines achieves a rank-1 accuracy of 89.8% and a mAP of 81.4%. If the batch hard pairwise loss is used this setup achieves a rank-1 accuracy of 87.3% and mAP of 79.2%. So in a closed-world setting the batch hard pair loss yields slightly worse results.

2) Open-world: For this part of the experiment we switch to the open-world scoring metrics, which have been explained in the third section. A distance threshold must be chosen, if the distance is larger than this threshold then the image is not a match, and if the distance is smaller than the threshold the image is saved as a match. This threshold is chosen at the point where the F1-score is maximized, this can be done since the precision, recall and F1-score depend greatly on the distance threshold. The curve of the false positives/false negatives is shown in Figure 13, the green line represents the threshold where the F1-score is maximized.

The achieved scores are shown in Table XII. The batch hard pairwise loss performs slightly better for an open-world setting.

Loss function	Precision	Recall	F1-score	Threshold
Batch all triplet	34.8%	23.6%	28.1%	0.82
Batch hard pairwise	38.9%	30.3%	34.1%	0.62

TABLE XII: Precision, recall, F1-score and threshold for the triplet loss variants

Changing to an open-world setting has a lot of impact on the achieved scores. To increase the scores we try and simulate a more realistic real-world scenario where instead of using single images for re-identification, the tracklet of images is used as an input. This should create a more robust embedding. The tracklet of images is combined into a single embedding by taking the average of all single image embeddings. The F1-score after applying averaging over the embeddings is shown in Table XIII.

Method	Precision	Recall	F1-score	Threshold
Averaging	93.2%	89.9%	91.5%	0.92

TABLE XIII: F1-score after applying averaging

VI. Conclusions and Outlook

In this study, a new dataset for re-identification of cyclists was created from actual camera footage in different municipalities. We evaluated different blurring techniques and their effect on the re-identification performance. We evaluated various state-of-the-art models for re-identification on our dataset, and replaced the large backbone architectures by smaller ones. Furthermore the effect of several heuristics from other deep learning areas was evaluated. We also evaluate several variants of the triplet loss function for open-world re-identification.

The experiments show that currently the most effective blurring method is YoloFace, this method impacts the re-identification score with less than 1%, however YoloFace

in the current state is not reliable enough. Therefore a median blur was used which impacts performance by around 4%. The best performing model on our dataset that uses global features is the Bag of Trick model, which achieves a rank-1 score of 84.7% and a mAP of 75.2%. Replacing the standard ResNet50 backbone by other backbone architectures with better performance on the ImageNet challenge can further increase the performance. Also similar performance can be achieved using backbone architectures with up to 5 times fewer parameters. Adding RandAugment to our dataset, and tuning the hyper-parameters can improve the score with up to 3%. Using Cosine decay instead of step decay further increases the performance by 0.7%. Furthermore we show that for an open-world scenario the batch hard pair loss yields better results than the other triplet loss functions used in closed-world re-identification. Open-world re-identification remains challenging, an effective method for increasing the performance in an open-world scenario is combining the embeddings of the tracklets per camera. This increases the F1-score to 91.5%.

In the future, we would like to improve our dataset. This could be done by including re-identification during the dataset creation process. Applying re-identification to the bounding boxes which have been detected per frame would increase the quality/quantity of the tracklets that are extracted, and would prevent simple tracking errors being present in the dataset. Also, the training set could include cross-camera identities which would increase the quality of the training set. Furthermore, the current state of the art performing models in re-identification are approaching perfect scores on benchmark datasets, anonymizing the data could make it more challenging to achieve high scores. We believe the current re-identification research is ready to make the step towards the more practical open-world re-identification, however currently the area of open-world re-identification remains greatly overlooked.

VII. Acknowledgements

This research was done in collaboration with Technolution

References

[1] M. Madhumita, "Who's using your face? the ugly truth about facial recognition," *Financial times*, 2019.

[2] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016.

[3] Q. Leng, M. Ye, and Q. Tian, "A survey of open-world person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[4] 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*, pp. 1116–1124, 2015.

[5] F. Xiong, M. Gou, O. Camps, and M. Sznaier, "Person re-identification using kernel-based metric learning methods," in *European conference on computer vision*, pp. 1–16, Springer, 2014.

[6] W. Li, X. Zhu, and S. Gong, "Person re-identification by deep joint learning of multi-loss classification," *arXiv preprint arXiv:1705.04724*, 2017.

[7] D. Gray, S. Brennan, and H. Tao, "Evaluating appearance models for recognition, reacquisition, and tracking," in *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, vol. 3, pp. 1–7, Citeseer, 2007.

[8] X. Li, A. Wu, and W.-S. Zheng, "Adversarial open-world person re-identification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 280–296, 2018.

[9] T. Yu, D. Li, Y. Yang, T. M. Hospedales, and T. Xiang, "Robust person re-identification by modelling feature uncertainty," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 552–561, 2019.

[10] S. Gong, M. Cristani, C. C. Loy, and T. M. Hospedales, "The re-identification challenge," in *Person re-identification*, pp. 1–20, Springer, 2014.

[11] 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, pp. 1528–1535, IEEE, 2006.

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

[13] A. Bedagkar-Gala and S. K. Shah, "A survey of approaches and trends in person re-identification," *Image and Vision Computing*, vol. 32, no. 4, pp. 270–286, 2014.

[14] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *European conference on computer vision*, pp. 791–808, Springer, 2016.

[15] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang, "Spindle net: Person re-identification with human body region guided feature decomposition and fusion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1077–1085, 2017.

[16] 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*, pp. 34–39, IEEE, 2014.

[17] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3908–3916, 2015.

[18] S. Wu, Y.-C. Chen, X. Li, A.-C. Wu, J.-J. You, and W.-S. Zheng, "An enhanced deep feature representation for person re-identification," in *2016 IEEE winter conference on applications of computer vision (WACV)*, pp. 1–8, IEEE, 2016.

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

[20] 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*, pp. 248–255, Ieee, 2009.

[21] 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*, pp. 770–778, 2016.

[22] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang, "Joint learning of single-image and cross-image representations for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1288–1296, 2016.

[23] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1386–1393, 2014.

[24] S. Ding, L. Lin, G. Wang, and H. Chao, "Deep feature learning with relative distance comparison for person re-identification," *Pattern Recognition*, vol. 48, no. 10, pp. 2993–3003, 2015.

[25] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

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

[27] H. Lawen, A. Ben-Cohen, M. Protter, I. Friedman, and L. Zelnik-Manor, "Attention network robustification for person reid," *arXiv preprint arXiv:1910.07038*, 2019.

[28] Z. Zhang, C. Lan, W. Zeng, and Z. Chen, "Densely semantically aligned person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 667–676, 2019.

[29] H. Luo, Y. Gu, X. Liao, S. Lai, and W. Jiang, "Bag of tricks and a strong baseline for deep person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[30] R. Quan, X. Dong, Y. Wu, L. Zhu, and Y. Yang, "Auto-reid: Searching for a part-aware convnet for person re-identification," *arXiv preprint arXiv:1903.09776*, 2019.

[31] X. Fan, W. Jiang, H. Luo, and M. Fei, "Spherereid: Deep hypersphere manifold embedding for person re-identification," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 51–58, 2019.

[32] 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*, pp. 2197–2206, 2015.

[33] H. Yao, S. Zhang, R. Hong, Y. Zhang, C. Xu, and Q. Tian, "Deep representation learning with part loss for person re-identification," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2860–2871, 2019.

[34] 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)*, pp. 480–496, 2018.

[35] Y. Fu, Y. Wei, Y. Zhou, H. Shi, G. Huang, X. Wang, Z. Yao, and T. Huang, "Horizontal pyramid matching for person re-identification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8295–8302, 2019.

[36] F. Zheng, C. Deng, X. Sun, X. Jiang, X. Guo, Z. Yu, F. Huang, and R. Ji, "Pyramidal person re-identification via multi-loss dynamic training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8514–8522, 2019.

[37] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.

[38] T. Chen, S. Ding, J. Xie, Y. Yuan, W. Chen, Y. Yang, Z. Ren, and Z. Wang, "Abd-net: Attentive but diverse person re-identification," *arXiv preprint arXiv:1908.01114*, 2019.

[39] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," *arXiv preprint arXiv:1905.00953*, 2019.

[40] G. Wang, J. Lai, P. Huang, and X. Xie, "Spatial-temporal person re-identification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 8933–8940, 2019.

[41] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, Z. Hu, C. Yan, and Y. Yang, "Improving person re-identification by attribute and identity learning," *Pattern Recognition*, 2019.

[42] 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*, pp. 1318–1327, 2017.

[43] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *ACCV*, 2012.

[44] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[45] 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.

[46] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, no. 2, pp. 261–318, 2020.

[47] M. Wang and W. Deng, "Deep face recognition: A survey," *arXiv preprint arXiv:1804.06655*, 2018.

[48] G. Welch, G. Bishop, *et al.*, "An introduction to the kalman filter," 1995.

[49] S. Balaji and S. Karthikeyan, "A survey on moving object tracking using image processing," in *2017 11th international conference on intelligent systems and control (ISCO)*, pp. 469–474, IEEE, 2017.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[51] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*, pp. 499–515, Springer, 2016.

[52] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[53] 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.

[54] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

[55] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1314–1324, 2019.

[56] 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*, pp. 4510–4520, 2018.

[57] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[58] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical data augmentation with no separate search," *arXiv preprint arXiv:1909.13719*, 2019.

[59] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[60] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[61] W. Yang and Z. Jiachun, "Real-time face detection based on yolo," in *2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, pp. 221–224, IEEE, 2018.

[62] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned cnn embedding for person reidentification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, p. 13, 2018.

[63] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8697–8710, 2018.

[64] F. Chollet, "Keras." https://github.com/fchollet/keras, 2015.

Algorithm 1 Extract tracklets from set of bounding boxes

$Frames \leftarrow$ List of sequential frames for a single camera
$TrackedObject \leftarrow$ ArrayList containing a currently tracked object's box sequence
$TrackedObjects \leftarrow$ ArrayList containing all TrackedObject Lists
for all $frame \in Frames$ **do**
    $Boxes \leftarrow$ result of Object Detection on $frame$
    for all TrackedObject $t \in TrackedObjects$ **do**
        for all $box \in Boxes$ **do**
            $IOU \leftarrow$ IOU score of box with last box of t ▷ Intersection Over Union
            if $IOU \leq 0.6 \ \& \ box_{Class} == Person \ \& \ box_{AR} \leq 0.6 \ \& \ box_{width} > 60px$ **then**
                $MatchType \leftarrow$ MATCHBOXES(box, t, IOU)
                if $MatchType == "Tracked"$ **then** ▷ Matched and tracked
                    $TrackedObjects[t].append(box)$
                else if $MatchType == "Stationary"$ **then** ▷ Matched but stationary
                    $TrackedObjects.replace(t[lastBox], box)$ ▷ Update the stationary box's values
                else
                    $TrackedObjects \leftarrow newTrackedObject(box)$

procedure MATCHBOXES(box,TrackedObject t,IOU)
    if $IOU < MIN\_IOU$ **then**
        return NoMatch
    else if $IOU > MAX\_IOU$ **then**
        return "Stationairy" ▷ Subject moves little

    if $|t| \leq 2$ **then** ▷ If t is too small to apply Kalman Filter use the naïve approach
        $\Delta t \leftarrow$ time between measuring box and last box in t
        if $\Delta t \leq 1$ **then**
            return "Tracked" ▷ Found significant overlap within required time-frame
    else
        $IOU_{Kalman} \leftarrow$ IOU with predicted next box in t using Kalman Filter
        if $IOU_{Kalman} \geq MIN\_IOU_{Kalman}$ **then**
            return "Tracked"
        else
            return NoMatch ▷ No tracked match found during procedure

3

Background information

In this chapter some additional background information on techniques used in this thesis will be discussed. In section 3.1 the basics of deep learning are discussed. In section 3.2 some popular backbone architectures used in this thesis are discussed, next the optimization of networks is discussed in section 3.3. In the final section person re-identification is discussed.

3.1. Deep learning

Deep learning has been the backbone of the advancement of many applications, such as computer vision, natural language processing, and speech recognition. Deep learning is considered a part of machine learning but there are some differences. Deep learning models require more data, take longer to train and often do not contain elements of feature engineering. In deep learning the input is passed through multiple layers / a hierarchy of transformations, instead of having a single, linear formula that calculated the output directly. Often the networks used are referred to as deep neural networks(DNN) and the most basic unit in these networks is a neuron as shown in figure 3.1.

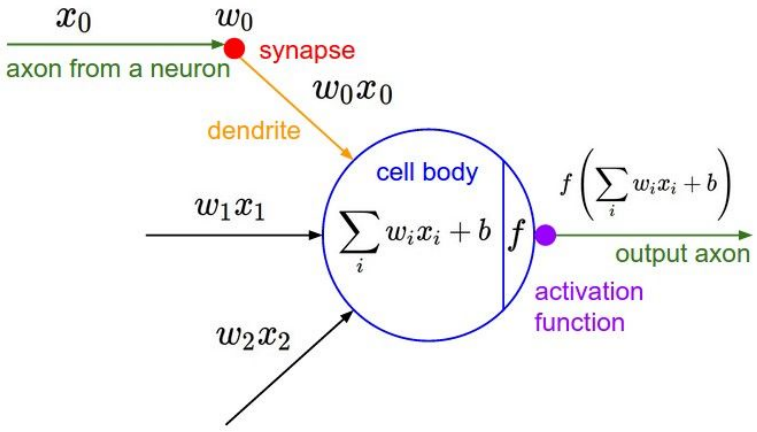


Figure 3.1: A single neuron as used in a neural network. Before the neuron, each input is multiplied with the corresponding weight and the results are summed. Next an optional bias term is added. Next the activation function is applied to the result. The output of this function is the output of a single neuron[3]

Convolutional Neural Networks (ConvNet) is a type of neural network that works well with images. Recently, ConvNet is the primary method for many computer vision tasks, such as image classification, image retrieval, object detection, and recognition.

3.1.1. Fully connected layer

A fully connected layer in a neural network connects all outputs of the previous layer to each neuron in the current layer. Each connection add a trainable parameter, the weight. The outputs of fully connected

layers are often used to classify images by using a softmax activation function.

3.1.2. Convolutional layer

Convolutional Neural Networks (CNN) are a type of neural network that works well with images. Currently, CNN's are the primary method for many computer vision tasks, such as image classification, image retrieval, object detection, and recognition. Meanwhile, a regular feedforward or fully connected neural networks are not used for dealing with image data because they do not scale well with images. At the core of these CNN's are the convolutional layers. In this layer a set of filters are used to slide over the input, and each of these filters produce a filter map as output. Convolutional layers can deal with inputs of many dimensions, but in this section we will focus on the convolutional layer that work with image data. Image data is often 2 dimensional data with 3 input channels(RGB). The convolutional filters will consist of a stack of kernels, equal to the number of channels from the input. The size of these filters differs greatly over the many available networks. The filters slide along the input width and height, producing a value for each location and mapping this to a feature output. In a convolutional layer these filters are the trainable weights. For each convolutional layer the amount of filters, the filter size, the stride and if padding is to be used. Padding is used to overcome border effects caused by the filter size, and the stride can be used to down sample the size of the feature map. An example of what the filters might look like is shown in figure 3.2.

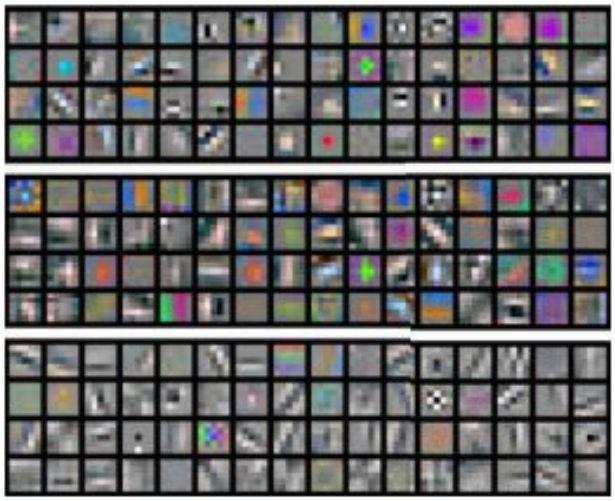


Figure 3.2: An example of what 64 5x5 convolutional filters look like[5], these are trained on different datasets.

3.1.3. Pooling layer

Pooling layers are an important part of neural networks, and are generally used for reducing the spatial resolution of the input from the layer. For filters this means the width and the height are scaled down. In a pooling layer a pooling region convolved over a region. The stride determines the step size when sliding the pooling region and therefore the reduction in resolution. A stride of 2 results in half the spatial resolution. We generally see two types of pooling; Average pooling and Max pooling, they are further explained in figure 3.3

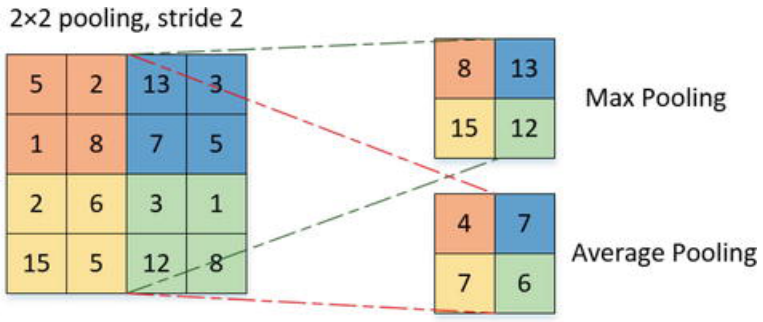


Figure 3.3: An example of pooling applied to a 4x4 filter, the difference between max and average pooling can also bee seen. For average pooling the average value of the pooling region is chosen, and for max pooling the largest value in the pooling region is used.

3.1.4. (Batch)Normalization

More recent CNN's, like ResNet50[10] use batch normalization. Batch normalization for each mini-batch, normalizes the output of the previous layer using the mini-batch mean and standard deviation. If batch normalization is applied higher learning rates can be used, because batch normalization makes sure there are no activations with very high or low values. In layer normalization, the statistics are computed across each feature over the entire dataset, and are independent of other examples.

3.2. Network optimization

Training or optimizing a network for a certain tasks is done by maximizing/minimizing a cost/loss function. Two important choice must be made; which function is going to be optimized?, and how is this function going to be optimized. First some popular loss functions will briefly be explained and next the optimizers will be discussed.

3.2.1. Loss Function

The performance of any model will greatly depend on the choice of loss function. Here two types of loss functions will be discussed. The cross-entropy loss and the triplet loss. For these loss functions the logits are often used, the logits the output of the final layer of the network.

Cross-entropy loss

Cross entropy loss is the most applied loss function for classification problems. A classification problems samples contain the truth labels and the prediction vector. This prediction vector is created by passing the logits through a softmax activation function, this turns them into class probabilities. Often the final layer of a classification network contains 1xN values, where N is equal to the amount of classes. The formulae for the cross entropy loss is:

$$L_{Crossentropy} = -\frac{1}{N}\sum_{i=0}^{i=N} Label * \log(Pred) \tag{3.1}$$

Triplet loss

The triplet loss can be applied directly to the logits, or the output of the final layer before the softmax activation. Here the images are represented on an hyperplane by the 1xN dimensional feature vector. An illustration of the triplet loss is shown in figure 3.4, here the goal is to pull positive matches closer together while pushing away negative matches.

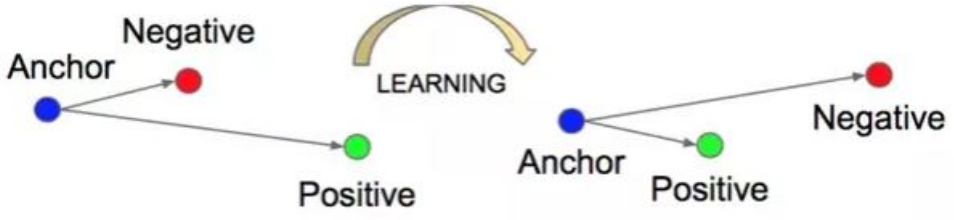


Figure 3.4: Triplet loss optimization [26]

The triplet loss is formulated as:

$$L_{triplet} = \sum_{a,p,n} [m + D_{a,p} - D_{a,n}]_+ \tag{3.2}$$

Here $D_{a,p}$ is the distance between the positive match and the anchor, and $D_{a,n}$ is the distance between the negative match and the anchor. The m is the margin which is a hyperparameter. Several other formulations of the triplet loss exist, these other formulations are used to combine the triplet loss with other loss functions when working with batches of images.

Center loss
the center loss is often added to maximize intra-class compactness, it is mainly popular in the facial recognition scene[33]. The formula for center loss is:

$$L_{center} = \frac{1}{2} \sum_{j=1}^{B} |L_j - C_j|^2 \tag{3.3}$$

L represents the logits/embedding for a certain input image, C is the center of the matching identity in the training set and B represents the batch size. This function learns a center for each class and calculates the euclidean distance of the sample to its center. The sum of these distances multiplied by 0.5 equals the center loss. Minimizing center loss will increase intra-class compactness, meaning positive pairs are pulled closer together. The center loss can be combined with the triplet loss to make the result more robust. A disadvantage of the center loss is that it is sensitive to outliers.

3.2.2. Optimizers

Besides selecting a model, it is also important to select a suitable optimizer for training the model. Stochastic gradient descent (SGD)[23] is a popular choice as optimizer, mainly the mini batch is used a lot. The standard SGD algorithm updates the weights after each sample, while the mini-batch variant updates the weights after each mini-batch. This means the standard SGD path to the minima is noisier (more random) than that of the mini-batch gradient. SGD minimizes a loss function by updating the parameters in the opposite direction of the gradient of the loss function. Several derivatives of the standard SGD algorithm exist. Often these derivatives show improved performance while requiring less settings to be fine tuned. An overview is created by Ruder et al.[24], here we will briefly discuss the most common ones.

- **Momentum** [21] adds a term to the standard SGD, which is the momentum term. The momentum term is based on exponentially weighted averages of the previous gradients, and together with the calculated gradient it calculates the step update.

- **Adagrad**[7] adapts the learning rate to the parameters, thereby performing larger updates for infrequent and smaller updates for frequent parameters. It is therefore suitable for usage with sparse data. A problem with Adagrad is that it is designed in such a way that its learning rate converges to a zero. When this happens the model stops learning.

- **Adadelta**[35] is an extension of Adagrad that prevents the learning rate from converging to 0. When using Adadelta we do not even have to specify a learning rate, it finds a suitable learning rate by itself.
- **RMSprop**[30] is similar to Adadelta as in that it combats the ever decreasing learning rate problem of Adagrad. It's an extension of Adagrad but it deals with the problem of the low learning rate in a different manner.
- **Adam**[13] uses an adaptive learning rate like Adadelta and RMSprop, but also uses a momentum term.
- **Adamax**[13] is an extension of Adam which aims at making the optimizer more stable and more suitable for sparse data.
- **Nadam**[6] combines Nesterov momentum with RMSprop, as Nesterov momentum generally gives better results than standard momentum. It can therefore be seen as an updated version of AdaMax.

  An example of the optimization using these different optimizers is shown in figure3.5. In this case a simple CNN was optimized on the MNIST[15] dataset, which consist of handwritten digits.

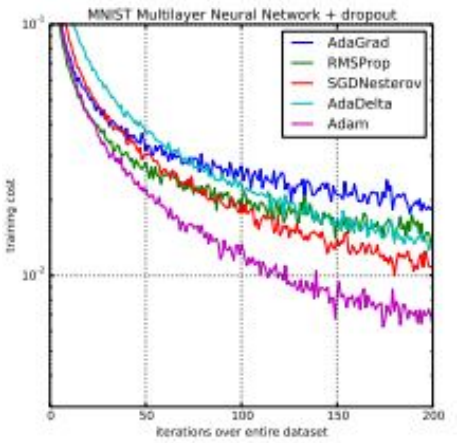


Figure 3.5: Different convergence speeds and final cost values for some of the optimizers[13]

3.2.3. Fine-tuning backbone network

When a task introduced for a deep learning network, the first thought would be to train it from scratch. However, in practice these deep neural nets have a huge number of parameters. If we train these neural nets with huge amounts of parameters on a relatively small dataset it would greatly affect the neural networks ability to generalize well, and often result in overfitting. Therefore, in practice often these large networks are often pre-trained on a large dataset, like ImageNet[4]. Another reason for not training a network from scratch is that it takes up alot of time, for example the ResNet50[10] network takes 14 days to train on the ImageNet dataset. The next step is to take the pre-trained network and fine tune it on the smaller dataset. Often the first step is adjusting the final layer of the network, since the new task will almost always contain a different amount of classes. While fine-tuning often a small learning rate is used, since we do not want to distort them too quickly and too much. A common practice is to make the initial learning rate ten times smaller than the one used for scratch training with randomly initialized weights. Also, it is common practice to freeze the weights of the first few layers of the pre-trained network. This is because the first few layers capture universal features like curves and edges that are also relevant to our new problem. We want to keep those weights intact. Instead, we will get the network to focus on learning dataset-specific features in the subsequent layers. The pre-trained networks weights are available through the keras library, or authors upload them on their github pages.

The advantage of using one of these optimizers over SGD is that these optimizers do not require the learning rate to be specified manually during training. During our research we chose to use the Adam optimizer[13] as this optimizer is the most popular in the re-identification field. We did not investigate the effect of changing the optimizer.

3.3. Backbone architectures

The choice of backbone architecture is an important factor for any deep learning model, here three types of backbone architectures are briefly explained. One of the largest scale classification challenges available is the ImageNet[4] challenge. Here there are over a million images available for 1000 different classes.

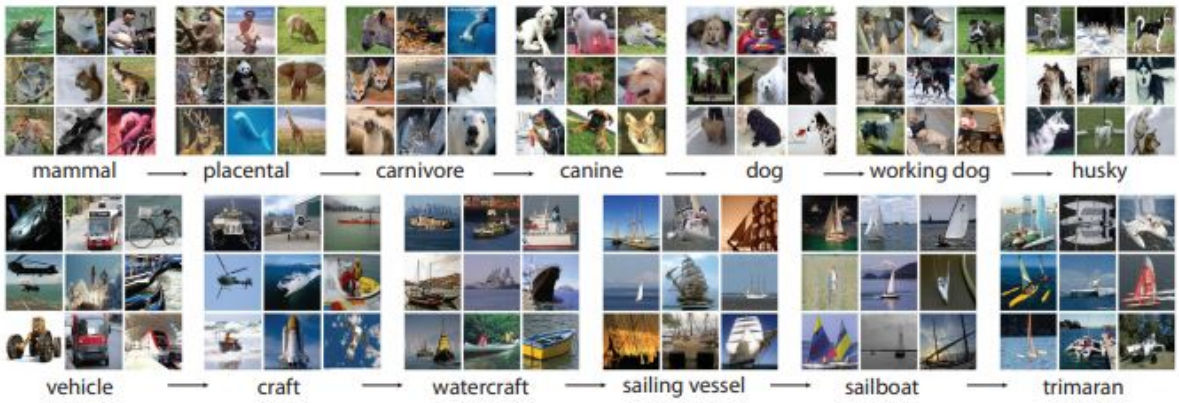


Figure 3.6: Some example classes of the ImageNet dataset[4]

In 2012 the state of the art model, which was created by Krizhevsky et al.[14], achived a top-5 test error rate of 15.3%. The second best entry achived a top-5 test error rate of 26.2%. Since then the deep convolutional neural networks have evolved rapidly. Currently the best performing model is the EfficientNetB7[28] which achieves a top-5 test error of 1.9%. These networks are often adapted for many other deep learning challenges, and in this section a few of the networks which seem interesting for person re-identification are discussed.

3.3.1. ResNets

The Resnet architecture consists of many building blocks, which vary depending on which architecture is chosen. In their paper He et al.[10] introduce varying depths from 16 to 152 layers. The core idea of ResNet is the skip connection and the use of batch normalization. This skip connection is shown in figure 3.7 and its goal is to counter the problem of the vanishing gradient, which was common for very deep networks. The vanishing gradient is the problem that the gradient shrinks to zero for deep networks as it is back-propagated. If the gradient shrinks to zero the model is unable to learn. This skip connection allows the gradient to skip layers and this stops the gradient form shrinking to zero.

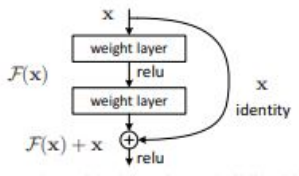


Figure 3.7: Residual learning: a building block[10]

Different network depths are suggested by the authors, they are shown in figure 3.12. These networks are built with several blocks of convolutions. The filter size and amount of filters is shown for each block of convolutions.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2 [3×3, 64 3×3, 64] ×2	[3×3, 64 3×3, 64] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3
conv3_x	28×28	[3×3, 128 3×3, 128] ×2	[3×3, 128 3×3, 128] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×8
conv4_x	14×14	[3×3, 256 3×3, 256] ×2	[3×3, 256 3×3, 256] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×23	[1×1, 256 3×3, 256 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512 3×3, 512] ×2	[3×3, 512 3×3, 512] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 3.8: Architecture of Resnet18, Resnet34, Resnet50, Resnet101 and Resnet152[10]. The networks are built with several blocks of convolutions.

The Resnet architecture won the Imagenet challenge in 2015.

3.3.2. MobileNets

MobileNets are class of efficient models for mobile and embedded vision applications. MobileNets are based on depthwise separable convolutions with the goal of building light weight deep convolutional models. So far there are three MobileNet versions available, MobileNetV1, MobileNetV2 and MobileNetV3 where V1 is the first version and the later versions can be considered improved versions.

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

Figure 3.9: MobileNetV1[12] architecture

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	-	

Figure 3.10: MobileNetV2[25] architecture

Input	Operator	exp size	$\#out$	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Figure 3.11: MobileNetV3[11] architecture. Here NL denotes the nonlinear activation function used, HS denotes h-swish and RL denotes RELU.

The architectures of the MobileNets are shown in figures 3.9, 3.10 and 3.11. The MobileNetV2 architecure uses two different blocks

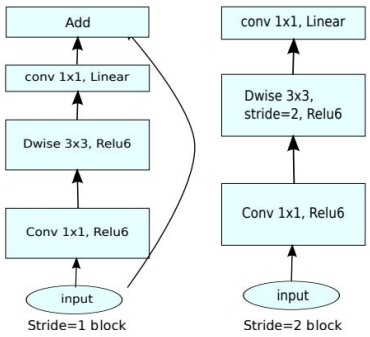


Figure 3.12: Bottlenecks used in MobileNetV2[25]

MobileNetV3[11] applied AutoML to find the best possible neural network architecture for a given problem, which is a type of reinforcement learning. They also added the swish activation function to some of the layers. All of the MobileNet architectures are designed for use on mobiles and embedded hardware, and they have a maximium of 5 million parameters. This makes them interesteing networks to use on board of camera systems for re-identification.

3.3.3. EfficientNets

EfficientNets were first introduced by Tan et al.[28], they showed that instead of just scaling networks in depth it is also useful to scale the network in width and image resolution, these scaling methods are visualized in figure 3.13. They created different scales of models ranging from B0 to B7 and from 5.3m parameters to 66m respectively. The EfficientNets largest variant is currently the state of the art model for the ImageNet challenge. The smallest variant EfficientNetB0 is able to outperform ResNet50 on the Imagenet challenge, while using more than 5 times less parameters. This makes it at interesting network for embedded hardware. Tan et al.[28] found their baseline architecture by using neural architecture search(NAS), and optimizing for accuracy and FLOPS.

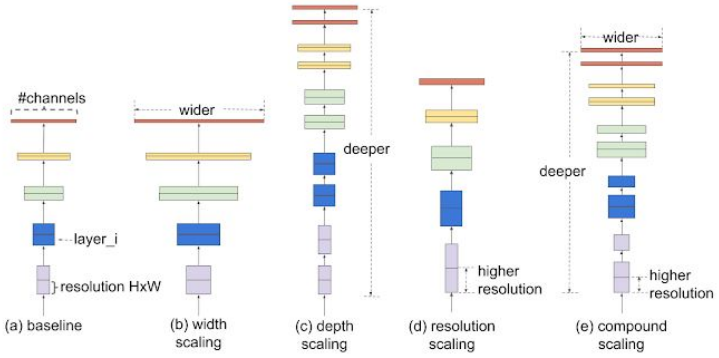


Figure 3.13: Different scaling methods used for creating the EfficientNet architectures[28]

3.4. Person re-identification

In this section person re-identification will briefly be discussed. Person re-identification as a task is quite simple to understand. As humans, we do it all the time without much effort. Our eyes and brains are trained to detect, localize, identify and later re-identify objects and people in the real world. Re-identification implies that a person that has been previously seen is identified in their next appearance using a unique descriptor of the person. Humans are able to extract such a descriptor based on the person's face, height and built, clothing, hair color, hair style, walking pattern, etc. A person's face is the most unique and reliable feature that humans use to identify each other. The field that is focused at recognizing faces is called facial recognition. The person re-identification field has alot of similarities with the facial recognition field, but both fields deal with different challenges. Some challenges seen

in re-identification are changes in color, lighting, view angle, pose of individuals, low resolution and (partial) occlusion of individuals. A few examples are given in figure 3.14.

Figure 3.14: Challenges (left to right): low resolution, occlusion, viewpoint, pose, and illumination variations and similar appearance of different people[34]

A typical end to end re-identification pipeline is given in figure **??**. First the person must be detected on an input video or image, then this person is tracked through a sequence of input images and tracklets are created. Tracklets are multiple image crops containing a certain person. In existing re-identification datasets these tracklets are already created. The next step is extracting features from these input images and matching them based on similarity. This is done by means of a backbone network. These backbone networks extract features from the input images and based on these features images are matched. In re-identification an input image is often compared to a gallery of images, the size of this gallery varies greatly over the different datasets. The images of the gallery will then be ranked based on similarity. An example of this ranking is shown in figure 4.6, here the images are ranked based on euclidean distance.

Figure 3.15: All of these images have a euclidean distance smaller than 0.6 and are classified as matches.

What makes re-identification challenging is that the people in the training set are often not part of the test set. This means re-identification is not a typical classification problem and often during inference a distance metric is used to express similarity of the images.

3.4.1. Part based vs global models

Two approaches can be distinguished in person re-identification, part-based models and global feature models, and they differ in the feature extraction part. Global features are learned from the entire image

and intend to capture the most discriminative features of appearance but may fail to capture discriminative local features. Therefore local features may be used so that local discriminative features can also be captured. Combining local and global features is a popular approach[17, 27, 31, 36, 37, 40] some authors randomly divided the image into parts[17, 27, 31] and extracted local features from these image parts and combine them with the features from the entire image. Others[36, 37] extract information about the pose and used this information to find certain body parts subsequently extracting features from these body parts. Other authors added external information to the images like attribute labels[19] or spatial temporal information[32]. Most researchers only use the global features/feed the entire image into their model.

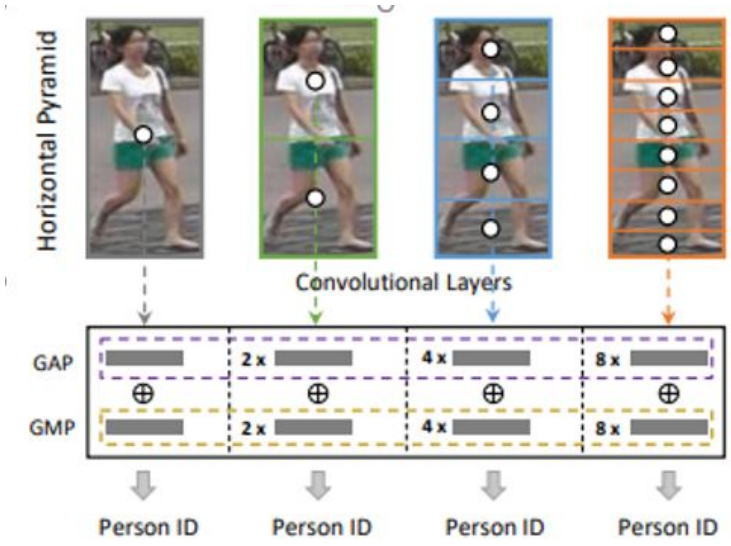


Figure 3.16: Zheng et al.[38] split a person into different horizontal parts of multiple scales. The feature representations produced by Global Average Pooling (GAP) and Global Max Pooling (GMP) of each part are then treated for re-identification independently.[38]

3.4.2. Datasets

Re-identification datasets contain three parts; A training set, a query set, and a gallery set. The size of the datasets differ greatly, and all of these datasets mainly contain pedestrians. In our case is to re-identify cyclists thus a new dataset must be created, which contains cyclists. An overview of current benchmarks datasets for re-identification is shown in Table 3.1.

Dataset	BBoxes	Identities	Detection method	# Cam
CUHK03[18]	28,192	1467	DPM, hand	2
CUHK01[16]	3884	971	hand	10
Duke MTMC[22]	36,411	1812	hand	8
ViPer[9]	1264	632	hand	2
Market1501[39]	32,364	1501	DPM, hand	6

Table 3.1: Re-identification benchmark datasets. Detection is done by hand or with the deformable parts model as described in [8]

Currently the most popular benchmark dataset for re-identification is the Market1501[39] dataset, a few examples are shown figure 3.17. Another popular benchmark dataset is the Duke-MTMC dataset[22], but this one was discontinued by the authors due to privacy violations of the people in the dataset[20]. This opens an interesting discussion; Cant re-identification be applied to anonymized persons? This question is further investigated in this work.

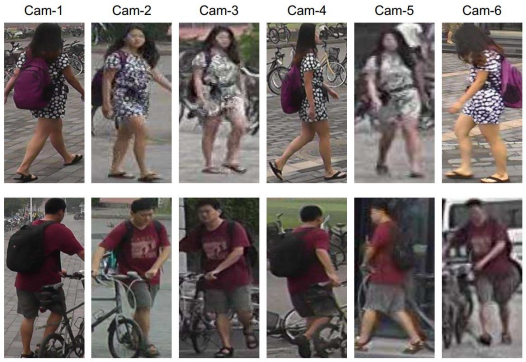


Figure 3.17: An example from the market dataset, this is the same person in the six different camera views.

3.4.3. Open vs closed-world

Generally speaking there are two approaches in re-identification, open and closed world situations. In a closed world situation, the assumpion is made that each identity in the query set can also be found in the gallery set. In real world situations this is alsmost never the case. In a closed world scenario the image with the smallest euclidean distance is chosen as a match, while in an open world scenario a certain threshold is chosen. Images that have a smaller euclidean distance than the threshold are considered to be a match, and if the euclidean distance is larger than the threshold the images is described as a negative match. Both cases also have their own scoring metrics, which are briefly explained below.

Closed-world

The cumulative matching curve (CMC) and the mean average precision (mAP) are used as the benchmark scoring metrics in person re-identification in a closed-world setting. The CMC measures the retrieval precision, given the list of closest matches from the gallery, the rank-x indicates the probability that the matching image is among the first x positions. The rank-1 accuracy gives the probability that the top ranked image is a match. In early research in person re-identification only the CMC scores were reported. But with growing datasets this score on its own is not a good metric of the models performance anymore thus the mAP score was added. The mAP takes into account the location of all matching images from the gallery. The mAP is calculated by first calculating the Average Precision (AP). Then the mean of all the average precisions is calculated and this forms the mAP. In figure 3.18 a calculation example of these scoring metrics is given. The mAP gives a better indication of the performance since it takes the position of each matching person into account.

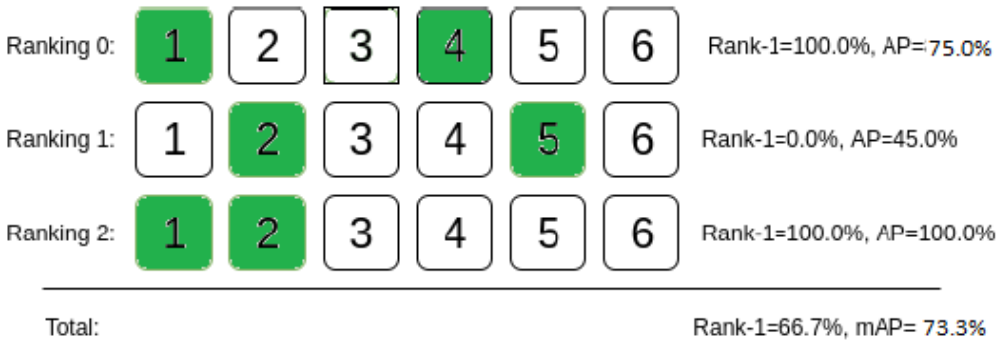


Figure 3.18: Example scores for rank1 and average precision. Only using the rank1 accuracy does not give a good indication of the performance if a large gallery set with multiple matches is used.

Open-world

In the real world application the gallery size may vary in size and not every query image has a match in the gallery set. The scoring metrics used for closed-world re-identification would not give a good indication of the models performance. Therefore the open world re-identification problem is viewed as

a binary classification problem. The model must classify if the query image matches with an individual gallery image or not based on the euclidean distance. The F1-score can be used as a scoring metric for open-world re-identification. First the precision and recall are calculated, next these can be used to calculate the F1-score. The precision and recall are defined in equation 3.4.

$$Precision = \frac{TP}{TP + FP} \text{ and } Recall = \frac{TP}{TP + FN} \tag{3.4}$$

With TP = True positives, FP = False positives and FN = False negatives. Next the F1-score is calculated as:

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \tag{3.5}$$

The F1-score helps us find a balance between the precision and recall, a value of 1.0 portrays a perfectly working system.

3.5. Object detection

To create a dataset of cyclists seen in different camera views, first object detection must be applied to the frames. Object detection algorithms take a video frame as an input and returns the locations, and classes of objects which have been detected in the frame. An example of objects detected in a video frame are shown in figure 3.19.

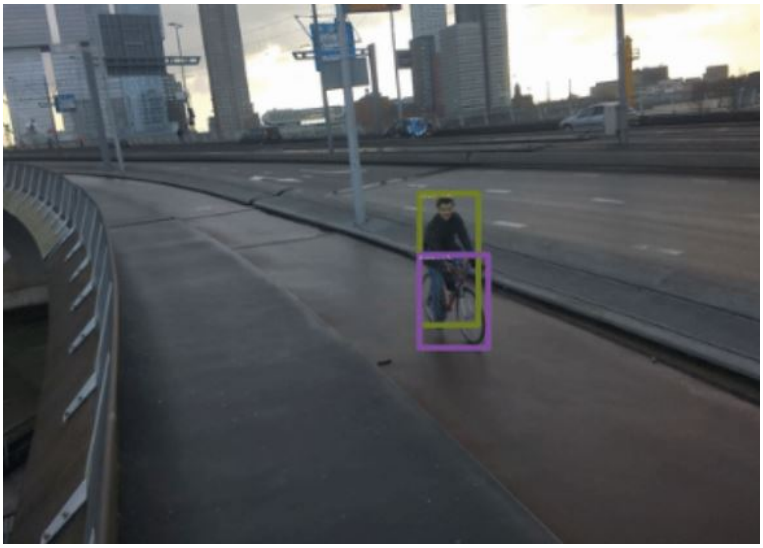


Figure 3.19: Object detection applied on Rotterdam video footage

Generally object detectors consist of two parts, a region proposal network, and a classification network. The region proposal network is used to find regions in the input image, which are then fed to the classification network, which gives a classification output.

3.6. Blurring

The computer vision library, OpenCV[1] offers three types of blurring that can be used to anonymize data; Average blurring, Gaussian blurring and Median blurring. In the process of blurring a filter of size N is convolved over the image and a new blurred image is calculated. The difference between the types of blurring lies in the type of filter used. For average blurring simply the average value of the pixels under the filter is calculated, for a Gaussian filter a Gaussian kernel is used as filter and for Median blurring simply the median value of all pixels under the kernel is calculated. An example person from the Market1501[39] is shown in 3.20.

Figure 3.20: Types of blurring, from left to right; Original, Average, Gaussian and Median.

4

Supplementary figures

Figure 4.1: An example of re-identification from our blurred dataset, the green boxes contain the correct match while the red ones are mismatches. The distance mentioned above the images is the euclidean distance, the smaller the euclidean distance the more similar the images are.

Figure 4.2: Example of perfectly working system, where the person is re-identified over three cameras and all others are discarded as non matching

Figure 4.3: Example of model recognizing red scooters

Figure 4.4: Example of difficult situations; a very unclear person and low illumination conditions

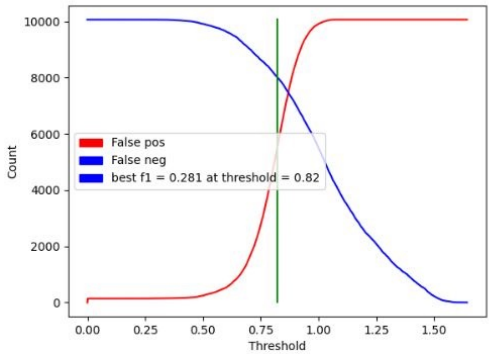


Figure 4.7: F1 score of best setup with batch all triplet loss

Figure 4.5: All of these images have a euclidean distance smaller than 0.6 and are classified as matches.

Figure 4.6: An example of re-identification from our blurred dataset, the green boxes contain the correct match while the red ones are mismatches. The distance mentioned above the images is the euclidean distance, the smaller the euclidean distance the more similar the images are. Here we see two distractors as input, and the returned euclidean distances are larger than 0.6 so these are correct.

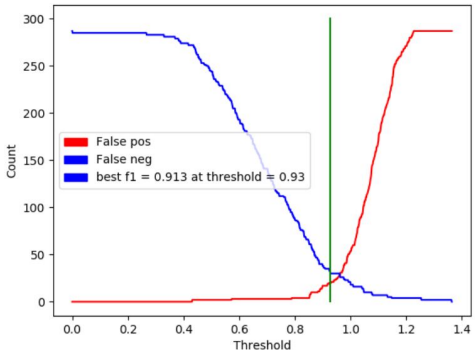


Figure 4.8: F1 score of best setup with batch all triplet loss and averaging of embeddings

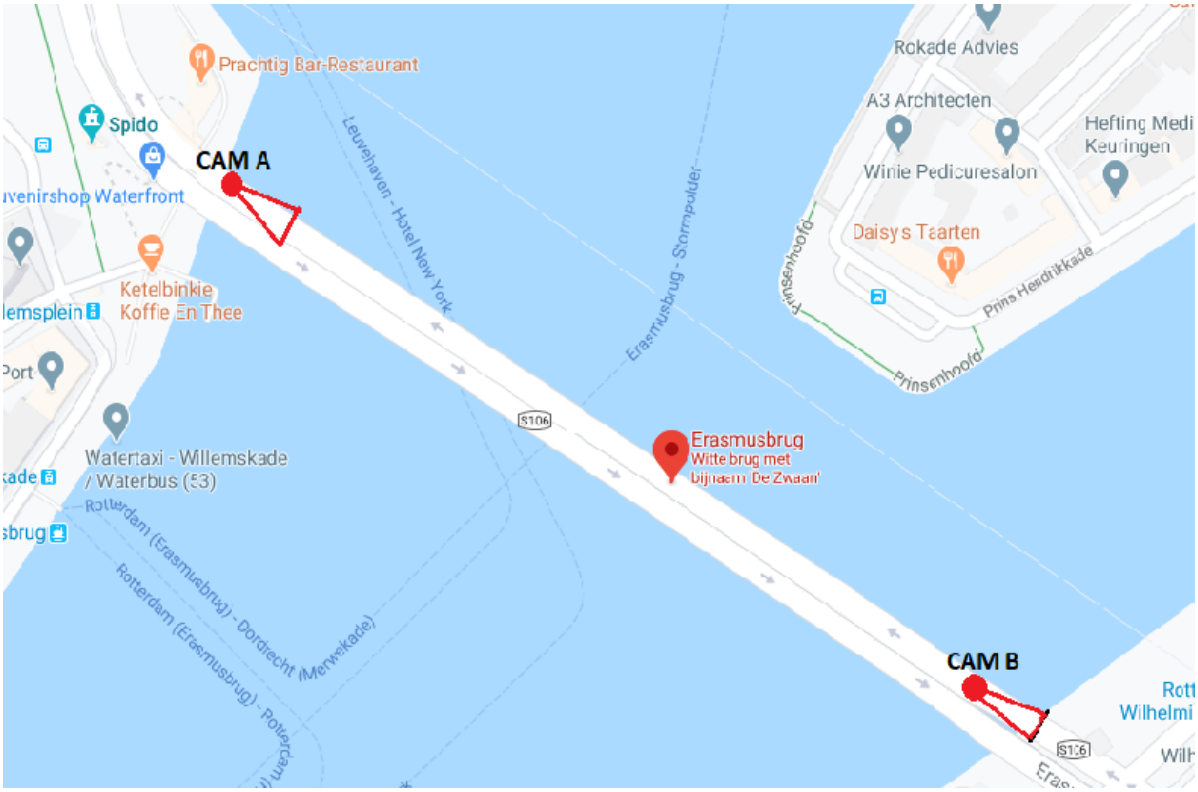


Figure 4.9: Camera setup in Rotterdam

5

Literature Report

This literature report is included for the interested reader on the topic, who wants a more elaborate overview of approaches takes in the re-identification field. It has been graded separately from the thesis.

REVIEW OF DEEP (OPEN-WORLD) RE-IDENTIFICATION

LITERATURE REVIEW

Michael Schoustra
3mE, Mechanical, Maritime and Materials Engineering
Delft University of Technology
Leeghwaterstraat, 2628 CN Delft
mikeschoustra@gmail.com

May 8, 2020

ABSTRACT

Person re-identification has seen fast-paced improvement in recent years. This review summarizes recent literature on the person re-identification field and compares the performance achieved on certain benchmark datasets. Some of the state of the art performing research on facial recognition was also included in this review as their approaches might also be applicable to the person re-identification field. The state of the art systems are all based on deep CNNs. Authors choose different approaches to solve the person re-identification task. Taking architectures that perform well on the Imagenet challenge and applying them to person re-identification has proven beneficial. The choice of features which are fed into the network is also of great importance. During training the choice of loss function is essential for the performance of the model; it teaches the backbone network which features should be extracted and which values the features should have. Two main loss functions could be distinguished; ID/verification loss and distance metric-based loss. Often these two losses are combined and jointly optimised. A lot of different heuristics can be applied to the data to increase the accuracy achieved on a certain dataset; however the effectiveness of each particular heuristic is hard to prove. Most of the research is focused on a closed-world setting as it is easier to solve. Also the researchers can easily compare their solution if they use the same benchmark dataset as others. Recently some models have been created to solve the open-world problem, however there are very few models available. Systems that perform well on a closed-set will also be able to perform well on open-world settings.

***K*eywords** Person Re-identification · Open-set · Deep Learning

1 Preface

The FlowCube is a product being developed by Technolution[1]. It is a single-box traffic sensor that aims to replace different conventional traffic monitoring systems (e.g. traffic radars, floating car data, induction loops), based on computer vision and edge AI. One of the initial use cases for FlowCube is measuring travel times and route selection of cyclists, for which there is currently no effective solution. There is also a strong societal urge to stimulate the usage of bicycles[2].

One of the purposes of this product is to map the route choices made by cyclists on the road. This information can be extracted by automatically matching the cyclist in different video streams. Matching the same cyclist in multiple video streams is known as re-identification. This is relevant because we want to create an automated system. An initial model to solve the cyclist re-identification was created by Technolution, it is desirable that this model is improved so that the product works better.

Since no dataset of cyclists exists to test the solution on, a dataset of labeled cyclists was created from video data of cyclists in Copenhagen, Rotterdam and Groningen. The ground truth of this data is known thus we can verify the effectiveness of our model.

It is required that the system has semi real-time performance and it must function as an embedded system. If a cyclist is detected in an image, the features of this cyclist will be extracted by a network and communicated towards a main server. The main interest is the path and flow data of the cyclist and once the features have been extracted the images can be destroyed thus the privacy of the individual cyclists can be respected.

2 Introduction

In recent years, person re-identification has become increasingly popular in the research community due to its application and significance. The re-identification task can be summarized as matching a pedestrian from an input image to a gallery set which has been captured by different cameras. It is a challenging issue since the appearance of a pedestrian can be very different in different camera views. Some challenges seen in the person re-identification task are the variations in lighting, view angle, pose of individuals, low resolution and (partial) occlusion of individuals(Leng[3],2019). Also the fact that some individuals may only appear in a single camera view, making it impossible for them to be re-identified. Recently models have learned to deal with these challenges and are able to outperform humans on certain benchmark datasets.

All of the state of the art systems can be considered deep re-identification networks, they are the most successful systems to solve the person re-identification challenge. Recently we have seen a huge increase in accuracy on various challenging datasets. However, a of the research in the re-identification task is focused on pedestrians, but how can this research be applied to other traffic participants. Another issue which isn't often addressed is the open/closed world concept. In a closed world every person can be re-identified as all of them appear in multiple cameras. However in the real world people often do not reappear in the camera's view.

The best performing models in the person ReID task all use deep architectures. Recently we have seen the Rank1 accuracy rise from 35,68% in 2015 by Zhang et al. [4] who used the null Foley-Sammon transform [5] to learn a null space in which image embedding are represented to 98.0% in 2019 by Wang et al. [6] who used spatial-temporal information to remove lots of irrelevant images from the search query. These accuracy's were achieved on the Market1501[7] dataset.

This literature review provides an overview of current methods for person re-identification. Several other surveys exist[8, 9, 10, 11]. These surveys focus on a wide range of approaches for person re-identification, in this survey we mainly discuss the deep learning based models as these are the most promising for solving the person re-identification task and will most likely be the models chosen for future work. In the surveys[9, 10, 11] deeply learned systems are put into a single category while there can be major differences between them. None of the previous surveys mention the heuristics used during training whilst these have great influence on the final accuracy achieved. In this survey a more in depth explanation of deeply learned person re-identification models is given and each model is described by four design choices. By categorizing person re-identification models in these four categories we can get a broader understanding in the effectiveness of the adjustments which researchers make to recent models.

3 Definitions

- Re-identification: According to Gong et al.[9] the re-identification pipeline can be summarized as follows: First features must be extracted from an input image, with these features a descriptor must be constructed and finally the probe and gallery images must be defined and matched based on the created descriptor. Extracting features from images is also very important for the tasks of object detection/recognition and since the introduction of convolutional neural networks(CNN) to the image recognition task they have dominated the field. They were first introduced by Lecun et al.[12]. For the person re-identification task we also focus on deeply learned systems as these are currently the most promising systems for solving person re-identification.
- Open and closed-set: As stated in the introduction most of the research done in the person re-identification field is applied to a closed-set. In a closed-set the assumption is made that every person which has been identified can be linked to a person in a image gallery. For a closed-set system the objective question is 'Which image from the gallery matches with the probe?' this obviously is not representative of a real world scenario where people can appear and disappear in every camera view. In an open-set not every person which has been identified can be re-identified as they may only appear in a single camera shot. Here the objective question is 'Does the probe appear in a certain gallery and in which images?'. The results achieved by models on closed-sets are represented in the Rank1 accuracy(%) which represents the odds of the correct match being the first ranked match in the query. The mean average precision(mAP) is also used for the closed-set re-identification:

$$mAP = \frac{TP}{TP + FP} \tag{1}$$

TP = True positive and FP = False positive. For open-set problems the most used evaluation terms are TTRs (True Target Rates) at certain FTRs (False target rates).

- Datasets: Several benchmark datasets exist in the person re-identification field, in this review we will mainly compare the scores that the models achieved on Market1501[7], DukeMTMC[13], CUHK01[14] and VIPeR dataset[15]. Details about these datasets are summed up in table 3. These images are taken from at least two different camera streams and up to 8 different camera streams. As can be seen in the table the amount of available data differs greatly per dataset, having less images to train on makes a dataset more challenging.

Dataset	No of Identities(train/test)	No of cameras	No of images(train/test)
Market1501[7]	751/750	6	12,936/19,732
DukeMTMC[13]	702/702	8	16,522/2,228
CUHK01[14]	961	2	3884
VIPeR[15]	632	2	1264

Table 1: Closed-set ReID benchmarks

- Feature extractor: One of the most successful convolutional neural network architectures for feature extraction are residual networks. Soon after the first success of convolutional neural networks the scientific community realized that it was necessary to have deeper networks to avoid overfitting. However, stacking more layers in a network led to the vanishing gradient problem. A vanishing gradient occurs, when back-propagating through a lot of layers with repeated multiplications. This makes the gradient extremely small and learning is not possible anymore. First introduced by He et al.[16], Residual neural networks(Resnet) were invented to tackle the 'vanishing gradient' problem by adding a identity shortcut to skip layers in the network. Many state of the art systems use Resnets.
- Loss function: The performance of the model will greatly depend on the choice of loss function. For training the network a function is defined which should be minimized or maximized. These loss functions differ greatly in their complexity. Also a function should be chosen which can be optimized otherwise the network will not be able to find an optimal solution. Different optimizers exist which are used to minimize the loss function and find the optimal solution, but these are beyond the scope for this review. In the person re-identification field we generally see two types of loss functions: softmax loss and metric-based loss. Softmax loss teaches the backbone architecture to output similar embeddings for similar persons. Distance metric loss teaches the backbone architecture to output embedding which have a small euclidean distance between the matches and a larger distance between false matches.

4 Methods for Deep person re-identification

In this section we provide an overview of models in the person re-identification field. Also some of the state of the art performing models in the facial recognition field are summarized. The initial systems in person Re-identification used hand crafted features from the computer vision field, nowadays all of the state of the art performing models use either deep features or a combination of deep features and hand-crafted features. To get an idea of the maximum achievable rank accuracy using non deep systems the following paper was included;

Wu et al.[17] created one of the best models that do not use CNNs in their approach. They argue that CNNs need large amount of training data which is not available in the person re-identification domain. They extract SIFT and color histogram features from an input image and subsequently apply principal component analysis(PCA) to reduce the dimensions. Next they use a Gaussian mixture model(GMM) to construct fisher vectors from the features which were extracted. They train their network using their own LDA(linear discriminant analysis) based loss function which enforces the feature vectors to become linearly separable. Their method achieves a rank1 accuracy of VIPeR[15] 44.11%, CUHK01[14] 67.12% and Market-1501 48.15%. Mainly the score achieved on the VIPeR[15] dataset is good, this dataset contains only two images per person.

4.1 Deep person re-identification

Before the rise of convolutional neural networks(CNNs), features used to be extracted from the images using hand crafted features from the computer vision field. Later CNNs were widely deployed in the field of image recognition, for example to detect pedestrians [18][19] and also on the Imagenet[20] challenge which contains over 14 million images of 20.000 categories. The state of the art performing models on the Imagenet[20] challenge all deploy a variant of CNNs, since the task of extracting features is an overlapping area between person re-identification and image recognition the

same type of systems can be used. In this chapter a distinction will be made between id/verification based approaches and distance metric based approaches. Also authors that created a system for an open-set and added external data are separated.

4.1.1 ID/verification based approach

Yi et Al.[21] introduce a siamese convolutional neural network(SCNN), where several CNNs are trained and the weights are shared. They first they divide the input image into three overlapping segments, for the head, torso and legs. They train the SCNN on each of the parts and then for each of the parts calculate the cosine difference between the pair of input images. The summation of these seperate similarity values indicates the similarity between the input images. They use a simple CNN with 5 layers to extract the features. They achieve an rank1 accuracy of 34.4% on the VIPeR[15] dataset.

Ahmed et al.[22] slightly improve the SCNN model introduced by [21] instead of looking at the similarties between the pair of input images they look at the differences. They use the global image features in their work and outputs a verification score thus if the pair of images belongs to the same class or not. They achieve a rank1 accuracy of 34.84% on the VIPeR[15] dataset and a rank1 accuracy of 65.0% on the CUHK01[14] dataset. The results reported are slightly better than Yi et al.[21] but they also added some augmentation to their data thus it is unknown which change is beneficial to the achieved accuracy.

Wu et al.[23] improve the model of Ahmed et al.[22] by using more convolutional layers with smaller filter sizes, which at that time was also the trend for state of the art performing models on the Imagenet[24] challenge. They use the same data augmentation as Ahmed et al.[22] and also suggest using a different optimizer to train the network might be beneficial to the final accuracy achieved. They achieve a rank1 accuracy of 71.14% on CUHK01[14] and 37.21% on Market1501[7].

Varior et al.[25] investigated the use of a gated function to improve the CNN model for person re-identification. They argue that features in the middle levels of a CNN might also be useful to compare between the input images to achieve this comparison they propose using a gating function between layers which compares the features of the SCNN at each layer. They apply the same data augmentation and optimization strategy as Wu et al.[23]. They achieve a rank1 accuracy of 65.88% on Market1501[7] and 37.8% on the VIPeR[15] dataset.

Zhao et al.[26] introduce Spindlenet, they use a CNN to find 14 body joints and combine these into 7 sub regions. Combined with the original image they now have 8 regions from which they extract features using a 5-layer CNN network and combine these into one feature vector using their Feature Fusion Net(FFN). They achieve a rank1 accuracy of 76.9% on the Market1501[7] dataset.

Zheng et al.[27] suggest that combining the identification and verification loss might be beneficial to the networks performance. They try different backbone CNNs; CaffeNet[20], VGG16[28] and Resnet-50[16]. They show that Resnet50[16] achieves the highest scores across multiple datasets and show that combining the verification and identification loss results in a rank1 accuracy of 79.51% on the Market1501[7] dataset. Only using verification or identification loss would have resulted in 64.58% and 73.69% respectively.

Sun et al.[29] introduce SVDNet, the goal of SVDNet is to tackle the correlation between the weights in the final fully connected layer. They use singular value decomposition(SVD) which is a similar technique as Principal component analysis(PCA) to make the weights less correlated. They perform ranking based on the euclidean distance between the obtained features. Their best performing setup, using Resnet50[16] as a backbone achieves a rank1 accuracy of 82.3% on the Market1501[7] dataset.

Zhong et al.[30] introduce a fully automatic and unsupervised re-ranking method for person re-identification. For each image the k-reciprocal nearest neighbours are calculated and stored in a vector. The final distance between two images becomes a combination of the euclidean distance between the k-reciprocal vector and the standard feature vector extracted by the backbone network. Using a standard Resnet50[16], they achieve a rank1 accuracy of 77.11% on Market1501[7].

To tackle the problem of occlusion, Zhong et al.[31] introduce random erasing data augmentation where they randomly erase patches of the training images and replace these patches with random pixel values, this way they try to improve the robustness of their model by adding occlusion to the training images. They report 89.13% rank1 accuracy on the Market1501[7] dataset using the network created by Sun et al[29]. This heuristic has a positive impact on the performance, a lot of researchers used this random erasing data augmentation in fututre work.

Li et al[32] argue that only using local or only global features is disadvantageous for the systems accuracy, they decide to combine both. They use Resnet39[16] which is a a compareable network to Resnet50[16] but slightly less deep. The global features are extracted from the entire image input and for the local features the image is divided into horizontal

stripes and features are extracted form these stripes. They are then concatenated into a single feature vector. The CNNs which extract the features do not share weights which is pretty uncommon in the person re-identification field. They achieve 85.10% rank1 accuracy on the Market1501[7] dataset.

Sun et al.[33] reshape the standard backbone network by removing the global average pooling(GAP) and any following layers from it. The output from the backbone network is now a 3D tensor, they split this tensor into parts. For each of the parts they now extract feature vectors by performing the average pooling operation separated for each part of the image. These vectors are then passed through a series of fully connected layers and a softmax to create id predictions per part. During testing the features extracted of each part are concatenated into a single vector. They achieve a rank1 accuracy of 93.8% on Market1501[7].

Li et al.[34] argue that using deep architectures for person re-identification is not optimal, because in the input images the persons are not aligned while for most facial recognition data this is the case. They propose using an attention mechanism which is able to locate the important pixels in an image. The attention mechanism makes the network focus on these pixels which belong to the person while making sure background regions in the input image are treated as less important. Their network which they name an Harmonious attention CNN (HA-CNN) has almost 10 times less trainable parameters compared to the often used Resnet50[16] network but still achieving a 91.2% rank1 accuracy on the Market1501[7] dataset.

Zhou et al.[35] argue that the features extracted from the images should be omni-scale or multi-scale, to match people and distinguish them from impostors small local regions like shoes are just as important as global whole body regions. To extract these omni-scale features they introduce their own OSnet which is a lightweight feature extractor for multiple image scales. They achieve a rank1 accuracy of 94.8% on the Market1501[7] dataset.

Wang et al.[6]argue that using a GAP layer which treats activation's on the same feature map equally regardless of their location makes the model less robust to absence of certain features. To overcome this disadvantage they add their spatial attention layer. They use the same network set up as Sun et al.[33] and only add their SA layer before the GAP layer. They achieve a 94.7% rank1 accuracy on the Market1501[7] dataset.

4.1.2 Distance metric based approach

Ding et al.[36] are the first researchers in the person re-identification field to use a distance metric based approach. They us a 5-layer CNN to extract features from a batch of images. They train their network using the triplet loss function, it was first proposed by Wang et al.[37] who used it in an image classification task. The triplet loss function takes a set of triplets as input, a set of triplets consists of an anchor image, one positive match and a negative match. The features of these images are calculated by the CNN, the triplet loss is minimized if the features of the positive pair have a small euclidean distance while those of the negative pair should have a large euclidean distance. This method is also more computationally efficient only the features from a probe have to be calculated and compared to the features of the images in the gallery which have already been computed. They achieve a rank1 accuracy of 40.5% on the VIPeR[15] dataset.

Wang et al.[38] propose an SCNN which generates an embedding for the single image(SIR) and for the cross-image(CIR). The SIRs are trained using the triplet loss and the CIR are trained using softmax loss. They thus combine the verification loss with the distance metric based loss. Both embeddings are combined to produce a single similarity measure for the two input images. In their paper they showed the addition of this pair loss only resulted in an accuracy gain of 0.5% on the CUHK01[14] and 0.6% on the VIPeR[15] datasets resulting in a final rank1 accuracy of 71.8% on CUHK01[14] and 35.76% on VIPeR[15]. Their score on the VIPeR dataset is significantly lower than the score of Ding et al[36] while they only added complexity by adding more layers and features.

Hermans et al.[39] introduce two new variants of the the triplet loss function used by Ding et al.[36], the batch hard and batch all variants. In the standard implementation, once a certain set of B triplets has been chosen, their images are stacked into a batch of size 3B, for which the 3B embeddings are computed, which are in turn used to create B terms contributing to the loss. In their implementation they form batches by randomly sampling P classes, and then sampling K images of each class. For each sample in the batch they select the hardest positive and hardest negative within the batch for computing the loss. They report a rank1 score of 84.90% on Market1501[7]. In their paper they also show that adding the re-ranking method[31] their rank1 accuracy rises to 86.67% on the Market1501[7] dataset.

Zhang et al.[40] combine two streams in their network, one which uses the image as an input an collects the global and local features another which from the input image first extracts the dense semantic aligned parts(DSAP) from the input images, these are passed through a similar CNN architecture. All features are extracted and combined to two feature vectors. They use the batch hard triplet loss from Hermans et al.[39] and the random data augmentation from Zhong et al[31]. They achieve a rank1 accuracy of 95.7% on the Market1501[7] dataset.

Luo et al[41] collected different heuristics used in the perosn re-identification field and evaluated the impact of different heuristics. They use a standard Resnet-50[16] pre-trained on Imagenet[20]. The first heuristic they apply is warming up the learning rate, which was first introduced by Fan et al.[42]. Next they added the random data augmentation which was first introduced by Zhong et al.[31]. Next they added label smoothing, which is common practice in a lot of classification tasks, first introduced by Szegedy et al.[43]. Another heuristic to improve accuracy is adjusting the size of the last stride, resulting in larger feature vectors. They also state that using a batch normalization layer after the feature layer and before the final fully connected layer. They also suggest using Center loss, which is used in facial recognition[44]. Center loss minimizes the intra-class compactness, they suggest combining the center loss with the identification loss and the triplet loss.

Wang et al.[45] show that adding attentive layers to the standard Resnet50[16] backbone can be beneficial to the networks performance. They Combine the identification and classification loss and achieve a 93.1% rank1 accuracy on the Market1501[7] dataset.

Chen et al.[46] build upon the idea of attentive layers used by Wang et al[45], they add regularization and split the network in to two branches; a global branch and an attentive branch. The attentive branch consists of two modules named Channel attentive module(CAM) and Part attentive module(PAM). The network combines the branches into a 2048-dimensional feature vector. They achieve a rank1 accuracy of 95.6% on the Market1501[7] dataset.

Lawen et al.[47] build upon the idea of using attention based CNN's from[34] and add the training heuristics used by Luo et al[41]. Only adding these heuristics to the baseline HA-CNN already increases the rank1 accuracy on the Market1501[7] dataset to 93.2%. Next they also make some adjustments to the baseline HA-CNN making it deeper and wider, these adjustments increase the rank1 accuracy on the Market1501[7] dataset to 96.2%. They show that using simple L2 normalization instead of the batch normalization used by Luo et al.[41]. They also show that the less complex version of their network with the same amount of parameters as the HA-CNN, is able to outperform the standard HA-CNN version by achieving a 95.8% accuracy on the Market1501[7] dataset.

Quan et al.[48] argue that the backbone network architectures which are often used for person re-identification; VGG[28], Inception[49] and Resnet[16] are not specialized for this task as they are are trained for image classification and not re-identification. They suggest using a neural architecture search network(NAS) which searches for an optimal architecture for solving the task. They find an architecture with 50% less parameters than the standard Resnet architecture and achieve a rank1 accuracy of 95.4% on the Market1501[7] set using re-ranking[30].

4.1.3 Open-set focus

Li et al.[50] introduce an open-set person re-identification model, their idea is to use a generative adversarial network(GAN)[51]to generate images which look alot like the target image and add these to the training set. This way the feature extractor will be able to learn to separate the generated image from the true target even though they look very similar. They define the open-set person re-identification problem as a target search; The model has a gallery of targets or people we want to find, images are then fed to the network and it reports if the input person is one of the targets. They report a TTR of 22.32% for a FTR of 1% on the Market1501[7], only slightly outperforming a simple Resnet50[16] baseline which gets a TTR of 20.79% for a FTR of 1% on the same dataset.

Yu et al.[52]propose that instead of representing each person image as a feature vector, it should be modeled as a Gaussian distribution with its variance representing the uncertainty of the extracted features. They use a standard Resnet50[16] as their backbone network. Representing the feature vector as a Gaussian distribution will make the network more robust against noisy training samples which is especially useful in the open world scenario. They achieve a rank1 accuracy of 87.3% on the Market1501[7] dataset and also propose an open-world scenario using the Market1501[7] dataset, in the open-world scenario they achieve a TTR of 87.88% for FTR of 1%.

4.1.4 Adding external information

Lin et al[53] manually added 27 attribute labels to several benchmark datasets, the network learns to extract these attributes for new input images. These attribute labels describe different attributes of the probe person, for example the type of clothing and the color of the clothing. They concatenate the 27 dimensional attribute vector with a 512 dimensional feature vector extracted by a standard Resnet50[16] backbone. The define an objective function which combines the identification features and the attribute labels and train their network. They achieved a rank1 accuracy of 87.04% on the Market1501[7] dataset.

Some researchers decided that instead of trying to improve the feature extraction, they would focus on adding spatial-temporal information. Adding spatial-temporal constraint can improve the accuracy by eliminating irrelevant gallery images. Cho et al.[54] first used this by creating a strong assumption, given a person image at timestamp t, this person

should appear at the next camera within t plus or minus some delta, they unfortunately do not run tests on any benchmark datasets. Wang et al.[6] further explore the idea of using spatial temporal information by running a parzen-window of the temporal statistics from the dataset and combining the spatial-temporal data and CNN features in to a joint metric. They report rank1 accuracy of 98.0% on Market1501[7].

4.1.5 Deep facial recognition

The facial recognition task is technically the same as the re-identification task; The probe face must be matched to one of the faces in the gallery. Two big differences can be seen between facial recognition and person re-identification. Firstly in the facial recognition field the datasets are larger, for example the Megaface[55] dataset which contains over a million pictures of more than 600.000 identities thus being more representative of real world scenarios. Secondly the differences per image in the facial recognition field are more subtle[9]. In the state of the art literature for facial recognition researchers take similar approaches as the researchers in the person re-identification field.

Schroff et al.[56] Introduced the triplet loss for facial recognition which was later adapted and used in the person re-identification research.

Liu et al. [57] introduce Angular Softmax where the softmax decision boundary only depends on an angle. In the re-identification field the main loss functions used were the triplet loss and the center loss, however in the facial recognition field some of the top performing networks use an angular margin based loss. Triplet loss requires sample mining while angular margin based losses do not. Three state of the art performing papers on the Megaface[55] dataset are; SphereFace[57],CosFace[58] and ArcFace[59]. Liu et al.[57] map the images on the surface of a hypsphere which limits the possible space distribution to a restricted angular space. To overcome the difficulty of optimizing the sphereface loss, which incorporates the angular margin in a multiplicative manner ArcFace[59] and CosFace[58] respectively introduced and additive angular/cosine margin which are able to converge without softmax supervision. Resnet[16] is used as backbone network by the authors of these papers. The results on the MegaFace[55] dataset are summed up in the table below.

	Rank1 accuracy(%)	Verification accuracy(%)
SphereFace[57]	75.61	89.14
CosFace[58]	77.11	89.88
ArcFace[59]	77.50	86.47

Table 2: Results on the MegaFace dataset

Fan et al[42] used the idea of embedding the images on a hypersphere plane for the person re-identification task, using the angular softmax function which was introduced by Liu et al[57] for the facial recognition task. They used a basic Resnet50[16] as their feature extractor and added some data augmentation and normalization, with these relatively simple adjustments they achieve a rank1 accuracy of 93.1% on the Market1501[7] dataset.

Authors	Training Loss	Backbone Architecture	Training heuristics	**Market1501**[7]	**CUKH01**[14]	**VIPeR**[15]	**DUKE**[13]	Year
Yi et al.[21]	Cosine	5-layer CNN	-	-	-	34.4%	-	2014
Ahmed et al.[22]	Softmax	4-layer CNN	DA	-	47.5%	34.8%	-	2015
Ding et al.[36]	Triplet	5-layer CNN	N, DA	-	-	40.5%	-	2015
Wu et al.[23]	Softmax	10-layer CNN	DA	37.2%	71.1%	-	-	2016
Wang et al.[38]	Triplet, SVM	8-layer CNN	DA	-	71.8%	35.8%	-	2016
Wu et al.[17]	LDA	3-FC layers	DA	48.2%	67.1%	44.1%	-	2017
Varior et al.[25]	Cosine	10-layer CNN	DA	65.9%	-	37.8%	-	2016
Zhao et al.[26]	Softmax	8-layer CNN	DA	76.9%	79.9%	53.8%	-	2017
Zheng et al.[27]	Softmax	Resnet50	DA	79.5%	-	-	-	2018
Sun et al.[29]	Softmax	Resnet50,Caffenet	DA	82.3%	-	-	76.2%	2018
Zhong et al.[30]	L2 distance	Resnet50	RR	77.1%	-	-	-	2017
Zhong et al.[31]	Softmax	Resnet50	RR,REDA	89.1%	-	-	84.0%	2018
Li et al[32]	Softmax	Resnet39	DA	85.1%	91.2%	50.2%	-	2017
Hermans et al.[39]	Triplet	Resnet50	DA,RR	86.7%	-	-	-	2017
Lin et al.[53]	Softmax	Resnet50	DA	87.0%	-	-	-	2018
Sun et al.[33]	Softmax	Resnet50	US	93.8%	-	-	83.3%	2018
Zhang et al.[40]	Softmax, Triplet	Resnet50	DA, REDA, BHT	95.7%	90.1%	-	86.2%	2019
Luo et al.[41]	Triplet, Center	Resnet50	WL, REDA, RR, BHT, LS, US	95.4%	-	-	90.4%	2019
Li et al.[34]	Softmax	HA-CNN	DA	91.2%	-	-	80.5%	2018
Wang et al.[45]	Triplet, Softmax	Resnet50+Attention	DA,REDA	93.1%	-	-	84.9%	2018
Chen et al.[46]	Softmax, Triplet	Resnet50+Attention	REDA,WL,N	95.6%	-	-	89.0%	2019
Lawen et al.[47]	Triplet	HA-CNN(mod)	REDA, LS, BHT,WL,N	96.2%	-	-	89.8%	2019
Zhou et al.[35]	Softmax	OS-Net	REDA,LS,WL	94.8%	-	-	88.6%	2019
Wang et al.[60]	Softmax	Resnet50	REDA,RR,WL	94.7%	-	-	89.0%	2019
Quan et al.[48]	Softmax, Triplet	Auto Re-id	RR	95.4%	-	-	-	2019
Wang et al.[6]	Softmax	Resnet50	US,ST	98.0*%	-	-	94.4%	2019
Fan et al.[42]	A-softmax	Resnet50	DA,WL,BN,N	94.4%	-	-	83.9%	2019
Yu et al.[52]	DistributionNet	Resnet50	DA	87.3%	94.2%	74.7%	-	2019

Table 3: Overview of Approaches used in person re-identification.(DA= Data augmentation[61])(SVD= Singular Value Decompisition[51])(REDA= Random erasing data augmentation[31])(RR= Re-ranking[30])(LS = Label smoothin[43])(N=L2 normalization[51])(BN= Batch normalization[51])(BHT= Batch hard triplet loss[39])(WL = warmup learning rate[41])(US = Upsampling[33])(ST = Spatial temporal information[6])(*Added spatial temporal information)

5 Discussion

The methods described in the previous chapter have been summed up in table 3. Each approach to the person re-identification problem can be separated into four design choices; Loss function, Backbone architecture, training heuristics and which image features are extracted. In this chapter we will discuss the different approaches for each of these design choices.

5.1 Backbone architecture

Before the network can be trained first features must be extracted from an image, these features are extracted by a backbone network. From the papers discussed in this review, it can be concluded that the deeper backbone architectures are more successful at solving the person re-identification task. The best performing non deep system was created by Wu et al.[62] and is still inferior to any state of the art deep methods. Many authors[21, 22, 36, 23, 38, 25, 26] decided to build their own convolutional neural networks and train them from scratch, while other authors[9, 29, 31, 30, 32, 39, 53, 33, 40, 41, 46, 6, 60, 42] used deeper networks which already proved to be successful for image recognition tasks like Resnet50[16], VGG16[24] and Caffenet[28]. From table 3 we can conclude that deeper backbone architectures outperform the more shallow ones used in earlier works. Most researchers use Resnet50[16] or one of its adjusted versions as their backbone network. Authors that used these deep networks had to use the pre-trained versions in their models because of the limited amount of training data available in the person re-identification task. Some more recent works have argued that using a backbone network which is pre-trained on Imagenet[20] is disadvantageous as it is not specialized in extracting features that distinguish persons but it is specialized in extracting features which distinguish objects. However until the discovery of attention based networks, which were used by[53, 46, 47, 35, 48] there simply wasn't a network which was able to achieve the same performance. Another possibility is to use a neural architecture search to search for an optimal network for the task, Quan et al.[48] deployed this idea. Using Resnet50 as backbone network is not coincidence, Sun et al.[29], Li et al.[32] and Zheng et al[27] tried different backbone networks like VGG16[24], Caffenet[28] and Resnet50[16] which were all top performing networks on the Imagenet[20] challenge.

5.2 Image features used

As discussed in 4.1 features are extracted by the backbone network, but which features are interesting? Two main types of features can be defined; Global and local features.

Global features are learned from the entire image and intend to capture the most discriminative features of appearance but may fail to capture discriminative local features. Therefore local features may be used so that local discriminative feature can also be captured. Combining local and global features is a popular approach[38, 26, 32, 40, 33, 35] some authors randomly divided the image into parts[38, 32, 33] and extracted local features from these image parts and combine them with the features from the entire image. Others[26, 40] extract information about the pose and used this information to find certain body parts subsequently extracting features from these body parts. Other authors added external information to the images like attribute labels[53] or spatial temporal information[6]. Most researchers only use the global features/feed the entire image into their model.

5.3 Loss function

Machines learn by means of a loss function. It's a method of evaluating how well specific algorithm models the given data. If predictions deviates too much from actual results, the loss function would become a very large number(Goodfellow[51],2016). Generally the objective is to minimize the loss function. In the field of person re-identification we can split the loss functions used into three main categories; distance metric based, verification based loss and identification based loss. For distance metric based loss functions the euclidean distance between two feature vectors is computed and these are compared. Minimizing the euclidean distance between two feature vectors extracted from images which belong to the same person is called using the euclidean loss or L2 loss. If we add a term to the loss function which also pushes feature vectors from different images away we get the triplet loss. The triplet loss not only maximizes inter class compactness, it also maximizes intra class distance.

Verification based loss treats the person re-identification problem as an binary classification problem, the network takes a pair of images as input and outputs a binary classification score 1 or 0. Identification based loss treats the person re-identification problem as a retrieval problem, each person is treated as a separate class, so instead of outputting if two input images are the same person or not, the output is a classification. Identification based systems are often optimized using the softmax or cross-entropy loss. Often identification and verification losses are combined to create a more robust system.

During testing and validation the true labels are not part of the final classification layer as these are new people, this means the performance needs to be measured by looking at the euclidean distance between the extracted feature vectors. The smaller this distance is, the more similar the images should be thus the greater the chance of a correct match. Often the gallery is ranked based on euclidean distance to the probe image. In an open world setting a threshold is set, when this threshold is exceeded by the closest match from the gallery then no match is given. The idea to embed image features on a hypersphere plane and then separate them by an angular margin, which was mainly used in the facial recognition field[57, 58, 59] can also be used for the person re-identification field, this was shown by Fan et al.[42].

5.4 Training heuristics

Like with any other deep learning problem the representation of the data is of extreme importance[51]. Data can be changed to best match your systems requirements, this is called data augmentation. Many different forms of data augmentation exist; Padding where zero values are added to image. Random cropping where a random crop is taken from an input image. Mirroring/flipping training data to create more data and to make the system more robust. Often the images are normalized so each pixel represents a value between 0 and 1 instead of 0 to 255. Some authors calculate the mean image and then subtract it from the input image. To make the model more robust against occlusion many author[31, 40, 41, 46, 47, 60] use random erasing data augmentation, here a random box of the image is selected and its pixel values are randomised. Label smoothing is another heuristic which is applied quite often by authors[41, 47, 35] here they adjust the labels by adding a threshold, by doing this the model is less confident on the training set. If the training set is small this could prevent overfitting. Re-ranking is also often applied for person re-identification, it was first introduced by Zhong et al.[30] and later added to their models by the following authors[31, 39, 41, 60]. For each image the k-reciprocal nearest neighbours are calculated and stored in a vector. The final distance between two images becomes a combination of this k-reciprocal distance and the euclidean distance. Warming up the learning rate is something that is popular in the re-identification field as well. It is used by the authors[41, 46, 47, 35, 60, 42] which are mainly the authors of the more recent papers(after 2018). Warming up the learning rate is a pretty simple heuristic where instead of starting with a high learning rate and slowly decaying it over time, the learning rate starts relatively low and then 'warms up' and then decays over time. Normalizing the feature is also a popular strategy for a lot of deep learning approaches[51] this also applies to the person re-identification problem. If the features are normalized each dimension of the feature vector is balanced. The features now have a Gaussian distribution near the surface of a hypersphere, thus keeping a compact distribution of features that belong to the same class. Mainly the authors who focused on distance metric based person re-identification used normalization in their setup[36, 41, 33, 42, 47]. Extracting more information from an image could also prove beneficial, by changing the lasts stride in the final filtering layer of Resnet50[16] some authors[33, 41, 60] extract a larger feature vector from the image and show that this increase the accuracy.

Training heuristics are an essential part of every person re-identification model however it is hard to estimate the effect of each heursitic. Luo et al.[41] try to show the influence of different heuristics on the accuracy achieved in their research. They first showed their standard accuracy and then subsequently add training heuristics and show the resulting accuracy but it still remains challenging to estimate the effect of a single heuristic, the final accuracy increased more than 7% with all heuristics applied. Lawen et al.[47] added training heuristics to the network of Li et al.[34] and the heuristics resulted in more than 5% accuracy gain on the Market1501[7] dataset.

5.5 Open-set

Much effort has been expended on developing methods for person re-identification. However existing research aims at maximising ranking performance on closed-set benchmark datasets these datasets are unrepresentative of scale and complexity of more realistic open-world scenarios. The most popular benchmark dataset in the research community is the Market1501[7] dataset. It has 750 identities in the training set and each identity has between 2 and 20 images. As argued in the open-set re-identification survey[3] the difference between the real world scenarios and the scenarios created in the datasets is still very big, so a lot of the work done can not be applied to real world use cases. The first steps to creating more releastic open-world scenarios are to add distractors to the probes and to add distractors to the gallery images. In this review two approaches to the open-set problem were discussed, in these approaches the authors modified the standard Market1501[7] so that the probe persons were not necessarily in the gallery thus giving the network the option to output that the probe does not exist in the gallery. Li et al.[50] tried to generate images which had a similar appearance as the targets to try and make the model more robust against similar looking people, however they only slightly outperformed a standard Resnet50[16]. Yu et al.[52] showed that modeling the features as Gaussian distributions with its variance representing the uncertainty of the extracted features can be very successful for open-set person re-identification.

5.6 Dataset

From table 3 we can conclude that the performance over the different datasets varies greatly. Some authors decide only to mention certain datasets in their research papers which may be beneficial to the accuracy of their system. For example Zhong et al.[31] achieve a rank1 accuracy of 84% on the DUKE datasets and 89.1% on the Market dataset, while Fan et al.[42] achieve a lower accuracy on the DUKE dataset 83.9% they achieve almost 5% higher accuracy on the Market150 datasets, similar examples are given in table3. So even though a system may work very wel on a particular dataset it might be inferior on other datasets. The accuracy achieved also differs greatly per dataset, one being more challenging[15] than the others[7, 14, 13]. The VIPeR[15] dataset can be considered more challenging because it only contains two images per person.

6 Future Directions

Backbone architecture The choice of backbone architecture is of extreme importance for the person re-identification task. The most used backbone architecture is Resnet50[16] which at the time of its introduction was the top performing model for the imagenet challenge. Recently new networks have been discovered which outperform the aforementioned architectures and some even have fewer parameters. The best performing architectures have been summed up in table 6. The InceptionV3[43] and Xception[63] were already applied to the person re-identification challenge by Rooijen et al.[64] and were beneficial to the accuracy. Using Efficientnet[65] might also be beneficial to the achieved accuracy.

Model	**Rank1(%)**	**#parameters**
Caffenet [28]	68.9	11m
VGG16 [24]	70.5	138m
Resnet50 [16]	75.9	26m
InceptionV3 [43]	78.8	24m
Xception [63]	79.0	23m
EfficientnetB1 [65]	79.2	7.8m
EfficientnetB2 [65]	80.3	9.3
EfficientnetB3 [65]	81.7	12m
GPIPE [66]	84.3	557m
EfficientnetB7 [65]	84.4	66m
Resnext101 [67]	85.4	829m

Table 4: rank1 accuracy on the Imagenet[20] challenge

Loss function In recent literature we have seen that the triplet loss and the softmax loss are most often applied. Very few research in the person re-identification field deviates from these loss functions, while in the facial recognition field a lot of research is aimed at finding a loss function which increases performance. From the facial recognition survey[68] we see that state of the art performing models have created their own loss functions, like the A-softmax[57],CosFace[58] and ArcFace[59]. So far only Fan et al.[42] adapted one of these functions, the A-softmax for the person re-identification field. Using one of the other above mentioned loss functions might also prove beneficial to the achieved accuracy.

Training heuristics Using can significantly improve performance, different heuristics are beneficial for different datasets. Some researchers however tend to overlook the heuristics and just focus on improving the loss function or backbone network. There are a few examples in literature where authors took an improved architecture/loss function and simply added some training heuristics resulting in a gain in final accuracy. Applying these heuristics on the work of Wang et al.[6] or Quan et al.[48] could potentially improve the state of the art accuracy. Also researching which type of heuristics are useful for which type of data can be very useful for the person re-identification research field.

7 Conclusion

In this review, an overview of the current top performing re-identification models is given. First, the person re-identification task and the connected issues were discussed. Second, a brief overview of the current state of the art performing models is given, these were split into ID/verification based approaches and distance metric based approaches. Some models from the facial recognition field were also included in this survey as the task of facial recognition is very similar to person re-identification. Each approach can be summarized in four design choices; Loss function, Backbone architecture, training heuristics and the features that were used. Finally, we suggest some future directions which might be beneficial to the achieved accuracy.

References

[1] Technolution. Flowcube description: https://www.technolution.eu/uploads/2019/07/flowcube-description.pdf, 2019.

[2] Michael Chertok, Alexander Voukelatos, Vicky Sheppeard, and Chris Rissel. Comparison of air pollution exposure for five commuting modes in sydney-car, train, bus, bicycle and walking. *Health promotion journal of Australia*, 15(1):63–67, 2004.

[3] Qingming Leng, Mang Ye, and Qi Tian. A survey of open-world person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[4] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a discriminative null space for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1239–1248, 2016.

[5] Yue-Fei Guo, Lide Wu, Hong Lu, Zhe Feng, and Xiangyang Xue. Null foley–sammon transform. *Pattern recognition*, 39(11):2248–2251, 2006.

[6] Guangcong Wang, Jianhuang Lai, Peigen Huang, and Xiaohua Xie. Spatial-temporal person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8933–8940, 2019.

[7] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.

[8] Di Wu, Si-Jia Zheng, Xiao-Ping Zhang, Chang-An Yuan, Fei Cheng, Yang Zhao, Yong-Jun Lin, Zhong-Qiu Zhao, Yong-Li Jiang, and De-Shuang Huang. Deep learning-based methods for person re-identification: A comprehensive review. *Neurocomputing*, 2019.

[9] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016.

[10] Apurva Bedagkar-Gala and Shishir K Shah. A survey of approaches and trends in person re-identification. *Image and Vision Computing*, 32(4):270–286, 2014.

[11] Riccardo Satta. Appearance descriptors for person re-identification: a comprehensive review. *arXiv preprint arXiv:1307.5748*, 2013.

[12] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[13] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016.

[14] Wei Li, Rui Zhao, and Xiaogang Wang. Human reidentification with transferred metric learning. In *ACCV*, 2012.

[15] Douglas Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, volume 3, pages 1–7. Citeseer, 2007.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Lin Wu, Chunhua Shen, and Anton Van Den Hengel. Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification. *Pattern Recognition*, 65:238–250, 2017.

[18] Wanli Ouyang and Xiaogang Wang. Joint deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2056–2063, 2013.

[19] Xingyu Zeng, Wanli Ouyang, and Xiaogang Wang. Multi-stage contextual deep learning for pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 121–128, 2013.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[21] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Deep metric learning for person re-identification. In *2014 22nd International Conference on Pattern Recognition*, pages 34–39. IEEE, 2014.

[22] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3908–3916, 2015.

[23] Lin Wu, Chunhua Shen, and Anton van den Hengel. Personnet: Person re-identification with deep convolutional neural networks. *arXiv preprint arXiv:1601.07255*, 2016.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[25] Rahul Rama Varior, Mrinal Haloi, and Gang Wang. Gated siamese convolutional neural network architecture for human re-identification. In *European conference on computer vision*, pages 791–808. Springer, 2016.

[26] Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1077–1085, 2017.

[27] Zhedong Zheng, Liang Zheng, and Yi Yang. A discriminatively learned cnn embedding for person reidentification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 14(1):13, 2018.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3800–3808, 2017.

[30] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1318–1327, 2017.

[31] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.

[32] Wei Li, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep joint learning of multi-loss classification. *arXiv preprint arXiv:1705.04724*, 2017.

[33] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin 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)*, pages 480–496, 2018.

[34] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2285–2294, 2018.

[35] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. *arXiv preprint arXiv:1905.00953*, 2019.

[36] Shengyong Ding, Liang Lin, Guangrun Wang, and Hongyang Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 48(10):2993–3003, 2015.

[37] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[38] Faqiang Wang, Wangmeng Zuo, Liang Lin, David Zhang, and Lei Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1288–1296, 2016.

[39] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[40] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Densely semantically aligned person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 667–676, 2019.

[41] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[42] Xing Fan, Wei Jiang, Hao Luo, and Mengjuan Fei. Spherereid: Deep hypersphere manifold embedding for person re-identification. *Journal of Visual Communication and Image Representation*, 60:51–58, 2019.

[43] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[44] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[45] Cheng Wang, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Mancs: A multi-task attentional network with curriculum sampling for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 365–381, 2018.

[46] Tianlong Chen, Shaojin Ding, Jingyi Xie, Ye Yuan, Wuyang Chen, Yang Yang, Zhou Ren, and Zhangyang Wang. Abd-net: Attentive but diverse person re-identification. *arXiv preprint arXiv:1908.01114*, 2019.

[47] Hussam Lawen, Avi Ben-Cohen, Matan Protter, Itamar Friedman, and Lihi Zelnik-Manor. Attention network robustification for person reid. *arXiv preprint arXiv:1910.07038*, 2019.

[48] Ruijie Quan, Xuanyi Dong, Yu Wu, Linchao Zhu, and Yi Yang. Auto-reid: Searching for a part-aware convnet for person re-identification. *arXiv preprint arXiv:1903.09776*, 2019.

[49] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[50] Xiang Li, Ancong Wu, and Wei-Shi Zheng. Adversarial open-world person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 280–296, 2018.

[51] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[52] Tianyuan Yu, Da Li, Yongxin Yang, Timothy M Hospedales, and Tao Xiang. Robust person re-identification by modelling feature uncertainty. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 552–561, 2019.

[53] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 2019.

[54] Yeong-Jun Cho, Su-A Kim, Jae-Han Park, Kyuewang Lee, and Kuk-Jin Yoon. Joint person re-identification and camera network topology inference in multiple cameras. *Computer Vision and Image Understanding*, 180:34–46, 2019.

[55] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4873–4882, 2016.

[56] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[57] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[58] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[59] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[60] Haoran Wang, Yue Fan, Zexin Wang, Licheng Jiao, and Bernt Schiele. Parameter-free spatial attention network for person re-identification. *arXiv preprint arXiv:1811.12150*, 2018.

[61] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[62] Shangxuan Wu, Ying-Cong Chen, Xiang Li, An-Cong Wu, Jin-Jie You, and Wei-Shi Zheng. An enhanced deep feature representation for person re-identification. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–8. IEEE, 2016.

[63] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[64] A.L. van Rooijen. Deep learning for person re-identification. In *Masters thesis*, 2018.

[65] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[66] Yanping Huang, Yonglong Cheng, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, and Zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *arXiv preprint arXiv:1811.06965*, 2018.

[67] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.

[68] Mei Wang and Weihong Deng. Deep face recognition: A survey. *arXiv preprint arXiv:1804.06655*, 2018.

Bibliography

[1] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[2] CBS. Factsheet cycling netherlands. *https://www.cbs.nl/-/media/imported/documents/2015/27/2015-factsheet-nederland-fietsland.pdf?la=nl-nl*, 2015.

[3] cs231n. Convolutional neural networks for visual recognition. *https://cs231n.github.io/*.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

[6] Timothy Dozat. Incorporating nesterov momentum into adam. 2016.

[7] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[8] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[9] Douglas Gray, Shane Brennan, and Hai Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, volume 3, pages 1–7. Citeseer, 2007.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1314–1324, 2019.

[12] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

[16] Wei Li, Rui Zhao, and Xiaogang Wang. Human reidentification with transferred metric learning. In *ACCV*, 2012.

[17] Wei Li, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep joint learning of multi-loss classification. *arXiv preprint arXiv:1705.04724*, 2017.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[19] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 2019.

[20] Murgia Madhumita. Who's using your face? the ugly truth about facial recognition. *Financial times*, 2019.

[21] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12 (1):145–151, 1999.

[22] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016.

[23] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[24] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[25] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[26] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[27] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin 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)*, pages 480–496, 2018.

[28] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[29] Technolution. Flowcube description: https://www.technolution.eu/uploads/2019/07/flowcube-description.pdf, 2019. URL `https://www.technolution.eu/uploads/2019/07/flowcube-description.pdf`.

[30] Tijmen Tieleman and Geoffery Hinton. Rmsprop gradient optimization. *URL http://www. cs. toronto. edu/tijmen/csc321/slides/lecture_slides_lec6. pdf*, 2014.

[31] Faqiang Wang, Wangmeng Zuo, Liang Lin, David Zhang, and Lei Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1288–1296, 2016.

[32] Guangcong Wang, Jianhuang Lai, Peigen Huang, and Xiaohua Xie. Spatial-temporal person re-identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8933–8940, 2019.

[33] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[34] Fei Xiong, Mengran Gou, Octavia Camps, and Mario Sznaier. Person re-identification using kernel-based metric learning methods. In *European conference on computer vision*, pages 1–16. Springer, 2014.

[35] Sixin Zhang, Anna E Choromanska, and Yann LeCun. Deep learning with elastic averaging sgd. In *Advances in neural information processing systems*, pages 685–693, 2015.

[36] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Densely semantically aligned person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 667–676, 2019.

[37] Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1077–1085, 2017.

[38] Feng Zheng, Cheng Deng, Xing Sun, Xinyang Jiang, Xiaowei Guo, Zongqiao Yu, Feiyue Huang, and Rongrong Ji. Pyramidal person re-identification via multi-loss dynamic training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8514–8522, 2019.

[39] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015.

[40] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. *arXiv preprint arXiv:1905.00953*, 2019.