Universiteit Leiden

Computer Science

Deep learning for person re-identification

Name: A.L. van Rooijen

Date: December 11, 2017

1st supervisor: Prof. Dr. Ir. F.J. Verbeek (Leiden University)
2nd supervisor: Dr. Ir. H. Bouma (TNO)

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Master Thesis

Deep learning for person re-identification

A.L. van Rooijen, Bsc.
Supervised by:
Prof. Dr. Ir. F.J. Verbeek (Leiden University) & Dr. Ir. H. Bouma (TNO)

December 11, 2017

Abstract

Person re-identification is the task of ranking a gallery of automatically detected images of persons using a set of query images. This is challenging due to the different poses, viewpoints, occlusions, camera configurations, image distortions, lighting conditions, image resolutions and imperfect detections, which all affects a person re-identification system's performance.

Recently deeply learned systems have become prevalent in the person re-identification field as they are capable to deal with the various obstacles encountered. One such a system is ConvNet using a coarse-to-fine search framework (ConvNet+C2F), which is developed with both a high retrieval accuracy as a fast query time in mind.

In this thesis we propose several adaptations to ConvNet+C2F to improve its performance. We use the novel convolutional model Xception to construct a new ConvNet called XConvNet, train it using the modern Adadelta model optimizer and demonstrate that a smaller coarse descriptor improves retrieval time and accuracy for C2F. With the proposed improvements XConvNet+C2F achieves state-of-the-art results on two different well known datasets for person re-identification, i.e. Market-1501 and CUHK03.

Furthermore, we investigated the possibilities of an artificial extension of the training set using generated images and study the effect of different batch sizes used in the classifier training.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Person re-identification (person re-id) is the task of finding the same person in multiple images or video resources, such as surveillance videos. This is relevant, because a manual search for individuals in these resources is a laborious task and is infeasible for larger real world camera networks; such networks can contain up to hundreds of different cameras.

Main difficulties for automating the person re-id task are the variations in colour, lighting, view angle, background, poses of individuals and used camera resolutions between the videos of the cameras in such a network. Deep learning is extensively used to overcome these obstacles and is becoming quite effective at it.

In recent years the field of person re-identification has seen a steady improvement in the accuracy on various challenging datasets. However, in support of these developments, most research focuses solely on accuracy without taking retrieval time into account. As a consequence, this research is not necessarily usable in practice. Therefore the central research research question of this thesis is:

can we obtain state-of-the-art accuracy and high speed performance?

It is required that the system works fast in a realistic setting, meaning that a response time less than a second when faced with hundreds of thousands images is desirable. The goal of our research focuses on combining the state-of-the-art person re-identification which demonstrate fast retrieval with the latest developments in the field of deep learning. Furthermore, we will look into the feasibility of generating additional training data in order to reduce the need for the time intensive construction of a large dataset.

This thesis is organized as follows. First recent and relevant developments in the field of person re-identification and deep learning are discussed in Section 2. In Section 3 we present the design and implementation. This is followed by Section 4 in which the state-of-the-art performance of our proposed method is demonstrated. Finally in Section 5, we interpret our results and discuss future work.

Chapter 2

Related work

In this chapter, we start with providing an overview of the current state-of-the-art for person re-identification in Section 2.1. Datasets used for person re-identification are presented in Section 2.2. In Section 2.3 several deep learning models suitable for person re-identification are discussed, which is followed by Section 2.4 detailing optimizers which can be used to train these networks. In Section 2.5 we point out several methods which can be used to improve the average query time of person re-identification methods.

2.1 Deep learning for person re-identification

In this section we provide an overview of recent developments in the person re-id field. Hereby we put an emphasis on deeply learned systems, as they are currently the most promising in solving person re-id. Papers included are mainly selected because they apply deep learning for person re-identification and report state-of-the-art results on public person re-identification datasets. However, some papers [72, 49, 78, 44] using hand-crafted features are mentioned, either for comparison or because of their large number of citations. Furthermore, we mention the Rank-1 between brackets to give an idea of their performance on various data sets. This Rank-1 reports the probability of a correct matching image to be in the first position of the ranking. When two numbers are listed, the first number corresponds to the accuracy obtained using Single Query and the second number corresponds to the accuracy obtained using multi-query.

Li et al. [38] add a filter pairing layer to a Siamese Convolutional Neural Network (SCNN), thereby creating a Filter Pairing Neural Network (FPNN). Due to this layer, the network is able to overcome geometric and photometric transformations that exists between its two inputs. This layer identifies and boosts similar features between the two input images, which are used by the final layer to do the verification. They demonstrate state-of-the-art results on the first Chinese University of Hong Kong dataset (CUHK01) (27.87%) and CUHK03 (20.65%). The CUHK03 data set is introduced in this paper and it is the first person re-id data set that also contains automatically constructed bounding boxes.

Yi et al. [80] divide a pair of input images into three overlapping segments, for the head, torso and legs respectively, and train a separate SCNN on each part. Where Li et al. [38] treat person re-identification as a verification problem, Yi et al. [80] view it as a ranking problem. To produce the needed similarity rankings, each SCNN computes the cosine difference between the embeddings of its inputs. The summation of these three separate similarity values indicates the final image similarity. Using a view specific SCNN they achieve state-of-the-art results on VIPeR (17.72%). However, with a generic SCNN that is first trained (i.e. pre-trained) on CUHK01 and subsequently trained on (i.e. fine-tuned) on VIPeR, they achieve an accuracy (34.4%) that significantly improves on the view specific SCNN.

Ding et al. [14] also approach person re-id as a ranking problem, but use a more computationally efficient method than Yi et al. [80]. By using a CNN with a fully connected (fc) layer as a last layer, they compute embeddings for the gallery images. Now, during test time only a computationally expensive embedding needs to be computed for the query image. The L2 (Euclidean) distance between this embedding and those of the gallery images defines a similarity ranking, where a smaller distance corresponds with a higher similarity. Furthermore, because they train their network using the triplet loss, more constraints are imposed during training, which subsequently aids in the models generalization ability. Therefore the network performs better on smaller data sets, when compared to those of Li et al. [38] and Yi et al. [80]. For the training of the model they devise an improved backpropagation algorithm and an optimized triplet generation scheme. Their network achieves state-of-the-art results on iLIDS (52.1%) and VIPeR (40.5%).

Ahmed et al. [2] improve the SCNN model by adding a cross-input layer. Instead of looking at the similarities between its inputs, as Yi et al. [80] did, this layer looks at the differences in neighbouring regions. Using this information the network is better able to verify if its two inputs belong to the same class (identity) or not. They achieve state-of-the-art results on CUHK01 (65.00%) and CUHK03 (54.74%) and VIPeR (34.81%). Furthermore, like Yi et al. [80], they show that pre-training on a different, larger data set, improves performance on another, smaller one.

Zhang et al. [83] introduce bit-scalable hashing. A ten layer CNN is trained to both extract deep features from raw input images and to transform these features into a hash code for the image. They demonstrate that this combined learning of feature extraction and hash function learning yields better performance than separating both tasks. The bits forming the hash code are weighted by the network to signify their importance. During testing time they can cut off the insignificant bits and thereby improve the time efficiency of their network. In order for their network to train properly, they extend the triplet loss to work with hamming distances on the hash codes. They demonstrate their approach on the CUHK03 data set (18.74%). However, their accuracy on the data set shows that the gained efficiency of using hash codes comes at the cost of a decreased accuracy.

Zhang et al. [82] use the null Foley-Sammon transform [22] for learning of a null space in which image embeddings are represented. By mapping image embeddings of the same class to a single point, it reduces the intra-class variance to zero. Its closed form formula and the algorithm's efficiency is a big advantage of their approach. They demonstrate its effectiveness on VIPeR (51.17%), PRID2011 (40.90%), CUHK01 (69.09%), CUHK03 (62.55%) and Market-1501 (61.02%/71.56%). For PRID2011 and VIPeR they investigate what happens when the model is used under semi-supervised conditions as opposed to fully-supervised learning. It shows that both the accuracies then drop to 41.01% and 35.80% respectively.

Wu et al. [71] improve the verification network of Ahmed et al. [2] with their PersonNet, by using more convolutional layers that use smaller convolutional filters. Furthermore, they propose to train the network using RMSprop, a backpropagation scheme capable of handling large variations in the magnitude of the gradients as opposed to the standard stochastic gradient descent. Using these techniques, they improve the state-of-the-art on the CUHK01 (71.14%), CUHK03 (64.80%) and Market-1501 (37.21%) data sets. For future work they note that using different models for different body parts might improve performance.

Chen et al. [5] combine the SCNN approach, that produces a similarity score for its inputs, and the CNN approach, which takes a single image as input. They stitch two input images together and feed it as a single image to an on AlexNet [36] based CNN. This net then produces a similarity score in order to rank gallery images w.r.t a query image. An advantage of this approach is that it can be combined with another (re-)ranking approach, kLFDA [60] for example, to improve accuracy. Their network is considerably larger than many previous models [38][80][14][2], which allows it to overcome transformation and illumination differences. They achieve state-of-the-art results on VIPeR (52.85%), CUHK01 (57.28%). Furthermore, they investigate the use of classification versus ranking. Their conclusion is that the absolute decision made by classification is inferior to the relative rankings used by ranking algorithms.

Xiao et al. [75] design a pipeline for learning generic and robust deep feature representations from multiple domains with CNNs. This is beneficial for training deep learning models for the person re-id task, as those models generally need a lot of data to train while currently there is not much data available. They note that when a CNN is trained using data from various domains some neurons learn domain specific knowledge while others learn representations shared across more domains. In order to improve feature learning under these circumstances they design

Domain Guided Dropout (DGD). This dropout scheme takes the impact that any given neuron has on the final decision of the network as a guide for deciding whether to (temporarily) switch off or not. The higher the impact of a neuron, the more important it is for a given domain and the less chance it has of being turned off by DGD. They demonstrate state-of-the-art results using DGD on CUHK01 (66.6%), CUHK03 (75.3%), PRID2011 (64.0%), VIPeR (38.6%), 3DPeS (56.0%) and iLIDS (64.6%).

Xiao et al. [76] use the VGG-16 model [59] as a basis to create a single CNN that performs both the person detection and re-identification steps, therefore not requiring cropped bounding boxes as input. During training they treat person re-id as a classification task by letting the net classify instances to a specific identity using a softmax layer. However, during testing time they treat person re-id as a ranking problem by removing the softmax layer and using the output of the now last fc layer as an embedding for an identity in the input image. The L2 distance between embeddings is used to construct a ranking. They show their network outperforms baseline alternatives on their own, newly constructed data set called CUHK-SYSU (78.7%).

Wu et al. [72] use densely computed SIFT (i.e. hand-crafted) features, in contrast with previously discussed approaches in which convolutional filters are used to extract image features. After applying PCA they use PCA projected SIFT descriptors to construct Fisher vectors, which form the input to a net consisting only out of fully connected layers. They argue that the limited amount of data available in this domain, and the limited capability of CNNs to model intra- and inter-class variations, means that using convolutional filters is sub-optimal. During training they enforce Linear Discriminant Analysis (LDA) on the network to produce a latent space in which the image embeddings become linearly separable. Their method achieves state-of-the-art on VIPeR (44.11%), CUHK01 (67.12%), CUHK03 (63.23%) and Market-1501 (48.15%).

Wu et al. [70] claim to present the first deep model that makes use of spatio-temporal features in the domain of person re-id. They achieve this by adding Gated Recurrent Units (GRUs) after the convolutional layers in their PersonNet [71], effectively transforming the model into an RNN type of model. By also using temporal pooling they generate a single embedding for an entire video sequence. The KISSME [35] metric is subsequently used to define a ranking between embeddings. Using temporal data also allows them to capture features like gait and pose, but differing frame rates, variable length sequences and occlusions within a sequence makes the training of the model harder. They report state-of-the-art results on iLIDS-VID (46.1%) and PRID2011 (69.0%).

Wang et al. [69] propose a SCNN which is trained to generate both an embedding for the single-image representation (SIR) and the cross-image representation (CIR) of the input images. The two SIRs are compared with each other using the L2 distance, whereas the CIR representation is compared to other CIRs using an (rank)SVM. Both similarity measures are combined to produce a single similarity measure for the two images. The authors mention that SIRs for images can be computed in advance, but CIR images cannot, as they depend on both the input images. Therefore the part of the model that is responsible for computing CIRs, is intentionally kept fast and small. Normally, similarity measures are used in a ranking approach, but by using a threshold value they transform this approach into a classification one. The model achieves performance comparable to state-of-the-art on CUHK01 (71.80%), CUHK03 (52.17%) and VIPeR (35.76%).

Cheng et al. [8] propose a multi-channel parts-based CNN, which can be viewed as an expansion of the model Yi et al. [80] introduced. Like the model of Yi et al., they use different CNNs (called channels) for different horizontal image regions (called parts). However, there are some significant differences: they use four instead of three images segments, the various models do share the first convolutional layer, also there is a CNN which receives the entire image as input and instead of a similarity measure an embedding is produced by the net. These embeddings are ranked according to their respective L2 distances. Furthermore, they advocate the use of an extra convolutional layer when the model is trained on a large data set. During training a combination of the pair-wise and triplet-wise training schemes is used. They report state-of-the-art on iLIDS (60.4%), VIPeR (47.8%), PRID2011 (22.0%) and CUHK01 (53.7%).

Varior et al. [68] use handcrafted features as input to a neural network which does not contain convolutional filters, similar to Wu et al. [72]. However, instead of SIFT, they use LOMO features and instead of fully connected layers, they use a RNN with long short-term memory (LSTM) cells. As their model processes an image from top-to-bottom, it is able to remember important context information. They argue that these cells thus looks at the entire image, whereas convolutional filters can only take look at a small section of it. The L2 distance between the embeddings produced by the network are used to define a ranking for the gallery w.r.t. a query image. They report performance to state-of-the-art methods on CUHK03 (57.3%), but are not state-of-the-art on VIPeR (42.4%) and Market-1501 (61.60%).

Varior et al. [67] introduce a gated function to improve the CNN model for person re-id. Normally a CNN uses only high-level features to decide on its inputs, but the proposed gated functions also allows for mid-level representations to be taken into account. The embeddings produced by this CNN are ranked using the popular L2 distance. They report state-of-the-art on Market-1501 (65.88/76.04%) and CUHK03 (68.1%) and comparable performance on VIPeR (37.8%).

Chen et al. [6] improve on the triplet loss with their proposed quadruplet loss, which requires the minimum inter-class distance to be larger than the maximum intra-class distance. They argue that the triplet loss suffers from a weaker generalization ability than their quadruplet loss because it still causes a relative large inter-class variation, thereby referring to the work of Cheng et al. [8]. Furthermore an extensive comparison between binary classification, the triplet loss and the quadruplet loss is made. Here they note that a weakness of the binary classification approach is that it learns an absolute, instead of a relative similarity threshold. They demonstrate state-of-the-art with their ALexNet [36] based quadruplet network on CUHK01 (81.00%), CUHK03 (75.53%) and VIPeR (49.05%).

Schumann et al. [56] use a Deep Learning model for Domain Perceptive (DLDP) selection to automatically discover prototype-domains in various person re-id data sets. During training a separate model is trained for each of these prototype-domains and during deployment the query image is matched against these domains. The model belonging to the best matched domain is then used to create an embedding for the query image in order to establish a ranking of the gallery images w.r.t. the query image. The authors claim that their approach thus do not need any training data of the target domain in order to attain high accuracy. They perform experiments on CUHK-SYSU [76] (76.7%) and PRW (45.4%).

Geng et al. [20] demonstrate the effectiveness of transfer learning for the person re-identification task. When pre-trained on ImageNet [13] their generic SCNN obtains improved accuracy. Furthermore, they also show that improvement can be gained by using both a verification loss and a classification loss during training. Additionally, they illustrate how to train on a unlabeled re-id data set by use of co-training. State-of-the-art performance of their model is shown on VIPeR (56.3%), PRID2011 (43.6%), CUHK03 (85.4%) and Market-1501 ($83.7sq/89.6mq$%).

Franco et al. [19] take the convolutional features learned by a CNN and use them in a covariance matrix, thereby constructing Convolutional Covariance Features (CCF). They demonstrate state-of-the-art performance on various data sets, but also mention that their technique has still a long way to go until it can be implemented in a real world system. For example, it still suffers from efficiency issues, both with respect to time and complexity.

Liu et al. [43] develop the Accumulative Motion Context (AMOC) network which uses the motion present in a video sequence to yield a descriptor for any individual. They combine a CNN with a RNN to create a generic Siamese model. To train the network they employ both the classification loss and the contrastive loss, like Wang et al. [69] do. They demonstrate state-of-the-art on iLIDS-VID (68.7%) and PRID2011 (83.7%).

Barbosa et al. [4] use a synthetic training data set, called SOMAset, to train a CNN, called SOMAnet, that is based on the Inception architecture. They note that the low resolution images often used for person re-identification allows for the use of synthetic training data. Furthermore, they manage to train their network such that it is able to recognize people based on their height, obesity and/or gender. By training on SOMAset and fine tuning on a specific training set, they reach state-of-the-art performance for CUHK03 (72.40/85.90%) and Market-1501(73.87/81.29%). Also they mention that the Inception architecture allows the network to be easily probed, which aids in identifying which image input regions are important to the net.

Yan et al. [77] design a LSTM network that aggregates image level features and human dynamics information into a single sequence level embedding. Their network is able to remember discriminative information, forget non-informative information and is simple, yet efficient. They demonstrate state-of-the-art on iLIDS-VID (49.3%) and PRID 2011 (58.2%).

Wu et al. [73] improve on the efficiency of existing methods by using a structured deep hashing CNN. Their network simultaneously learns deep features and optimizes a hashing function. Furthermore, they design a new triplet loss that uses hard negatives to make the network converge faster. They demonstrate the efficiency and the state-of-the-art performance of their model on CUHK03 (37.41%) and Market-1501 (48.06%). Their network is based of AlexNet because they compare performance with other papers where AlexNet was used. However, other nets should also be usable. They use the network output plus an additional fully connected layer and sigmoid function, which constitutes the hash layer. During test time the sign of the values in the embeddings is used to construct the hash codes. A positive number becomes a 1 and a negative value becomes a 0.

Chen et al. [7] design a view-specific re-identification framework from a feature augmentation point of view, called Camera corRelation Aware Feature augmenTation (CRAFT). This framework is able to optimize a generic person re-id algorithm with view-specific sub-modules. They claim their approach is also suitable for a network with consisting out of a multitude of cameras. The CNN they employ is based on AlexNet [36] and performs state-of-the-art on Market-1501 (68.7/77.0%), CUHK01 (74.5%), CUHK03 (84.3%) and VIPeR (50.3%).

Lin et al. [41] learn correspondence structures between a pair of input images, i.e. patch matching, with a boosting based method and an additional global constraint. It therefore lies in the realm of the handcrafted methods. They report state-of-the-art performance on VIPeR (51.4%), PRID340S (65.1%) and 3DPeS (61.4%).

Paisitkriangkrai et al. [49] propose a handcrafted approach that uses multiple low-level visual features and an ensemble of various metrics. They demonstrate that an optimization algorithm based on maximizing the correct identifications among top candidates performs better than one based on the triplet framework. They demonstrate the effectiveness of their approach on iLIDS (61.3%), PRID2011 (20.0%), VIPeR (47.7%), 3DPeS (47.9%), CUHK01 (57.5%) and CUHK03 (69.0%).

Yang et al. [78] propose a method that tries to circumvent the shortage of available labeled data in person re-id domain by learning from both labeled and unlabeled data. The algorithm establishes pairwise relationships between pairs of unlabeled data using the k-nearest neighbors algorithm. This data is used together with the labeled data and formulated as an eigenvalue problem. Solving this yields a discriminative projection from the input space to a discriminative subspace. This approach is thus to be classified as a handcrafted method and state-of-the-art is reported on VIPeR (51.0%), CUHK01 (62.0%), PRID2011 (34.2%), PRID450S (66.9%) and 3DPeS (54.4%).

Zhang et al. [84] use multiple CNNs to tackle the video-based person re-id problem. Representative frames are extracted from a sequence, according to the walking profile of a subject. These frames are fed into independent CNNs, which produce an embedding for their respective inputs. Using feature pooling the most descriptive features are then used to form a single descriptor for that person. They demonstrate state-of-the-art on iLIDS-VID (60.2%), PRID2011 (83.3%), SDU-VID (89.3%).

Zhu et al. [91] use a generic SCNN to extract deep features from two input images. Both the absolute difference between the two feature vectors and the result of their multiplication is used to determine a similarity score. Their Deep Hybrid Similarity Learning (DSHL) method advanced the state-of-the-art on the VIPeR (44.87%) and CUHK03 (66.41%) data sets. Furthermore they make a comparison on computation time w.r.t. their method and several other state-of-the-art methods, which is in favour of their approach.

Zhu et al. [90] propose a Part-based Deep Hashing (PDH) model which learns to extract deep features from images simultaneously with the optimization of a hash function. Their approach differs from that of Wu et al. [73] in that they divide an image into four horizontal strips for which different hash codes are calculated by four different models, all based on AlexNet, as opposed to using a single net to calculate a hash code for the entire image. The concatenation of these four hashes form the code for the entire image and is used for ranking the gallery w.r.t. a query image. They demonstrate that their model has state-of-the-art performance on the Market-1501 (56.80% single query) and Market-1501+500k (38.39% single query) data sets, when compared to other hashing based architectures.

Qi et al. [50] adopt and tweak the CNN of Szegedy et al. [64] to extract deep features from images. These embeddings are ranked using an ensemble of various distance metrics (cosine, XQDA). Doing this they demonstrate to achieve state-of-the-art on VIPeR (47.8%), CUHK01 (77.0%), CUHK03 (85.1%), PRID2011 (66.7%), 3DPeS (85.9%) and iLIDS (63.3%).

Lin et al. [42] demonstrate that an on ResNet-50 [24] based baseline CNN can be improved by letting it train on both re-id learning and attribute learning. Combining the re-id loss with that of the attribute loss thus yields a better CNN feature extractor, which they call the Attribute-Person Recognition (APR) network. State-of-the-art performance is reported on DukeMTMC-reID (70.69%) and Market-1501 (84.29% single query). They use the embedding produced by the person re-id part of the APR net in combination with the L2 distance to produce a ranking of the gallery images w.r.t. a query image. Furthermore, they make the claim that the PETA data set does not contain enough training samples to be effective. They try to alleviate this problem by annotating attributes on the Market-1501 and DukeMTMC data sets, which are made publicly available.

Sun et al. [61] view each weight vector within the fully connected layers in a CNN (AlexNet / CaffeNet [36] and ResNet50 [24]) as a projection basis and note that these weights are usually highly correlated. By employing Singular Vector Decomposition (SVD) during training, they enforce the orthogonality constraint on these weights in order to make the weights less correlated. They perform ranking based on the L2 distance between the obtained features. Their best SVDnet (based on ResNet50) thus produces more discriminative descriptors, which they demonstrate on Market-1501 (82.3%), CUHK03 (81.8%) and DukeMTMC-reID (76.7%). Training of the network is performed by a Restraint and Relaxation Iteration (RRI) scheme. This means that the network is first trained until convergence (Initialization). Then the eigenlayer is computed and fixed and again the net is trained until convergence (Restraint). Finally, the eigenlayer is unfixed and the network is trained for a few iterations (Relaxation). Repeating this scheme several times slowly introduces orthogonality in the produced embeddings. The reason for using SVD is that embeddings of different identities should be orthogonal for the L2 distance to be effective. They also demonstrate that this technique is suitable for the VGG-16 net [59]. In order to measure the correlation within a matrix (Weight matrix) they devise a new measure. Furthermore, they mention that re-ranking might improve their method, as might the use of both verification and classification losses as by Geng et al. [20] and Zheng et al. [87].

Ma et al. [44] demonstrate a Time Shift Dynamic Time Warping (TS-DTW) model, which does not require labeled data to learn to match video sequences for the person re-id problem. Their methods uses Spatio-Temporal Pyramid Sequences (STPS), which would categorize their techniques under the handcrafted methods. They demonstrate their performance on the data sets: PRID2011 (41.7%) and iLIDS-VID (31.5%).

Zhong et al. [89] propose a fully automatic and unsupervised re-ranking method for person re-identification. The underlying hypothesis for their method is that if a gallery image is more similar to the probe in k-reciprocal nearest neighbors, it is more likely to be the true match. Given an image, their method encodes the k-reciprocal nearest neighbors into a vector, which is used to calculate the Jaccard distance. Together with the original distance, this Jaccard distance determines the new ranking. They demonstrate the effectiveness of their approach on the Market-1501 (77.11%) and CUHK03 (67.6%) data sets. Their method is independent of the used algorithm to generate the initial ranking. In their experiments they used a combination of a CNN based on ResNet [24] and the KISSME [35] distance metric.

Xiao et al. [74] build on the idea that identity classification, attribute recognition re-identification share the same mid-level semantic representations (i.e. convolutional filters). Based on this they propose a deep learning person re-id method that transfers knowledge of mid-level attribute features and high-level classification features. Their frameworks starts learning on identity classification (using Market-1501) and attribute recognition tasks (using PETA) and transfers learned knowledge to the person re-id tasks (using CUHK03). Furthermore, they extend a RNN containing LSTM cells with a spatial gate so that it is able to pay attention to certain spatial parts in each recurrent unit. Doing this yields state-of-the-art performance on CUHK03 (74.8%). They indicate that integrating person re-id with (single) object tracking can be a useful future direction of research. Also for real world applications of person re-id the influence of object detection and tracking performance has to be taken into account.

Jin et al. [31] improve learned embeddings by adding a Feature Re-Weighting (FRW) layer and employing an additional center loss during training. The FRW layer is a single layer positioned as the last layer, that is responsible for amplifying or reducing the values used to create the embedding. Intuitively they can be used to amplify values belonging to the important center of the image and decrease the values belonging to the edges of the images, for the person re-identification task. Next to the often used identification loss, they also advocate the use of a center loss. By calculating the center of the embeddings belonging to a specific individual, and moving those embeddings towards their center, they reduce the intra-class variance of the embeddings. Distances between embeddings is computed using the L2 metric. They demonstrate the effectiveness of their approach on the VIPeR (50.4%), CUHK01 (70.5%) and CUHK03 (82.1%) data sets.

Li et al. [40] note that many contemporary person re-id systems focus on local or global feature representation alone, in contrast, they propose a CNN (a smaller variant of ResNet-50 [24]) that jointly learns local and global features. This Jointly Learning Multi-Loss (JLML) network uses one loss for local features and another one for global features, which are matched using metrics like the L2 distance to perform ranking. In their work a local features corresponds with features learned from a horizontal stripe of the person bounding box, whereas a global feature is computed using the entire image as input. For all these features a vector representation is computed, which gets concatenated together to form the final image representation. They compute these features by using different CNNs, which all share the same first convolutional layer, like in the work of Cheng et al. [8] Furthermore, a feature selection learning mechanism is introduced to further enhance the learned features. State-of-the-art performance is reported on CUHK01 (%), CUHK03 (83.2%), Market-1501 (85.1%/89.7%) and VIPeR (%).

Li et al. [37] propose an inception SCNN to learn semantic features and combine this with the null Foley-Sammon transform metric learning approach, see also the work of Guo et al. [22]. They perform experiments on Market-1501 (70.61%), CUHK03 (62.03%), PRID-2011 (28.00%) and VIPeR (41.14%), on all these benchmarks they perform well below (at least 10 percentage points) state-of-the-art.

Hermans et al. [26] note that in the field of person re-id the triplet loss is losing ground in favour of the now often used classification loss. See for example the recent work of Zheng et al. [88], Lin et al. [42], Yao et al. [79], Sun et al. [61] and Li et al. [40], who all present state-of-the-art models on the Market-1501 data set. However, Hermans et al. show that a variant of the triplet loss can perform on par with the just mentioned state-of-the-art papers using the classification loss. They demonstrate this, also using k-reciprocal re-ranking [89] and on the Market-1501 [85] (86.67%/90.53%) dataset. Without this re-ranking step the performance drops slightly on Market-1501 (84.92%/90.53%).

Bak et al. [3] use a one-shot learning approach to learn a metric for person re-id that needs far less data to train then many other supervised methods. They first learn a deep texture representation from intensity images with CNNs. Due to training a CNN using only intensity images, the learned embedding is colour-invariant and shows high performance even on unseen data sets without fine-tuning. To account for differences in camera colour distributions, they learn a colour metric using a single pair of ColorChecker images. They thus split the learning of the (KISS) metric into a texture and colour part. A colour invariant intensity metric (texture) is learned first. After which the method only requires a single instance per camera to adjust for the colour as it is perceived by that respective camera. For the colour they use hand-crafted features and they learn a Mahalanobis distance. Furthermore, they actively try to ignore background distortions by separating it from the foreground. They claim their method is competitive with other supervised methods, but only requires a single example rather than hundreds that are required for normally for supervised learning. Experiments were performed on: VIPeR (34.3%), CUHK01 (45.6%), iLIDS (51.2%) and PRID2011 (41.4%). Their one-shot approach is very interesting due to the minimal amount of needed data to train the system.

Fan et al. [17] consider the issue of learning a deep features with only a few or no labels. They propose a Progressive Unsupervised Learning (PUL) method to transfer pre-trained deep representations to unseen domains. It forms an effective baseline for unsupervised feature learning. Their method iterates between pedestrian k-means clustering and fine-tuning of the CNN to improve the original model trained on the irrelevant labelled data set. Initially the CNN trains on a small amount of reliable examples, which are located near to cluster centroids in the feature space. As the model becomes stronger in subsequent iterations, more images are being selected as CNN training samples. Both are thus improved until convergence. They demonstrate that their network outperforms a baseline ResNet-50 based network on DukeMTMC-reID, Market-1501 and CUHK03, but the numbers are still to be updated.

Yao et al. [79] compare the person re-id task with the image retrieval one. Doing this they employ a CNN based on GoogLeNet [63] to extract a coarse query image embedding and several finer ones. The coarse embedding is used to quickly narrow down the search space, after which the finer embeddings are used as robust descriptors to find matches in the narrowed down gallery. They demonstrate their network using the Person-520K data set (which they also introduce) (64.58%), CUHK03 (75.13%) and Market-1501 (84.64% single query).

Zheng et al. [88] use a Deep Convolutional Generative Adversarial Network (DCGAN) [51] to generate artificial training examples for the person re-id problem. Using these extra generated samples they improve on a baseline CNN on the CUHK03 (55.32%, 1.6% improvement), Market-1501 (83.97/88.42%, 4.37% improvement) and DukeMTMC-reID (67.68%, 2.46% improvement) data sets.

A table with an overview of the performance on two important person re-identification datasets is provided in Section 2.2, after the introduction of the respective datasets.

2.2 Person re-identification datasets

The Market-1501 [85] and CUHK03 [39] datasets are used for the experiments. They contain over a 1000 identities each, which are covered by more than 10,000 bounding-boxes generated by the Deformable Parts Model (DPM) [18] pedestrian detector. The amount of images available makes that they are suitable for training convolutional models. The use of the pedestrian detector means that bounding-boxes deviate from ideal ones, as misalignments and false positives are common to appear in the gallery. So methods acting on these datasets need to be able to cope with these imperfections. Also, for both datasets the images are gathered using six cameras. This requires methods with good generalization abilities, instead of camera pair-wise tuned approaches [86]. These properties make that they are good representations of a realistic person re-identification scenario. Furthermore, the Market-1501 dataset comes with an extension that enables testing methods in a large gallery containing more than 500,000 images. Subsection 2.2.1 details the Market-1501 dataset and its extension and subsection 2.2.2 presents the CUHK03 dataset.

2.2.1 Market-1501

Market-1501 [85] is a challenging dataset often used in literature [4, 7, 20, 26, 40, 42, 61, 67, 68, 72, 71, 73, 79, 82, 88, 89, 90]. It consists of 28,849 images with a total of 1501 labelled identities. The data is collected using six different cameras, each with another viewpoint. Every image is defined by an automatically created bounding-box of a person, detected using the DPM. Examples are presented in Figure 2.1. The use of such a detector ensures that the gathered data is a realistic representation of a real-world scenario. Due to this DPM, bounding-boxes are in general imperfect. This can also means that there are misaligned as in Figure 2.2 or that they are false-positives as in Figure 2.3. Furthermore, the authors of Market-1501 also provide an extension to the gallery subset, adding 500,000 distractor images to mimic a realistic setting. Table 2.1 gives a detailed overview of Market-1501.

Table 2.1: Statistics of the (extended) Market-1501 data. The train data is used for the training of a model. During testing the images in the query part are used to rank those in the gallery. Additionally, the distractor images can be added to this gallery.

		camera						
subset	part	1	2	3	4	5	6	All
train	-	2017	1709	2707	920	2338	3245	12,936
test	query	668	530	717	248	622	583	3368
	gallery	2738	2224	3199	1365	2762	3625	15,913
	distractor	165,225	321,255	13,520	0	0	0	500,000

The standard Market-1501 dataset, i.e. without the distractor images, contains a total of 28,849 images with 1501 different identities. The data is split into a training set containing 12,936 images with 751 identities and a test set containing 15,913 with 750 identities. No identities are shared between these two sets. Furthermore, the test set is divided into a query and a gallery part. In the test phase the query images are used to rank the gallery images w.r.t. their similarity. Note that every query image depicts a single person, whereas a gallery image can either depict a person or contain a false positive, i.e. footage of the background. The quality of this ranking determines the quality of the algorithm and is measured using the Cumulative Matching Characteristic or the Mean of Average Precisions, which are fully explained in Section 4.1.1. All identities present in the query set are guaranteed to be present in the gallery set.

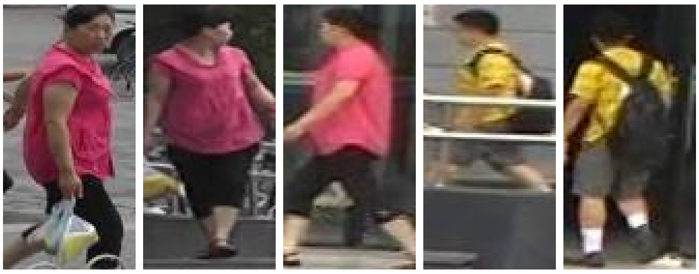

Figure 2.1: Bounding-boxes in the gallery of Market-1501

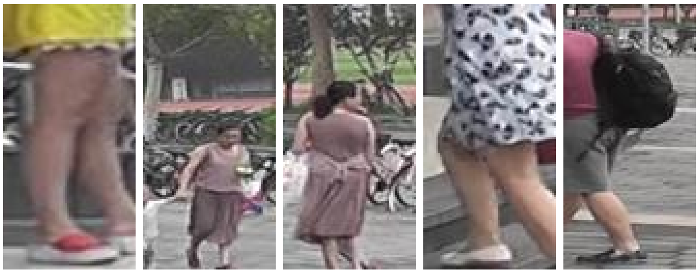

Figure 2.2: Misaligned bounding-boxes in the gallery of Market-1501

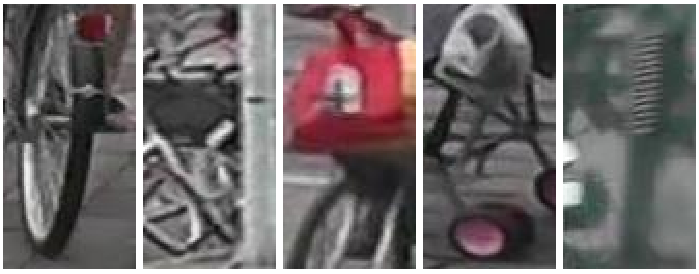

Figure 2.3: False-positive bounding-boxes in the gallery of Market-1501

To the standard test set 500k distractor images can be added, creating the extended test set. The pieces making up this extended test set are thus the query, gallery and distractor parts in Table 2.1. More specifically, these distractor images are added to the gallery set of the test data, the query set remains unchanged. These distractor images do not contain any identities present in any of the other used parts, hence the name. Moreover, they often do not even contain an image of a (complete) person but just random scenery, similar to the images in 2.3. The addition of these images ensures that an algorithm can be tested in a realistic fashion. Since it is likely that in a real-world scenario a gallery contains many distractor images due to the use of an automatic bounding-box detection algorithm like the DPM. The 500k extension is provided by the authors of the Market-1501 dataset. This extended test dataset is used to test the scalability of our approach and is referred to by "extended test set".

An overview of the works listed in Section 2.1 that report results on the standard Market-1501 dataset is presented in Table 2.2. Results are in ascending order of their respective Rank-1 score. The Rank-*r* score indicates the probability that at least a single correct matching images is in the first *r* positions of the ranked gallery. For a detailed explanation of the mean of average precisions (mAP) and Rank-*r* see Section 4.1.1.

Table 2.2: State-of-the-art single query results on Market-1501, ordered by Rank-1 score.

Authors	mAP(%)	Rank-1(%)	Rank-5(%)	Rank-10(%)	Rank-20(%)
Wu [71]	18.57	37.21	-	-	-
Zhu [90]	26.06	47.89	-	-	-
Wu [73]	-	48.06	61.23	75.67	87.06
Wu [72]	29.94	48.15	-	-	-
Zhang [82]	35.68	61.02	-	-	-
Varior [68]	35.31	61.60	-	-	-
Varior [67]	39.55	65.88	-	-	-
Chen [7]	42.3	68.7	-	-	-
Barbosa [4]	47.89	73.87	88.03	92.22	95.07
Zhong [89]	63.63	77.11	-	-	-
Sun [61]	62.1	82.3	-	-	-
Geng [20]	65.60	83.75	-	-	-
Zheng [88]	66.07	83.97	-	-	-
Lin [42]	64.67	84.29	93.20	95.19	97.00
Yao [79]	64.6	84.64	-	-	-
Hermans [26]	69.14	84.92	94.21	-	-
Li [40]	65.5	85.1	-	-	-

2.2.2 CUHK03

CUHK03 [39] is like Market-1501 extensively used in the literature [2, 4, 6, 7, 20, 31, 38, 40, 49, 50, 61, 67, 68, 69, 72, 71, 73, 74, 75, 79, 83, 82, 88, 89, 91]. With 13,164 bounding-boxes of 1360 pedestrians detected with the DPM its size is about a third of that of Market-1501. Additionally 13,165 hand labelled bounding-boxes covering the same identities are also available. Both the automatically detected and labelled boxes are subdivided into train and test subsets. With the test subset having a query and gallery part, as further specified in Table 2.3. Images of CUHK03 are similar to those in the Market-1501 dataset, for which examples are shown in Figure 2.1.

Table 2.3: Statistics of the CUHK03 data.

subset	part	#bounding-boxes	
		labelled	detected
train	-	7368	7365
test	query	1400	1400
	gallery	5328	5332

An overview of the works listed in Section 2.1 that report results on the unlabelled CUHK03 dataset is presented in Table 2.3. Results are in ascending order of their respective Rank-1 score. For a more thorough explanation of the mean of average precisions (mAP) and the Rank-*r* score see Section 4.1.1.

Table 2.4: State-of-the-art results on the unlabelled CUHK03 with the old structure, ordered by Rank-1 score.

Authors	mAP(%)	Rank-1(%)	Rank-5(%)	Rank-10(%)	Rank-20(%)
Zhang [83]	-	18.74	48.39	69.66	81.03
Li [38]	-	20.65	-	-	-
Wu [73]	-	37.41	61.28	77.46	88.42
Wang [69]	-	52.17	-	-	-
Ahmed [2]	-	54.74	-	-	-
Varior [68]	-	57.37	80.1	88.3	-
Zhong [89]	67.6	61.6	-	-	-
Varior [67]	51.25	61.88	80.9	88.3	-
Zhang [82]	-	62.55	90.05	94.80	98.10
Wu [72]	-	63.23	89.95	92.73	97.55
Wu [71]	-	64.80	89.40	94.92	98.20
Zhu [91]	-	66.41	-	-	-
Paisitkriankrai [49]	-	69.0	87.0	92.0	-
Barbosa [4]	-	72.40	92.10	95.80	98.50
Xiao [74]	-	74.8	-	-	-
Yao [79]	-	75.13	90.16	94.92	-
Xiao [75]	-	75.30	-	-	-
Chen [6]	-	75.53	95.15	99.16	-
Sun [61]	84.9	81.8	-	-	-
Jin [31]	-	82.1	96.2	98.2	-
Li [40]	-	83.2	98.0	99.4	99.8
Chen [7]	72.41	84.3	97.1	98.3	99.1
Zheng [88]	87.47	84.6	97.6	98.9	-
Qi [50]	-	85.1	97.5	98.0	99.6
Geng [20]	-	85.45	-	-	-

2.3 Convolutional models

An important property of Convolutional Neural Networks (CNNs) is that these models are able to learn useful features by training them with example data. This ensures that one is not required to come up with a handcrafted feature detector, which often show decreased performance when a large variety of images need to be detected. This explains why the usage of deep learning methods has become increasingly popular in the field of person re-identification. The usage of CNNs enables researchers to design methods that are increasingly able to perform in situations where images are taken from various viewpoints, suffer from distortions, are taken under different lighting conditions or with different types of cameras. However, it is important to note that CNNs are only able to do these tasks when the data they are trained on resembles the data for which they are trained. For the field of person re-identification this means that the datasets need to contain imperfect images which might be distorted and contain identities in different poses and lighting conditions and taken from different viewpoints. And above all, there needs to be a large amount of these images for the CNNs to be able to generalize well enough. It is therefore that this field has seen an increase in the size of the used datasets which are constructed using an increasing number of cameras and filled with imperfect images. Examples of these datasets are the Market-1501 and CUHK03 datasets discussed in Section 2.2.

As mentioned in Section 2.1 state-of-the-art methods for person re-identification mostly use deep learning models. Three examples of this are ResNet50 [25], InceptionV1 (GoogLeNet) [63] and VGG [59]. ResNet50 is used by Sun et al. [61], Lin et al. [42], Li et al. [40] and Zhong et al. [89]. Then InceptionV1, also known as GoogLeNet, is used by Li et al. [37], Barbosa et al. [4], Oh et al. [48] and Yao et al. [79]. The VGG architecture is used by Sun et al. [61] and Xiao et al. [76]. All of these works are discussed in more detail in Section 2.1.

These models were all top performing on the ImageNet challenge, however, more recent models have been developed which obtain a better accuracy and/or have seen a reduction in the number of parameters, making them faster to train and use. Most of the aforementioned papers present general approaches for solving person re-identification using a deep learning model. This means that these approaches can also be applied to newer, improved deep learning models and thereby improving their accuracy. See Table 2.5 for an overview of state-of-the-art deep learning models for computer vision.

Table 2.5: Top performing models on ImageNet, adapted from Chollet [10]. Ordered by Rank-1 score on the ImageNet dataset.

Model	Rank-1(%)	Rank-5(%)	#parameters	#layers
MobileNet [29]	66.5	87.1	4,253,864	88
GoogLeNet [57]	68.7	88.9	11,193,984	22
VGG16 [59]	71.5	90.1	138,357,544	23
VGG19 [59]	72.7	91.0	143,667,240	26
ResNet50 [25]	75.9	92.9	25,636,712	168
InceptionV3 [64]	78.8	94.4	23,851,784	159
Xception [9]	79.0	94.5	22,910,480	126
InceptionResNetV2 [62]	80.4	95.3	55,873,736	572

It is clear from Table 2.5 that the InceptionV3 [64], Xception [9] and InceptionResNetV2 [62] models outperform the others. However, to the best of our knowledge this has not yet been done for the field of person re-identification. This makes them especially interesting for us, as it is probable that combining these networks with top performing person re-identification methods can advance the state-of-the-art. Furthermore, MobileNet [29] is relatively small in size with approximately four million parameters, as that is fewer than half of the number of parameters of GoogLeNet, the second smallest network in presented in the table. This might make it a suitable option for works which require an algorithm to run in a minimum amount of time.

2.4 Model optimizers

Besides selecting a model, it is also important to select a suitable optimizer for training such a model. For this Stochastic Gradient Descent (SGD) [52] is a popular choice in general [53] and in the field of person re-identification. Especially its mini-batch variant is common practice in the deep learning community. It minimizes an objective function parametrized by a model's parameters by updating the parameters in the opposite direction of the gradient of the objective function w.r.t to the parameters [53].

However, there exist several derivatives of SGD which often demonstrate improved performance and require less settings to be experimentally fine-tuned [53]. Some prominent examples are Adagrad [16], Adadelta [81], RMSprop [66], Adam [34], AdaMax [34] and Nadam [15].

Next is a brief overview of these optimizers. The focus is on the intuition behind these optimizers. For the interested reader the technical rapport of Sebastian Ruder [53] might be a valuable source for a more complete description of these optimizers, including their respective definitions.

- Adagrad 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 is an extension of Adagrad that prevents this from happening. When using Adadelta we do not even have to specify a learning rate, it finds a suitable learning rate by itself.

- RMSprop is similar to Adadelta as in that it combats the ever decreasing learning rate problem of Adagrad. It too is an extension of Adagrad but handles the way it solves the problem differently.

- Adam uses an adaptive learning rate like Adadelta and RMSprop, but also makes use of a term which acts like momentum.
- AdaMax is an extension of Adam which tries to make the optimizer more stable and more suitable for sparse data.
- Nadam 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.

A strong case for using one of these optimizers over SGD is that they do not require the user to manually adjust the learning rate during training, but instead use heuristics to decide on what learning rate to use. One only has to choose the learning rate to start with, and in most cases the default learning rate recommended by the respective authors is adequate [53].

To the best of the authors knowledge SGD seems to be the optimizer of choice in the field of person re-identification and other optimizers are rarely taken into account. It is therefore interesting to see how these optimizers can be used to train models for person re-identification and thereby improving their accuracy. Especially as our goal is to find a method that is both fast and accurate. Using an advanced optimizer over standard SGD is an opportunity for improving the accuracy of an already fast method for example.

2.5 Descriptor-reduction methods

A general trend that can be observed in the works discussed in Section 2.1 is that deep learning models are used to construct vector representations for images. The representation of a query images is then compared to those of the gallery images using a distance measure, the Euclidean distance for example. A short distance indicates that the images are similar, i.e. they are likely to belong to the same identity. However, in order to achieve both a high retrieval accuracy and fast response times we need to focus on both the size of these descriptors as their quality. To achieve this, a network that produces small descriptors could be used. However, it is also feasible to use a network to construct a large, information rich image descriptor and then reduce this descriptor using a different method. An example of this is the work of Yao et al. [79]. They use Principal Component Analysis (PCA) [28] to reduce a large 1024 dimensional descriptor to 128 dimensions. This enables their method to keep the average query time under a second, while still being able to achieve state-of-the-art accuracy on the extended Market-1501 dataset containing over 500,000 images.

The reason PCA is suitable for this is twofold. First, it is an operation that preserves most of the important information and tends to keep points that are close to each other near each other. Second, most of the intensive computation can be done in an offline step and only fast computations are needed in the online stage. Specifically, it can compute the new principal components using the entire gallery before any query is issued and new descriptors coming in can easily be transformed into this new system of axes. When a query is actually being issued, the descriptor needs to be computed using the ConvNet. Then this descriptor needs to be transformed using the pre-computed principal components.

This means that, besides PCA, any method adhering to these two properties can be used to perform the needed reduction of the coarse descriptors. Notable examples for which this is true are: kernel PCA [47], sparse PCA [12], dictionary learning [46], factor analysis [32], latent Dirichlet allocation [27], Non-negative Matrix Factorization (NMF) [11], truncated Singular Value Decomposition (SVD) [23] and fast Independent Component Analysis (fast ICA) [30]. What follows is a short introduction of the various techniques, for a more elaborate explanation we refer to the respective sources.

- PCA uses orthogonal basis transformations to find a new coordinate system to represent the given data in. The axes, or principal components, in this new system are constructed such that they correspond to directions of maximal variance in the data. By dropping axes which contain the lowest amount of variance, the dimensionality of the data can be reduced with a minimum loss of information.

- Kernel PCA is non-linear where PCA is a linear process as the new axis are always a linear combination of the old ones. It first maps the data to a new feature space using a non-linear function after which it applies linear PCA.
- Sparse PCA is a way of leaving out certain variables which only marginally contribute to the found principal components. The principal components are usually a linear combination of all variables in the data, but with this method only variables strongly contributing to the principal components are kept.
- Dictionary learning aims at finding sparse codings for the data. However, a difference with PCA related techniques is that it does not require that the various components be orthogonal to each other, granting it a greater flexibility to adapt the representation to the data.
- Factor analysis aims at finding a model relating the old features to less, newer features, like PCA does. But the approach it takes differs as it does this by explicitly defining a linear model relating those features.
- Latent Dirichlet Allocation tries to find hidden variables which explain the observed data is shown. Non-negative Matrix Factorization (NMF) factorizes a matrix, i.e. the matrix, into smaller ones. This is inherently tied to clustering as the columns, or features, of the original samples matrix are a direct result of the multiplication of the factorized matrices.
- Truncated Singular Value Decomposition (SVD) is another technique for factorizing the matrix containing the data sample. However, this method uses a random component to come to its solution. It is especially useful for large amounts of data.
- Fast Independent Component Analysis (fast ICA) has as goal to find a linear representation of non-Gaussian data so that the components are statistically independent.

Given the list of these suitable reduction methods it is interesting to see how they perform when used to reduce the descriptors used for person re-identification. Using these methods we have the opportunity to decrease the size of the descriptors and thereby the time needed to perform the distance calculations. As these methods are especially interesting for us as they are specifically designed to perform their reduction whilst retaining most information, thereby decreasing retrieval time and maintaining good performance.

Finally, we want to point out another reduction method which does not comply with the requirement that most of the computation can be done in the offline step, but still might be useful. This method is t-distributed stochastic neighbour analysis embedding (t-SNE) [45] and is primarily aimed at creating visually pleasing plots of clustered data. It achieves this by taking into account the distribution of the original data, the original descriptors, and that of the lower dimensional descriptions of the data, the reduced descriptors. This method might be useful when accurate descriptors are required, but the amount of descriptors is small or the query time is of less importance.

Chapter 3

Design and implementation

This thesis focuses on the design of a person re-identification system with state-of-the-art performance with regard to both accuracy and query speed. We observe that the field of person re-identification has seen a rapid development in the reported accuracy. For example, the top Cumulative Matching Characteristic (CMC) rank-1 performance on the widely used Market-1501 data set has increased from 37.21% (single query) in 2016 [71] to 85.1% (single query) in 2017 [40]. However, most works discussed in Section 2.1 focus solely on increasing this accuracy, without taking the average query time into account. An exception to this, is the work of Yao et al. [79] which has a focus on fast performance and high accuracy. From Table 2.2 in Section 2.2.1 it can be seen that they achieve state-of-the-art results, which we define as having a Rank-1 higher than 80%. What makes their work stand out is that they achieve this while also focusing on keeping the average query time at an acceptable level while processing a large gallery. To be precise, they report an average query time of 180ms on a gallery containing more than 500,000 images. This is in line with our requirement that the final system needs to be able to perform on a large gallery with hundreds of thousands of images within a second. This makes their work a good starting point for further research.

In short, Yao et al. propose a combination of using their ConvNet with a coarse-to-fine-search (C2F) framework. The ConvNet is responsible for the achieved accuracy and the C2F framework for the obtained speed. To construct this model, they adapt a GoogLeNet design, but as explained in Section 2.3 there are newer models available which demonstrate improved performance over GoogLeNet on ImageNet. In Section 3.1 we first detail how ConvNet is constructed and subsequently propose to investigate if more modern Convolutional Neural Networks than GoogLeNet can be used to improve ConvNet's performance.

Additionally, Yao et al. use Stochastic Gradient Descent to train this ConvNet, but as indicated in Section 2.4 more advanced optimizers are currently available. We therefore wish to investigate if these optimizers can be used to improve the ConvNet.

Furthermore, the C2F framework that is responsible for the fast performance, is detailed in Section 3.2. As further explained in that section, the key to the fast performance is that a coarse descriptor is constructed for which its dimensionality is reduced from 1024 to 128 dimensions using Principal Component Analysis (PCA). Aim of this thesis is to also investigate if this C2F can be improved by reducing the coarse descriptor to another dimensionality. Additionally, PCA is in principle not the only suitable method for reducing the dimensionality of a descriptor, as described in Section 2.5. Therefore we desire to know if another reduction method than PCA is suitable for creating the coarse descriptor for C2F.

Another, more general problem in the field of person re-identification, is the limited amount of available annotated data. This is especially important since deep learning methods perform better in general when trained on larger amounts of data. A work that attempts to alleviate this problem is that of Zheng et al. [88]. Using artificially generated training images they show to improve the training of deep learning models for person re-identification. They achieve state-of-the-art performance on both Market-1501 and CUHK03, for which results are presented in Table 2.4 and Table 2.2, both in Section 2.2. As will be explained in Section 3.3, their method can be used to generate artificial training images which are shown to improve the performance of a deeply learned person re-identification system. We wish to investigate if this method can also be used to improve the training of ConvNet.

3.1 ConvNet

Central to the fast and accurate approach of Yao et al. is the fully Convolutional Neural Network (CNN) ConvNet. Originally the GoogLeNet [63] model forms the basis for this ConvNet, but we propose to also look at other, newer models which are introduced in Section 4.6.

This original ConvNet is created by replacing the fully connected layers in GoogLeNet with a classifier block. This classifier block consists of an additional convolutional layer with C kernels, thereby producing a single feature map of $H \times W$ for every class in the training dataset. For example, $C = 751$ for the Market-1501 dataset and with input images the size of (512×256) the height and width becomes: $H = 16, W = 8$. A schematic of the resulting ConvNet is presented in Figure 3.1

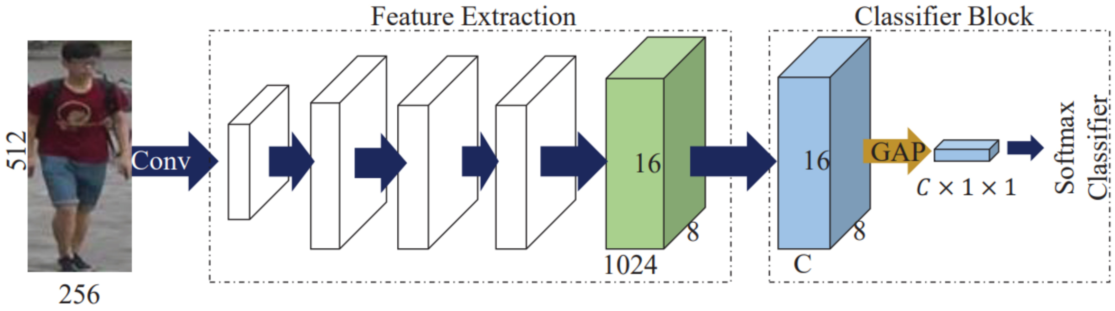

Figure 3.1: Schema depicting ConvNet, adapted from Yao et al. [79]

The values in each feature map are combined into a single average score per feature map. These values represent the confidence the network has that a given image belongs to a certain identity. Global Average Pooling (GAP) is used to create this single value per feature map. The GAP operation does not use trainable parameters and the added convolutional layers contain less parameters than the original dense layer. This reduction in the number of trainable parameters aids in the prevention of over-fitting.

After training, this added classifier block is discarded entirely and the output of the now last convolutional layer of ConvNet is used to generate image descriptors, which is described in Section 3.2.

In general, a model is thus transformed into its ConvNet variant by removing all dense layers present in the network. Then a classifier block with C convolutional filters is appended to this reduced model during training, as is done for the original ConvNet based on GoogLeNet. During testing this classifier block is removed and the output of the now last convolutional layer is taken to create a descriptor. We propose to use this procedure for the model introduced in Section 4.6. These are, listed in ascending order of top-1 accuracy [10] on the ImageNet challenge [54]: MobileNet [29], VGG16 [59], VGG19 [59], ResNet50 [25], InceptionV3 [64], Xception [9] and InceptionResNetV2 [62].

For all networks except for the two VGG variants this means that the single last dense layer is removed. The VGG variants originally end with three dense layers, which will now all be removed. We expect therefore the performance of the VGG networks to be lower than for the other networks. This because they initially depend more on the fully connected layers.

The input image dimensions determines the dimensionality of the final output filters as these networks are now fully convolutional. Image dimensions of 512×256 are used as much as possible for training these ConvNet variants, as suggested by Yao et al. This yields a final ConvNet output of 16×8 for most of the models. This ensures that the models are tested on

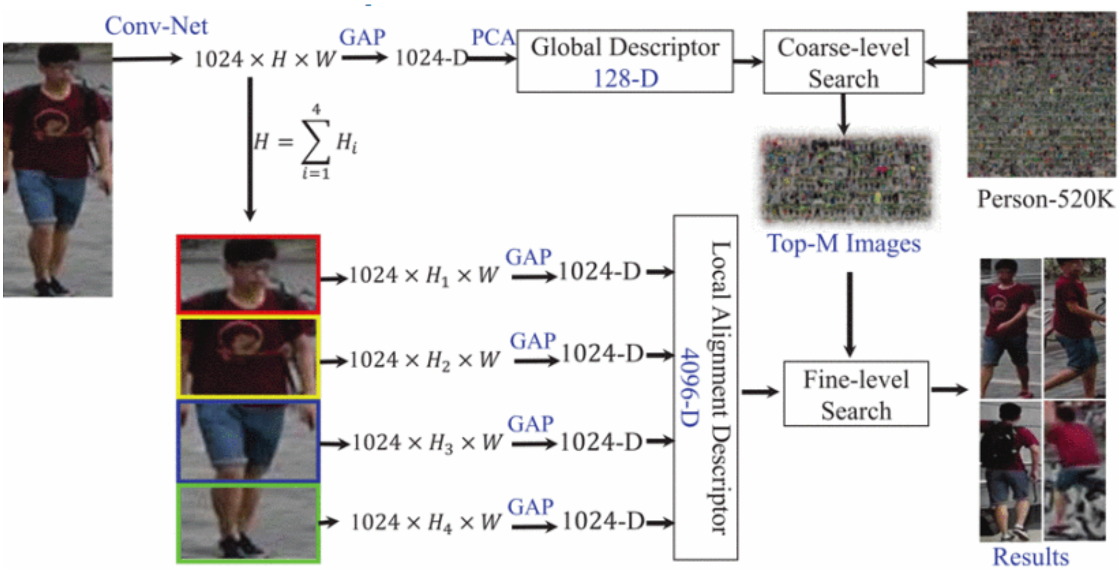

Figure 3.2: Schema depicting coarse-to-fine search when used with ConvNet based on GoogLeNet, by Yao et al. [79]. The 1024 dimensionality depends on the number of filters in the last layer of the used CNN, which is 2048 for Xception.

the same image sizes and at the same time aids in the construction of the descriptors. Especially the construction of the fine descriptor depends on the fact that the number of rows of the final output filters is evenly divisible by 4. This is further explained in Section 3.2.

Unfortunately, this approach is not suitable for the InceptionV3 and InceptionResNetV2 networks, since these setting would yield a final output of 14×6. To alleviate this problem, the image dimensions are set slightly higher to 586×299 for these two networks, which does yield a final output of 16×8 and respects the $2:1$ height-width ratio as much as possible. Also, MobileNet requires its inputs to be square and no larger than 224×224, therefore those dimensions are used for this particular net.

3.2 Coarse-to-fine search

Like Yao et al. [79] we formulate person re-identification as a retrieval problem. This goes along the following lines: given a query image I_q and a set of images forming the gallery $G = \{(I_1, y_1), (I_2, y_2), \ldots, (I_N, y_N)\}$ with I_n an image and y_n the corresponding identity label. The goal is to sort those N images based on their similarity to the query image I_q, more similar images first. This can be represented by the ordered set $\{r_1, r_2, \ldots, r_N\}$ where r_i is the index that would place image I_i in order. The objective function can then be summarized as follows:

$$min \sum_{i=1}^{N} r_i f(y_q, y_i), \quad f(y_q, y_i) = \begin{cases} 1 & \text{if } y_q = y_i \\ 0 & \text{if } y_q \neq y_i \end{cases} \tag{3.1}$$

where y_q is the label for the query image I_q. The objective is thus to place images belonging to the same identity in front of the queue.

Figure 3.2 depicts the working of coarse-to-fine search as proposed by Yao et al. The dimensionalities in this figure correspond to the system which uses descriptors created by the original ConvNet based on GoogLeNet.

In order to produce this queue the coarse-to-fine search framework makes a trade-off between accuracy and speed by using two different descriptors, a global descriptor f^g and a local descriptor f^l. They are used in two different stages, the coarse search and fine search. First, in the coarse search stage, the smaller global descriptor is used to rapidly reduce the search space.

Then during the fine search stage, the larger local descriptor is used to re-order the images in the reduced search space. This process of searching is hence called coarse-to-fine search.

For the construction of these descriptors the classifier block of the used ConvNet is discarded and the output of the now last convolutional layer is used. This output is denoted as:

$$X \in \mathbb{R}^{K \times H \times W} \text{ with } K = 2048, H = 16, W = 8$$

Based on this output X, the global feature descriptor f^g is constructed using:

$$f^g = [f_1, f_2, \dots, f_K]^T, \quad f_k = \frac{1}{W \times H} \sum_{h=1}^{H} \sum_{w=1}^{W} X_{k,h,w}$$

where $X_{k,h,w}$ is the activation value on location (k, h, w). Then to create the coarse descriptor, PCA is used to reduce the dimensionality D of the coarse descriptor. For example, when using GoogLeNet Yao et al. reduce the dimensionality of the coarse descriptor from 1024 to 128.

This global descriptor gives a crude description of the image and due to its reduced size can be used to rapidly reduce the search space. The following local descriptor f^l is then used to search this reduced search space for the true matches. It is assumed that most images display the same person configuration, i.e. the head is in the upper region followed by the torso, upper legs and lower legs. Therefore the local descriptor is constructed out of four horizontal stripes of the last convolutional layer.

$$f^l = [f^{l_1}; f^{l_2}; f^{l_3}; f^{l_4}]$$

where

$$f^{l_i} = [f_1, f_2, \dots, f_K]^T,$$

$$f_k = \frac{1}{(H/4) \times W} \sum_{h=(i-1)*(H/4)+1}^{i*(H/4)} \sum_{w=1}^{W} X_{k,h,w}, \quad i \in [1, 4]$$

Thus local descriptor f^l has dimensions (4×2048). Since both the descriptors are computed using the same activation function, the network needs to be used only once in order to construct them both.

3.3 Artificial training images

By extending the original training images in the Market-1501 dataset with artificially generated images, Zheng et al. [88] demonstrate a way to improve the training of a convolutional network. The reasoning is that the real images will allow the network to perform person re-identification and the generated images are there to prevent the network from overfitting on the training data. The generated images are obtained by using a Deep Convolutional Generative Adversarial Network (DCGAN) which is introduced by Radford et al. [51]. The used implementation is accessible online and provided by Kim [33]. When trained, this DCGAN is able to produce artificial images that are similar to the ones in the Market-1501 training dataset.

After augmenting the training data with these artificial images, they train a slightly adapted ResNet50 [24] using a classification loss. The number of neurons in the original ResNet50 is changed from 1000 to 751, equal to the number of identities in the Market-1501 training data. All images are resized to 256 × 256 using bilinear sampling and a random crop of size 224 × 224 is taken as the input to train the network.

The Label Smoothing for Outliers (LSRO) is employed to train the network. In a nutshell, for the real images this loss acts as a normal cross-entropy loss and for the DCGAN generated images it ensures that the network does not assign a high confidence value to any of them. This is important, since the DCGAN generated images do not have an own, separate class and also do not have the same identity as any of the available classes. Essentially the network is always forced to assign a wrong label to a DCGAN generated image, which acts as a regularization method. However, the LSRO ensures that the network does assign a label to a DCGAN generated image with high confidence. Zheng et al. [88] provide an intuitive example for why this is useful: “If we only have one green-clothed identity in the training set, the network may be misled into considering that the colour green is a discriminative feature, and this limits the discriminative ability of the model. By adding generated training samples, such as an unlabelled green-clothed person, the classifier will be penalized if it makes the wrong prediction towards the labelled green-clothed person.”

Formally, the loss is defined as follows:

$$l_{LSRO} = -(1-Z)log(p(y)) - \frac{Z}{K}\sum_{k=1}^{K} log(p(k))$$

Where $Z = 0$ for a real image and $Z = 1$ for a generated image. In the case of a real image the loss function is the standard cross-entropy loss, whereas in the case of a generated image the loss function prevents the network to predict a high probability for any given class.

Furthermore, they compare the above described method with two other approaches: All-in-one, meaning assign a new label $K + 1$ for all the generated images to fall into, and Pseudo label, which dynamically assigns the real image label having the highest probability for a generated image as being the correct label. Both these methods did work, but are inferior to the LSRO.

Figure 3.3 displays some randomly selected images generated by a DCGAN trained on the Market-1501 dataset. When compared to real images, see Figure 2.1, it is clear that there are similarities between the two sets of images. Although it is easy for a human observer to make a distinction between the two, it is also intuitive that a trained network might benefit from these generated images, as they bear some resemblance to the real images. We indicate the training data that contains these generated images as the “extended training data”.

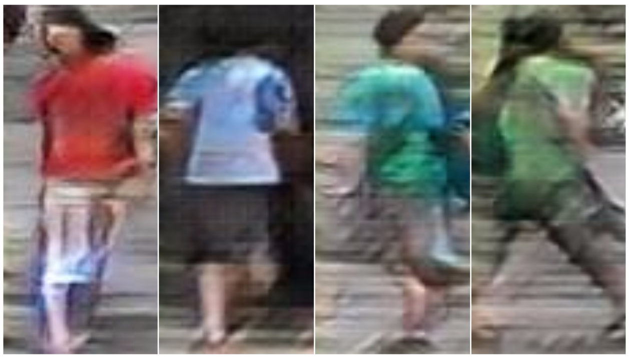

Figure 3.3: Images in the extended train set generated with DCGAN

Chapter 4

Experiments and Results

In Section 4.1 the experimental setup is discussed. With the experiments described in the subsequent sections of this chapter we first aim to replicate the performance of ConvNet using the coarse-to-fine framework (ConvNet+C2F), as reported by Yao et al. [79] in Section 4.2. Furthermore, an attempt is made to provide more insight into the scalability of the ConvNet+C2F approach in Section 4.3. Using this baseline, we continue by investigating the effect of using DCGAN generated images during the training of ConvNet+C2F in Section 4.4. Furthermore, we research ways to improve training using more modern optimizers than standard SGD in Section 4.5. In Section 4.6 we dive into the question if GoogLeNet is not better replaced by another network as a basis for ConvNet+C2F. This is followed by Section 4.7 where we show that the best performing model in the previous experiment also performs well on the CUHK03 dataset, instead of on the Market-1501 used in all other experiments. This is an important indication that the chosen approach generalizes to other data and is not only fit for the Market-1501 dataset. In Section 4.8 it is demonstrated that using ImageNet pre-trained weights is not a necessity to obtain good results. The influence of the used batch size is investigated in Section 4.9. Furthermore, in Section 4.10 we show the effect of using different dimensionality-reduction methods and different amount of dimensions for the coarse descriptor.

4.1 Experimental setup

4.1.1 Evaluation metrics

For the evaluation of methods used in the field of person re-id it is common to report the Cumulative Matching Characteristic (CMC) Rank 1 (Rank-1) and the mean of average precisions (mAP). The Rank-1 reports the probability of a correct matching image to be in the first position in the re-ordered gallery. The mean of average precisions is designed such that it takes into account the index of every matching image and not only the position of the first matching image. Both evaluation metrics are discussed in detail in the following subsections.

Cumulative matching characteristic

The goal of using the Cumulative Matching Characteristic (CMC) is to measure the retrieval precision. When an algorithm returns a list of possible matches for a given query image, then the Rank-r indicates the probability that the first matching image is in one of the first r positions. For example, a Rank-5 of 80 means that in 80% of the cases the algorithm places at least one matching image in the top five returned images. Note that this metric does not take into account every position of every matching image, it only looks at the position of the highest placed matching image. Therefore the mean of average precisions is also often reported alongside this measure.

Mean of average precisions

The Mean of Average Precisions (mAP) is often reported together with the CMC rank to better indicate the quality of a certain retrieval algorithm in the field of person-reid. Where the CMC Rank-1 only takes into account the position of the first image in the re-ranked gallery that matches with the query image, the mAP takes into account the position of all matching images. The mAP works by first calculating the average precision (AP) for each separate query. The mean of every measured AP then constitutes the mAP. By doing this the precision and recall of the algorithm is taken into account.

An example of the mAP compared to the CMC metric is displayed in Figure 4.1. Each row depicts the result of a single query. It can be seen that the Rank-1 of the first and last ranking is equal, as both have a match in the first position. However, the average precision for each query differs, as additional correct matches are placed at different positions. In this thesis more emphasis is put on the mAP since it gives a more complete image of the overall performance of a method.

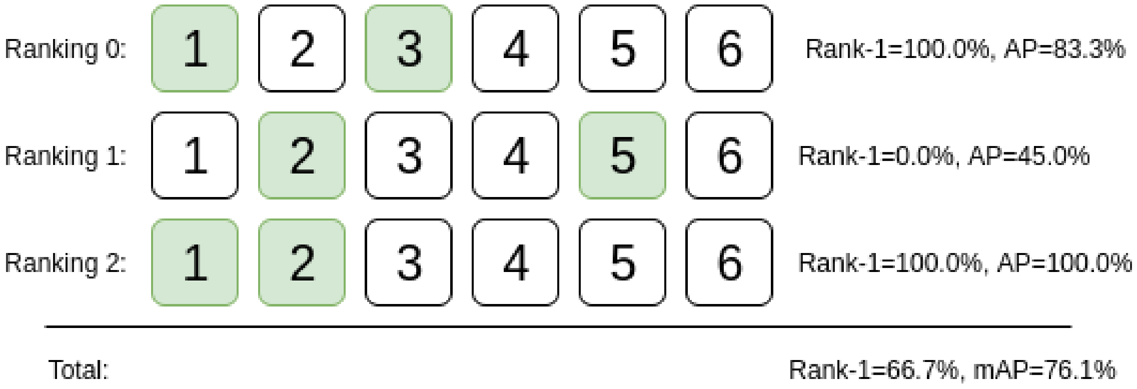
Ranking 0: 1 2 3 4 5 6 Rank-1=100.0%, AP=83.3%
Ranking 1: 1 2 3 4 5 6 Rank-1=0.0%, AP=45.0%
Ranking 2: 1 2 3 4 5 6 Rank-1=100.0%, AP=100.0%
Total: Rank-1=66.7%, mAP=76.1%

Figure 4.1: Comparison of CMC and mAP. Each row depicting the result of a query, with valid matches highlighted.

4.1.2 Distance metrics

Various distance metrics can be used to compute the similarity between the descriptors of the images in question. Yao et al. [79] use the Euclidean distance metric and Zheng et al. [88] use the cosine distance metric. If L2 normalization is used before measuring the Euclidean distance it yields the same results as when the cosine distance is used. The Euclidean distance is used for the experiments in this work.

4.1.3 Data pre-processing and augmentation

The data pre-processing and augmentation step described hereafter are used for all experiments, unless explicitly stated otherwise. As they are common to almost all experiments, they are mentioned here once and only explicitly referred to again when they are not used, or used in a different way.

Most models in the experiments are initialized with ImageNet pre-trained weights. For such networks it is a common practice to subtract the values $104, 117, 123$ from the red, blue and green channels respectively, before putting an image through the network. This is true for both the training and testing data.

Furthermore, as a way to perform data augmentation, images in training data sets are vertically flipped with a chance of 50 before being put through a network.

4.1.4 Machine configuration and software platforms

Both training and testing is performed using two machines. One machine is equipped with an Nvidia Titan X as GPU and an Intel Core i7-4790 CPU. The other machine is equipped with six Nvidia GTX 980 Ti GPUs and two Nvidia Titan X GPUs combined with a Intel Xeon E5-2650 v30 CPU.

For this work both Caffe [58] and Keras [10] with Tensorflow [1] for a back-end are used. More specifically, the experiments in Sections 4.2, 4.3 and 4.4 use exclusively Caffe and all others use exclusively Keras.

4.2 Coarse-to-fine search accuracy

The aim of this experiment is to train GoogLeNet and ConvNet+C2F for person re-identification, as described by Yao et al. [79]. These results can then be used as a baseline for the subsequent experiments.

Where GoogLeNet can serve as a basic approach for solving the person re-id problem, ConvNet+C2F is a more advanced method. The latter technique yields a higher accuracy since it uses classifier blocks instead of fully connected layers. Additionally the coarse-to-fine search should yield faster queries. Caffe [58] is used as the deep learning framework to implement the models. This to ensure a fair comparison with the work of Yao et al, who also make use of Caffe. Both models are trained on standard Market-1501. Settings are taken from the work of Yao et al. where possible. The networks are initialized with ImageNet pre-trained weights, which are obtained via the Caffe distribution [58]. Stochastic Gradient Descent (SGD) is used to optimize the model's weights during training. For ease of reference the detailed training settings are listed in Table 4.1.

Table 4.1: Settings coarse-to-fine search accuracy

Setting	Value
Batch size	32
Learning rate	0.01
Learning rate decay factor	0.75
Learning rate decay steps	2500
Total steps	50,000
Momentum	0.9
Weight decay	0.0005
Image dimensions GoogLeNet	224 × 224
Image dimensions ConvNet+C2F	512 × 256

The "learning rate" is the initial learning rate used when training is started. The "learning rate decay factor" is the factor used to change the learning rate during training. This is done every "learning rate decay steps". The "momentum" is a factor used to steer the gradient descent algorithm in the general direction indicated by the previous steps. The "weight decay" acts as a regularizer for the training of the weights by penalizing large values. Also note that the input image dimensions differ for GoogLeNet and ConvNet+C2F. The larger input images needed to train ConvNet+C2F are there to ensure that the descriptors are large enough to be effective. Furthermore, the $2:1$ aspect ratio improves its performance, as shown by Yoa et al. The fully connected layers in GoogLeNet force us to choose between changing the input image dimensions or initializing the network with pre-trained weights. Similar to the work of Yao et al. we choose the latter. Note that the descriptor sizes are used as they are proposed by Yao et al., thus a coarse descriptor with 128 dimensions and a fine descriptor with 4096 dimensions.

Note, however, that both the momentum and weight decay are not mentioned by Yao et al, but omitting these settings lets the Rank-1 accuracy drop with more than 20 percentage points.

Results are listed in Table 4.2. In spite of our best efforts to exactly replicate their baseline, our implementation of the baseline GoogLeNet obtains a higher accuracy than the one presented

Table 4.2: Accuracy comparison of GoogLeNet and ConvNet+C2F implementations in Caffe. The bottom two rows present our work.

Model	mAP(%)	Rank-1(%)
GoogLeNet, Yao [79]	54.9	76.4
ConvNet+C2F, Yao [79]	64.6	84.6
GoogLeNet	58.6	79.7
ConvNet+C2F	54.1	79.5

by Yao et al. It achieves a mAP of 58.6% compared to the 54.9% of Yao. However, our version of ConvNet+C2F demonstrates a decreased mAP accuracy of 54.1% compared to our baseline GoogLeNet.

We assume that this discrepancy in accuracy can be explained by the difference between the two implementations. This is put to the test in Section 4.6, where we investigate the usage of different base models for ConvNet+C2F in the Keras [10] framework.

4.3 Coarse-to-fine speed-up

We aim to achieve a speed-up by using ConvNet+C2F instead of GoogLeNet, as reported by Yao et al. This speed-up is defined as the average query time won by using the C2F framework. Remember, this speed-up is to be expected because the default method compares a large query descriptor of 1024 dimensions with all of the gallery descriptors, whereas the C2F method uses a much smaller 128 dimensional descriptor for this step. Only in the second fine step does C2F use a larger descriptor of 4096 dimensions. Finer details of both methods are presented in Section 3.2.

For this experiment the same setup is used as is described in Chapter 4.2, also shown in Table 4.1. Note that for his experiment the settings are used as they are proposed by Yao et al. Thus the coarse descriptor has a dimensionality of 128 and the fine descriptor has a dimensionality of 4096. Results are presented in Table 4.3.

Table 4.3: Average query time comparison of GoogLeNet and ConvNet+C2F implementations in Caffe. The bottom two rows present our work.

	Market-1501			Market-1501+500k		
	Coarse(ms)	Fine(ms)	Total(ms)	Coarse(ms)	Fine(ms)	Total(ms)
GoogLeNet, Yao [79]	-	-	-	-	-	960
ConvNet+C2F, Yao [79]	-	-	-	-	-	180
GoogLeNet	23	N/A	23	728	N/A	728
ConvNet+C2F	4	9	13	219	16	235

We observe that by using the coarse-to-fine search framework a speed-up is achieved on both the normal Market-1501 test set and its extended version. On the standard test set a speed-up of 10 milliseconds is achieved, which amounts to an average query time decrease of 43%. On the extended test set a speed-up of 493 milliseconds is obtained, meaning that the average query time decreases with 68%.

Our experiments thus show that adding 500,000 extra gallery images to the 15,913 default images, increases the average query time with a factor 32 from a total of 23 to 728 milliseconds for GoogLeNet. Thus an increase with a factor of 27 in the gallery size amounts to an increase with a factor of 32 in the average query time. This compared to ConvNet+C2f, which sees an increase in average query time with a factor 18 from 13 to 235 milliseconds.

In comparison, Yao et al. [79] measure an average query time of 960 milliseconds using the GoogLeNet baseline network on the extended Market-1501+500k test dataset. When employing the coarse-to-fine search method they report a speed-up to 180 milliseconds on average, a decrease of 81.25%.

Our results are comparable to those of Yao et al. Their larger percentual decrease of average query time is partly due to their slower baseline. Compared to our baseline they demonstrate a decrease of 0.75%. The remaining difference may be explained due to differing implementations, but is small enough for us to view our version as effective.

Furthermore, we also take a look at the scalability of this approach. It is clear that GoogLeNet scales worse than ConvNet+C2F. Where the former sees its average query time per image in the dataset increase faster than the increase of the gallery size, the latter demonstrates a slower growth in average query time compared to the gallery size growth. This is mainly due to the small size of the coarse descriptor, which is used to trim down the search space.

4.4 DCGAN training

This experiment investigates if the usage of DCGAN generated images during training is beneficial for the accuracy of ConvNet+C2F. To demonstrate this we first wish to show that the performance of the GoogLeNet baseline can be improved by using DCGAN generated images during training, as GoogLeNet forms the basis of ConvNet+C2F. See Table 4.4 for the settings used during training. All images are resized to 256 × 256 using bilinear sampling and a random crop of size 224 × 224 is taken as the input to train the network. The Label Smoothing for Outliers (LSRO) is employed to train the network. To create the extended training set 240,000 artificially generated images are added to the standard training set. Zheng et al. demonstrated that this is an effective number of artificial images to improve training.

Table 4.4: Settings DCGAN training

Setting	Value
Batch size	32
Learning rate	0.01
Learning rate decay factor	0.1
Learning rate decay steps	50,000
Total steps	150,000
Momentum	0.9
Weight decay	0.0005
Image dimensions	224 × 224

However, as described in the Section 4.4.1, the usage of DCGAN generated images for GoogLeNet training is not straightforward. Therefore we also make use of a more sophisticated approach in Section 4.4.2.

4.4.1 Standard training

Using the DCGAN generated images right from the start when training GoogLeNet is a reasonable first approach. In this manner the GoogLeNet can benefit from the regularization offered by the usage of the generated images throughout the entire training process.

We use the experimental setup for GoogLeNet as described in Section 4.2 with the difference that we now train the network on the extended train set containing the 24,000 generated images.

However, experimentation showed that after adding the DCGAN generated images to the training set and starting the training of the GoogLeNet network, the training loss becomes infinite and thus prevents the network from learning. Results for this observation are omitted.

The reason for this is that the generated images cause high loss values during training. This, combined with the fact that computers use limited precision numbers, causes the loss value to become infinite. A common case when this phenomenon occurs is when training is started with a too high learning rate, causing the loss value to explode during training. For example, training GoogLeNet on the standard Market-1501 dataset with Stochastic Gradient Descent and a learning rate of 0.1, instead of 0.01, also lets the loss value become infinite. However, in this case we cannot start with a lower learning rate as that prevents the successful fine-tuning of the network on the Market data. It is therefore concluded that starting the training on both the normal and the generated images is not feasible for this experiment.

4.4.2 Pre-training and fine-tuning

A solution to the infinite loss problem described in Subsection 4.4.1 is to pre-train GoogLeNet on the standard Market train set until its loss value has stabilised to low value. Then subsequently fine-tuning on the extended Market train set whilst using the same learning rate as before thus circumvents the infinite loss error, while still allowing the use of a high enough learning rate.

To this end this experiment is set up in which GoogLeNet is first trained on the standard Market-1501 dataset. Once the loss of the model starts to approach zero, the DCGAN generated training images are added to the training set. The training scheme used here is as follows. Fine-tuning is started with a learning rate of 0.01 and the learning rate is decreased with a factor 10 every 50,000 steps. Training is continued for 150,000 steps, where after the learning rate has decreased to 0.0001 and no further performance gains are to be expected. Furthermore, during training the loss value of GoogLeNet on the training data is monitored in order to measure the effect that the generated images have on this value.

Results are listed in Table 4.5. Model A-x corresponds with GoogLeNet trained on the standard train set, i.e. the pre-training step. We have three variants of A, numbered zero to two, each subsequent number indicates that GoogLeNet is pre-trained for 50,000 more steps. Model B is created by continuing with fine-tuning on the extended train set where model A-0 stopped. Model C is created in the same manner, but now by continuing where model A-1 stopped.

Table 4.5: Pre-training and fine-tuning GoogLeNet on standard and extended Market-1501 respectively. Model A is GoogLeNet trained using real images. Model B is a continuation of model A at snapshot A-0, but with generated images added. Likewise, model C is a continuation of model A at snapshot A-1.

Model	Pre-train steps	Fine-tune steps	mAP(%)	R1(%)	R5(%)	R10(%)	R20(%)
A-0	50,000	0	52.7	76.8	89.6	92.6	95.3
A-1	100,000	0	59.6	80.6	91.8	94.7	96.5
A-2	150,000	0	59.5	80.5	91.7	94.7	96.7
B	50,000	100,000	59.1	80.3	91.7	94.6	96.9
C	100,000	50,000	57.8	80.0	91.6	94.2	96.2

Furthermore, the training loss of model B is plotted in Figure 4.2. Note that every 50,000 steps the learning rate is dropped and that after the first 50,000 steps the generated images were added.

For GoogLeNet pre-trained on the standard train set the following is observed. Training A-0 for 50,000 steps with a learning rate of 0.01 yields a mAP of 52.7%. Then continuing to A-1, it is observed that further training with a learning rate of 0.001 improves the accuracy to 59.6%. But continuing training to A-2, it can be observed that another 50,000 steps with a learning rate of 0.0001 does not have a large influence on the mAP any more. It is clear from this that the most ground is covered during the first 50,000 training steps when the learning rate is still set to 0.01.

The fact that most of the reduction in the loss value is obtained in the first 50,000 steps of the training supports the claim that the training needs to start with a learning rate of 0.01 in order to obtain a reasonable accuracy. Furthermore, the training loss with the 0.01 learning rate stagnates well before the first 50,000 steps have passed. This indicates that this amount of steps

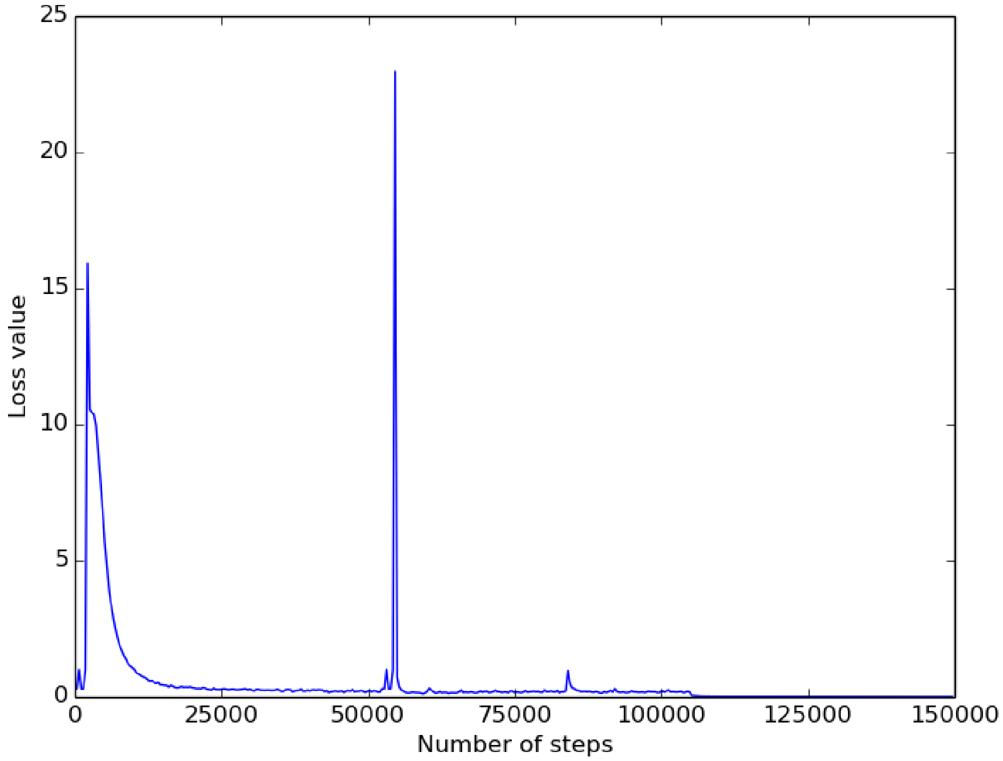

Figure 4.2: The GoogLeNet training loss during 50,000 steps of pre-training and 100,000 steps of fine-tuning.

is a generous estimate for the needed number of training steps with this particular learning rate. Only when the learning rate is dropped to 0.001 after 50,000 steps the accuracy improves again, as is indicated by the accuracies of A-1. On the other hand, it is also clear from the results in Table 4.5 that it does not hurt the model's accuracy when training it for a prolonged period with a stagnated training loss. In Figure 4.2 it can be seen that in the first 50,000 steps the loss starts high and slowly converges to zero. Then, when the generated images are added after the first 50,000 steps, a sharp peak of the loss value is observed. The reason the network now can handle this increase of the loss value caused by the generated images, is that the loss generated by the standard images is already low.

This approach thus indicates that GoogLeNet can be trained using the DCGAN generated images. From the results in Table 4.5 it is inferred that the use of these images during training is not beneficial for the accuracy of the network trained in Caffe.

Why is this the case? First, Zheng et al. used DCGAN generated images to improve the ResNet50 mAP accuracy from 51.5% to 55.1%. However, GoogLeNet trained without generated images already achieves a performance of 58.6%. The room for improvement by using the images as regularization is already used by the intrinsic regularization of GoogLeNet itself. Therefore training with these generated images does nothing for the model's accuracy.

Another option might be that since GoogLeNet is superior to ResNet50, it is able to distinguish the generated images from the real images during training and thereby negating the regularization effect that the usage of the generated images initially offered.

Finally, note that training using the generated images is independent from ConvNet+C2F and the used Caffe implementation, we find it therefore not necessary to repeat this experiment using the Keras implementation used from Section 4.5 and onward.

4.5 Optimizers

Using a more suitable optimizer might be another way of improving the training of ConvNet. In this section it is investigated if there is a better model optimizer to train the network for person re-identification which yields a better accuracy than standard Stochastic Gradient Descent (SGD) [52]?

Yao et al, Zheng et al. and many others [86] in the field of person re-id use SGD to train their respective models. When using SGD one needs to choose the learning rate and its decay by hand. The values chosen for these parameters are of great influence on the training and the final accuracy of any model. Both Yao and Zheng do not present results investigating the chosen SGD settings. Furthermore, there also exist more advanced optimizers than SGD. For example Adagrad [16], Adadelta [81], RMSprop [66], Adam [34], AdaMax [34] and Nadam [15], which are all extensions of standard SGD and often work better in general [53].

The advantage that these optimizers have over SGD is that they are able to automatically tune the used learning rate during training. Usage of such optimizers thus could improve training of a deep learning model for person re-id. To the best of the authors knowledge SGD seems to be the standard optimizer in the field of person re-identification and optimizers like Adadelta have not been used yet. For this experiment the optimizer settings are used as they are proposed by there respective authors. These settings are presented in Table 4.6.

Table 4.6: Overview of various Stochastic Gradient Descent (SGD) variations with recommended settings [53]

Optimizer	Initial learning rate	Decay	ρ	ϵ	β_1	β_2
Adagrad [16]	0.01	0	–	1e−8	–	–
Adadelta [81]	1	0	0.95	1e−8	–	–
RMSprop [66]	0.001	0	0.9	1e−8	–	–
Adam [34]	0.001	0	–	1e−8	0.9	0.999
AdaMax [34]	0.002	0	–	1e−8	0.9	0.999
Nadam [15]	0.002	0.004	–	1e−8	0.9	0.999

In contrast with the experiments in Sections 4.2, 4.3 and 4.4, Keras [10] with TensorFlow [1] as backend is used for this experiment and all the following experiments. Keras is a deep learning framework like Caffe, but based on TensorFlow [1] and Theano [65]. Since ResNet50 is publicly available in Keras, and GoogLeNet is not, we use ResNet50 as the model in the following experiment. For each optimizer we train ResNet50 on the standard Market-1501 dataset for 50 epochs. Results are listed in Table 4.7.

Table 4.7: ResNet50 performance with various optimizers on Market-1501 implemented in Keras, ordered by mAP.

Optimizer	mAP(%)	Rank-1(%)	Rank-5(%)	Rank-10(%)	Rank-20(%)
RMSprop	29.8	58.5	78.1	84.4	90.0
Adam	31.3	57.1	77.3	83.0	87.6
Nadam	36.1	63.6	80.8	86.3	90.9
Adagrad	36.7	62.0	81.8	87.4	91.5
SGD	46.4	70.5	86.3	90.9	93.9
AdaMax	46.5	72.0	87.5	91.2	94.1
Adadelta	59.2	80.7	91.8	95.0	96.6

These results indicate the effect an optimizer has on the performance of the network. For example, ResNet50 trained on standard Market-1501 using the Adadelta optimizer yields a mAP of 59.2%. It thereby performs almost on par with our implementation of GoogLeNet, when trained as described in Section 4.4. Furthermore, this is better than the mAP of 51.48% reported for the with SGD trained ResNet50 baseline by both Yao et al. and Zheng et al. Note that the mAP of the SGD reported here is different from that reported in literature as the default SGD settings were used. However, Adadelta still demonstrates superior performance to the results reported in literature.

The fact that Adadelta performs so well in this experiment and that it is able to handle the adjustment of the learning rate during training shows that it is a suitable candidate to replace SGD as an optimizer for the ConvNet+C2F framework. Therefore Adadelta is used for the subsequent experiments as the optimizer to train models.

4.6 Models for person re-identification

The main goal of the experiments in the following subsections is to investigate whether other base models for ConvNet also show improved performance. While ConvNet+C2F achieves state-of-the-art results when using GoogLeNet as a basis, there are no comparisons between GoogLeNet and other networks as a basis for ConvNet+C2F. This is especially interesting since there are more recent networks available than GoogLeNet, which show improved performance on the ImageNet challenge [54].

Similar to the experiment in Section 4.4.2, Keras with Tensorflow is now used to implement and train the models. This allows to check whether a different training implementation than Caffe demonstrates improved performance of using the ConvNet approach.

Furthermore, all models are trained using Adadelta as an optimizer, such that it can be verified that Adadelta does not only work well in combination with ResNet50, as was shown in Section 4.5.

The results are split out such that it can be indicated how much ConvNet is responsible for the change in accuracy and what is caused by the use of the coarse-to-fine search framework. In Subsection 4.6.1 it is investigated how well several models perform for person re-identification without applying the ConvNet modifications or using the coarse-to-fine search framework, both described in Section 3.2. In Subsection 4.6.2 the networks are modified in the same manner as GoogLeNet was in order to create ConvNet. At last the modified models are used with the coarse-to-fine search framework in Subsection 4.6.3, to measure their final performance.

4.6.1 Baselines

In this experiment a baseline performance is established for several models for person re-identification. The models that are used are introduced in Section 2.3, which are all top performing models for the ImageNet challenge.

Specifically, the following models are looked at, listed in ascending order of top-1 accuracy [10] on the ImageNet challenge [54]: MobileNet [29], VGG16 [59], VGG19 [59], ResNet50 [25], InceptionV3 [64], Xception [9] and InceptionResNetV2 [62].

All models are used as they are proposed by their respective authors, using the output to the final softmax layer to construct the needed descriptors. Exception to this are the descriptors generated by the VGG networks. These networks end in fully connected layers, which makes this approach unsuitable. The fully connected layers do not maintain a spatial ordering for the values and this shows extremely poor performance in conducted tests. Instead, the input to the first fully connected layer is used to create the descriptors for these networks. Each descriptor is created by applying Global Average Pooling (GAP) per convolutional filter output. Note that in this experiment the models are not trained with the classifier block and do not use the coarse-to-fine search framework for retrieval.

The dimensionality of a descriptor varies with the model that is used to produce it, because this dimensionality directly depends on the number of filters in the last convolutional layer. This dimensionality is recorded in order to find correlations between the corresponding accuracies and average query times.

Each model is trained for 75 epochs on the standard train set using Adadelta as an optimizer. The choice for Adadelta is based upon the results presented in Section 4.5, which demonstrate the effectiveness of Adadelta. The settings proposed by its author are used, as described in Section 2.4. Images are fed to the networks in batches of 16 during training. Images are resized to 224 × 224. This so that the fully connected layers in those models can be initialized using the ImageNet pre-trained weights, as is also described in Section 4.2.

The model which shows the best performance is trained for five times in order to validate its results. This is needed since the training of each model is a stochastic process. The extensive training time of each model does not permit us to do this for all results. Results are reported in Table 4.8.

Table 4.8: Performance of baseline models on Market-1501, ordered by mAP.

Model	mAP(%)	Rank-1(%)	avg.-query-time(ms)	descriptor-size
VGG19	25.4	46.5	10	512
VGG16	26.4	49.3	10	512
ResNet50	49.4	73.2	39	2048
InceptionV3	49.8	74.1	39	2048
InceptionResNetV2	51.9	76.9	29	1536
MobileNet	54.9	79.2	20	1024
Xception	57.5	79.0	39	2048

The two VGG models, VGG19 with a mAP of 26.4% and VGG16 with a mAP of 26.4%, perform almost twice as bad as the ResNet50 model, which has a mAP of 49.4%. It is followed by the InceptionV3 and the InceptionResnetV2 models with 49.8% mAP and 51.9% mAP. The last two models, MobileNet and Xception, show an mAP performance of 54.9% and 57.5% respectively.

As anticipated it can be seen that the VGG models perform badly. We suspect this is due to the fact that they now cannot use the fully connected layers. However, another possibility is that their relatively large number of parameters makes them over-fit on the training set. This experiment is not conclusive in this matter. The other models, however, do show reasonable performance. The gap between the lowest mAP of those, ResNet50, and Xception is with 8.1 absolute difference considerable. While, it is not clear from these results how the models perform in the ConvNet+C2F settings, it is likely that Xception will come out on top.

4.6.2 ConvNets

In this experiment the effect of transforming the models from Subsection 4.6.1 into a ConvNet variant is tested, i.e. with a classifier block appended as described in Section 3.2.

A model is transformed into its ConvNet variant by removing all dense layers present in the network. Then a classifier block with 751 convolutional filters is appended to this reduced model during training, as is done for the original ConvNet based on GoogLeNet. During testing this classifier block is removed and the output of the now last convolutional layer is taken to create a descriptor. This is done by using Global Average Pooling on each convolutional filter. This method is described in greater detail in Chapter 3. For all networks except for the two VGG variants this means that the single last dense layer is removed. The VGG variants originally end with three dense layers, which will now all be removed. We expect therefore the performance of the VGG networks to be lower than for the other networks. This because they initially depend more on the fully connected layers.

As in Section 4.6.1 each model is trained for 75 epochs on the standard train set using Adadelta as an optimizer. Image dimensions of 512 × 256 are used as much as possible for training the ConvNet variant, as suggested by Yao et al. This yields a final ConvNet output of 16 × 8 for most of the models. Unfortunately, this approach is not suitable for the InceptionV3 and InceptionResNetV2 networks, since these setting would yield a final output of 14 × 6. This especially complicates the extraction of the four parts of the fine descriptor, as 14 is not evenly divisible by 4. To alleviate this problem the image dimensions are set slightly higher to 586 × 299 for these two networks, which does yield a final output of 16 × 8 and respects the 2 : 1 height-width ratio as much as possible. Also, MobileNet requires its inputs to be square and no larger than 224 × 224, therefore those dimensions are used for this particular net. Note that these models do not use the coarse-to-fine retrieval framework. Results are listed in Table 4.9.

Table 4.9: Performance of ConvNets adapted models without C2F on Market-1501, ordered by mAP.

Model	mAP(%)	Rank-1(%)	avg.-query-time(ms)	descriptor-size
VGG16	24.6	48.2	10	512
VGG19	24.8	49.0	10	512
ResNet50	57.2	82.2	39	2048
InceptionV3	61.1	82.7	39	2048
MobileNet	62.1	84.7	20	1024
InceptionResNetV2	63.2	82.9	29	1536
Xception	64.2	83.8	39	2048

Again a reduced performance is measured for the VGG models, with a mAP of 24.8% for VGG19 and a mAP of 24.6% for VGG16. Furthermore, a small gap appears between the ResNet50 model and the other better performing models. It achieves aa mAP of 57.2% where InceptionV3, MobileNet, InceptionResNetV2 and Xception obtain mAPs of 61.1%, 62.1%, 63.1% and 64.2% respectively. Xception thus achieves the best score in this experiment.

Again it is confirmed that the VGG models are not suitable for this specific approach. Somewhat surprising is the fact that MobileNet seems to perform on par with the other, larger models. Especially since its descriptor is relatively small. Furthermore, except for the VGG models all models perform far better now that they are trained with the classifier block, as opposed to without as shown in Subsection 4.6.1. This shows the usefulness of using the ConvNet approach.

4.6.3 ConvNets with coarse-to-fine search

In this experiment the effect of using the coarse-to-fine search framework for the models trained in Subsection 4.6.2 is investigated.

Thus for each trained model coarse and fine descriptors are generated as they are created for ConvNet, as described in Section 3.2. In short PCA is used to reduce the dimensions of the coarse descriptor, after which the larger fine descriptor is used to re-rank the pre-selection made in the coarse step. Note that only the usage of the coarse-to-fine search framework os added, which does not require to re-train the models. Furthermore, PCA is used to reduce the dimensionality of the coarse descriptors for all ConvNet models to 128, as per paper of Yao et al. [79]. The results are presented in Table 4.10.

Table 4.10: Accuracy for various models as a basis for ConvNet+C2F, ordered by mAP.

	accuracy(%)		avg query time(ms)			descriptor size	
Model	mAP	Rank-1	coarse	fine	total	coarse	fine
ConvNet+C2F, Yao [79]	64.6	84.6	–	–	8	128	4096
VGG19	32.0	53.5	5	14	19	128	2048
VGG16	34.0	55.3	5	14	19	128	4096
MobileNet	54.4	81.1	5	11	16	128	6132
InceptionV3	61.4	82.1	5	14	19	128	2048
InceptionResNetV2	63.3	84.4	5	8	13	128	8192
ResNet50	64.8	86.4	5	3	8	128	8192
Xception	66.5	85.8	5	3	8	128	8192

It can be observed from the table that the Xception model demonstrates the best performance, which was also the case in Subsection 4.6.1 (baseline) and Subsection 4.6.2 (ConvNets). Therefore this experiment is repeated for Xception for an additional four times to verify its validity. This results in an average mAP of 66.5% and an average Rank-1 of 85, 8%, as reported in the table. Furthermore, the respective standard deviations are 0.7 and 0.5.

Now that a mean and standard deviation of the mAP is known, a one sample t-test can be performed in order to test whether the difference in performance between Xception and its closest competitor ResNet50 is significant. With a standard deviation of 0.7 the standard error is $\frac{0.7}{\sqrt{5}} = 0.31$ and thus the t-statistic is calculated by:

$$\text{t-statistic} = \frac{66.5 - 64.8}{0.31} = 5.43$$

The sample size of five has four degrees of freedom, resulting in a p-value of 4.60 for $p = 0.01$. As the t-statistic is higher than this, it can be concluded that the chance that the result of ResNet50 is coming from the same distribution as the results of Xception is smaller than 1%. In other words, Xception is significantly better than its closest competitor ResNet50.

Furthermore, most networks show an increase in accuracy between both their base variant and ConvNet and between ConvNet and ConvNet+C2F. Exceptions to this are VGG16, VGG19 and MobileNet. The first two show a slight decrease of accuracy between the base variant and the ConvNet variant. The last one demonstrates a large drop in accuracy when its ConvNet variant is used with C2F.

Also all models show a decrease of average query time when used with C2F as opposed to without. This query time is in direct relation with the size of the final descriptors.

A network is viewed as a suitable basis for ConvNet+C2F when its performance with regard to accuracy an average query time is similar to, or better than ConvNet+C2F based on GoogLeNet. As said Xception demonstrates this behaviour and it is therefore concluded that it is a better basis for ConvNet than GoogLeNet is. In subsequent experiments ConvNet is used with as basis network Xception instead of GoogLeNet. To make this distinction clear it is referred to with the name Xception ConvNet (XConvNet).

Furthermore, the results demonstrate that the ConvNet approach does in general improve the accuracy of the networks in question. It is thus concluded that the discrepancy between our ConvNet+C2F in Section 4.2 and that of Yao et al. due to the way the model has been trained using Caffe. Since the models are now implemented using Keras and trained using Adadelta, without changing the way the various descriptors are created or ranking is performed. Also the fact that Adadelta is able to optimize Xception such that it achieves this performance shows its suitability.

4.6.4 Market-1501 extended test data

The last part of this experiment is dedicated to testing the handling of irrelevant images by the ConvNet+C2F variant that achieves the highest accuracy on the standard Market test set. Due to time constraints only select the best performing model is selected. For this the change in accuracy is taken into account. No training is required for this experiment as the best ConvNet+C2F from Section 4.6.3 is used. This Xception ConvNet is used to extract descriptors for the extended Market-1501 test set after which ranking is performed as usual. Results are listed in Table 4.11.

Table 4.11: XConvNet+C2F on extended Market-1501

	accuracy(%)		avg query time(ms)			descriptor dims	
Model	mAP	Rank-1	coarse	fine	total	coarse	fine
ConvNet+C2F, Yao [79]	46.74	64.58	–	–	180	128	4096
XConvNet+C2F	58.0	79.4	138	14	152	128	8192

It can readily be observed that the best model from Section 4.6.3, Xception ConvNet+C2F with 128 dimensional descriptors, achieves a mAP accuracy of 58.0% and a Rank-1 of 79.4% on the extended dataset. This is an improvement on the mAP of 46.74% and the Rank-1 of 68.9 as reported by Yao et al. [79]. Comparing these results to those on the standard Market-1501 the following can be seen. The mAP score drops with 8.5 percentage points from 66.5% to 58.0% and the Rank-1 of 85.8% drops with 6.4 percentage points to 79.4%.

These results indicate that Xception ConvNet+C2F is also effective when there exists a large amount of distractor images in the gallery. This makes it a suitable method for real world applications, as in such applications it is expected that many such distractor images exist due to the use of automatic pedestrian detectors. However, we note that it is important to reduce the number of distractor images as much as possible, since they have a non-negligible negative influence on the performance of the model.

Finally, we note the difference in timing between the Xception ConvNet+C2F of 152 milliseconds reported here and the one reported for ConvNet+C2F in Section 4.3 of 235 milliseconds. This difference can fully be explained by the fact that comparison optimizations specific to Python, have acquired during the implementation of the Caffe experiment, have been applied to the subsequent Keras implementation used here. These implementations could also be applied to the Caffe implementation, but this would require to reimplement the Caffe based experiment without the prospect of affecting any of the conclusions drawn from the respective experiment.

4.7 Xception ConvNet+C2F on CUHK03

Xception ConvNet+C2F (XConvNet+C2F) proved to be effective for the Market-1501 dataset, as demonstrated in Section 4.6. It is now desirable to know if Xception ConvNet+C2F is truly effective at handling the person re-id problem and if its promising results generalizes to another dataset.

Here it is investigated if Xception ConvNet+C2F also performs on the CUHK03 dataset. This dataset shares some important properties with Market-1501, such as that it is frequently used in literature, uses the DPM detector to automatically detect pedestrians, uses six cameras and has more than one match per identity in the gallery subset. Furthermore, this dataset is more challenging than Market-1501 since it is about three times smaller. Here the new training/testing protocol for this dataset as proposed by Zhong et al. [89] is used. This allows us to test the network in a more realistic setting than the older single-shot setting. This new format is similar to the one used for Market-1501 in that the data is divided into a training set and a test set. The latter is further divided into a query and gallery part. About half of the identities are placed in the train set, the remaining ones are put into the test set. This in contrast with the older approach of selecting only 100 identities for the test set. Note that this new setting makes CUHK03 much harder, especially since there are also less images in the training subset. It is therefore likely that the mAP and Rank-r scores calculated for CUHK03 will be considerably worse than those for Market-1501.

Furthermore, the dataset comes with two different sets of bounding boxes. The "detected" bounding boxes are obtained by using a automatic pedestrian detector [18], the "labelled" boxes are created by human observers.

Xception ConvNet+C2F is trained for 75 epochs using batches of 16 with the Adadelta optimizer.

Table 4.12 compares the performance of Xception ConvNet+C2F with that of the state-of-the-art on CUHK03. All results shown are based on the new data setting. Results for both the detected and labelled bounding boxes are presented.

Table 4.12: Performance of XConvNet+C2F on CUHK03 using the new dataset structure, ordered by mAP on the detected subset.

	detected		labelled	
Method	mAP(%)	Rank-1(%)	mAP(%)	Rank-1(%)
DPFL, Chen [7]	37.0	40.7	40.5	43.0
SVDNet, Sun [61]	37.83	41.50	37.83	40.93
XConvNet+C2F	41.3	41.9	46.8	49.1

It is observed that our method achieves improved performance compared to that of recent state-of-the-art methods which also used the new data format. This is true when the detected bounding boxes are used, but even more so when the labelled bounding boxes are shown to the network.

The fact that Xception ConvNet+C2F also shows good performance when trained on CUHK03 shows that the method generalizes to other datasets and is therefore a viable option for person re-id.

Furthermore, when comparing the mAP and Rank-1 scores of Xception ConvNet+C2F and that of its closest competitor, SVDnet [61], it can be seen that the mAP differs the most for the detected bounding boxes. The mAP increases with 3.17 percentage points, whereas the Rank-1 increases with 0.4 percentage points. This indicates that Xception ConvNet+C2F is especially better able to place correct, but harder, matches higher up in the ranking than SVDnet.

Using the labelled bounding boxes instead of the detected ones lets the accuracy of our model improve even more. The usage of such labelled bounding boxes instead of automatically detected ones lets the method of Chen et al. [7] increase with 3.5% (absolute) and that of Sun et al. [61] with 0%. Whereas our method increases with 5.5% (absolute) from 41.3% to 46.8%. This means that an improvement in the used detector also translates to a greater improvement for a system also using Xceptino ConvNet+C2F instead of using one of the other two person re-id methods.

4.8 Training Xception ConvNet+C2F from scratch

Literature suggests [38] that datasets the size of Market-1501 are large enough for convolutional networks to be trained from scratch, instead of being fine-tuned from an ImageNet pre-trained network. In this section it is investigated if this is true for Xception ConvNet+C2F (XConvNet+C2F).

Xception ConvNet+C2F is trained on Market-1501 both with and without pre-training on ImageNet. This is repeated for five times. Apart from the model pre-training, all settings are the same for both instances: Adadelta is used as optimizer, batches of size 16 and training for 75 epochs.

In Table 4.13 the results are displayed. Average performance and the according standard deviations are obtained by repeating both variants five times.

Table 4.13: Average performance XConvNet+C2F on Market-1501 with and without pre-training on ImageNet

Model	avg mAP(%)	std mAP(%)	avg Rank-1(%)	std Rank-1(%)
XConvNet+C2F from scratch	66.3	0.6	85.5	0.6
XConvNet+C2F pre trained	66.5	0.7	85.8	0.5

It is clear from the results that training Xception ConvNet+C2F with or without ImageNet initialized weights does not show a considerable change in accuracy, as both average mAP's and average Rank-1 scores are within a single standard deviation of the other.

From this it is concluded that while using this initialization is not harmful for training, neither is it beneficial. These observed results can be explained by the following two causes. First, it could be that the data of ImageNet is substantially different from that of Market-1501 such that the features learned from the former do not aid detection for the latter. However, literature [75] also indicates that the use of ImageNet can be useful when training on Market-1501. We therefore draw the conclusion that training with Adadelta improves the training such that the aid of the ImageNet learned features is not necessarily needed. It still might be the case that training with pre-initialized weights might converge faster than training without.

4.9 Batch size influence

Experimentation indicated that the batch size is of influence on the ConvNet+C2F accuracy, in this experiment the validity of this observed result is investigated.

Xception ConvNet+C2F is trained on Market-1501 and is initialized using ImageNet pre-trained weights. Training is done with a batch size of 16 and 8, GPU memory constrains us from using a larger batch size. This is repeated for five times. Apart from the model initialization and batch size, all settings are the same for both instances: Adadelta is used as optimizer and training is performed for 75 epochs.

In Table 4.14 the results are displayed. Average performance and the according standard deviation are obtained by repeating both variants five times.

Table 4.14: Batch size influence on the performance of XConvNet+C2F on Market-1501

Model	batch-size	avg-mAP(%)	std-mAP(%)	avg-Rank-1(%)	std-Rank-1(%)
XConvNet+C2F	8	62.8	0.9	83.1	0.7
XConvNet+C2F	16	66.5	0.7	85.8	0.5

The results show a clear gap in both the mAP and Rank-1 accuracy when training Xception ConvNet with 8 or 16 images in a single batch. The mAP score when using a batch size of 16 is 66.5% while it is 62.8% with a batch size of 8.

A difference in performance when using differing batch sizes is not necessarily expected, but also not surprising. Various training elements, such as the batch normalization layers or the Adadelta optimizer calculate their values on a per batch basis. Therefore adjusting the batch size can influence results, as is seen here.

Furthermore, these results prompt the question what happens when the batch sizes are increased even further. However, using our currently available system setup, specified in Section 4.1.4, in combination with the large image input dimensions makes that the GPU runs out of memory when attempting this experiment. A way around this would be to use smaller images, but this probably will come at the cost of some accuracy, as mentioned in the work of Yao et al. [79]. The question is then if the increase in accuracy gained by the larger batch size outweighs the loss in accuracy by using smaller images and thus smaller descriptors. This could be an interesting point for future research.

4.10 Coarse descriptor dimensionality reduction

The goal for the experiments in this section is to take a look at the way the coarse descriptor is created for the coarse-to-fine search framework. The approach taken consists out of three steps. First, in Subsection 4.10.1 it is investigated whether using PCA to reduce the coarse descriptor to a different number of dimensions than 128 is beneficial. Second, in Subsection 4.10.2 a look is taken at how other reduction methods perform compared to PCA. Third, a separate look is taken at how t-SNE can be used to perform the needed reduction in dimensions in Subsection 4.10.3.

4.10.1 Varying dimensions with PCA

Is the reduction of the coarse descriptor to 128 dimensions optimal? Yao et al. [79] show that reducing the descriptors to this is successful but do not provide conclusive evidence that it is optimal. Second, if the number of dimensions can be reduced to two or three and still have a reasonable performance, it might be feasible to use t-SNE to perform te reduction, as posed in Section 2.5.

The Xception ConvNet+C2F trained in Section 4.6 is used to extract the descriptors for the standard Market-1501 and CUHK03 test sets. This to ensure that the dimensionality is not over-fitted on a single data set. The PCA is applied to reduce the dimensionality of the coarse descriptor. Results are thus based on using the Xception ConvNet in combination with the coarse-to-fine search framework. They are presented in Table 4.15 and in Table 4.16.

Table 4.15: Effect of coarse PCA descriptor dimensionality for XConvNet+C2F on Market-1501

coarse descriptor dim	mAP(%)	Rank-1(%)
256	67.7	86.6
128	67.7	86.6
64	67.8	86.6
32	68.3	86.6
16	69.7	86.5
8	72.4	85.7
4	72.2	81.6
3	69.2	75.4
2	61.8	64.6

Table 4.16: Effect of coarse PCA descriptor dimensionality for XConvNet+C2F on CUHK03-detected

coarse descriptor dim	mAP(%)	Rank-1(%)
256	41.3	41.9
128	41.3	41.9
64	41.3	41.9
32	41.5	41.9
16	42.0	41.8
8	42.6	41.9
4	43.0	41.6
2	40.9	38.4

The results show initially an increasing trend in the mAP score when the coarse dimensionality is reduced. For Market-1501 this trend sets in starting from 64 dimensions until 8 dimensions are used. The same trend is visible for CUHK03 but it sets in starting from 32 dimensions to 4 dimensions. In both cases a large drop in the mAP score is observed when going from 4 to 2 dimensions. Furthermore, the Rank-1 score remains stable for most iterations and starts to drop when the dimensionality is sufficiently getting reduced. For Market-1501 this starts at a dimensionality of 8 and for CUHK03 this starts at 4 dimensions.

From the results it is clear that using 64 or 254 dimensions makes no real difference with using 128 dimensions. This suggest that using 128 dimensions is an overestimate of the number that is actually needed to achieve good performance.

The upward trend in the mAP when reducing the dimensionality indicates that the system profits from the use of less dimensions. This can be explained by the fact that PCA reduces dimensions containing the least amount of variation, or information, combined with the fact that an unweighed Euclidean distance is used to match the descriptors. This means that noisy values containing little information get weighted in an equal manner as values containing useful information. By removing some of this noise PCA makes that the model performance increases. This notion is strengthened by the observation that the Rank-1 score does not seem to increase. This indicates that especially difficult cases for which it is vital that only important information is taken into account see an improvement.

Furthermore, the observation that the increase in mAP sets in when a smaller number of dimensions is reached for CUHK03 than for Market-1501 can be explained by the size of the two datasets. Market-1501 is about three times as large as CUHK03 and therefore more descriptive descriptors are needed to do the ranking. This idea is confirmed by the fact that for CUHK03 the reduction in performance sets in when 2 dimensions are used, whereas the same thing can be seen to happen for Market-1501 at 4 dimensions.

4.10.2 Other descriptor-reduction methods

How do other reduction methods compare to PCA for reducing the coarse descriptor in the coarse-to-fine search frame work? While PCA is proven to work well with the ConvNet approach by Yao et al., it is not the only available descriptor reducer. Variations on PCA and even entirely different approaches might also work well as long as they adhere to the following requirements. They must be able to defer most of the calculations to the offline processing step and close points before the reduction must be close after it. For example, PCA can calculate the desired new system coordinates in the offline step. Then in the online step it can place new points in this new space without having to look at the entire dataset again.

In this experiment a look is taken at several decomposition techniques which share this property. They are: kernel PCA [47], sparse PCA [12], Dictionary learning [46], factor analysis [32], latent dirichlet allocation (LDA) [27], NMF [11], truncated SVD [23], fast ICA [30] and also t-SNE [45]. These methods are also introduced in Section 2.5.

As in Subsection 4.10.1 the Xception ConvNet+C2F trained in Section 4.6 is used to extract the descriptors for the standard Market-1501. A look is taken at the most interesting points indicated by the results in Table 4.15 and in Table 4.16. These are at 32 dimensions when the increase in the mAP score sets in and at 8 dimensions when it is at its peak. Results are displayed in Table 4.17

Table 4.17: Performance of XConvNet+C2F with various reduction methods on Market-1501

	8 dims		32 dims		128 dims	
Method	mAP(%)	Rank-1(%)	mAP(%)	Rank-1(%)	mAP(%)	Rank-1(%)
kernel PCA	72.3	85.6	68.3	86.6	67.7	86.6
factor analysis	72.4	85.7	68.5	86.6	68.4	86.6
fast ICA	72.4	85.7	68.5	86.6	68.4	86.6
PCA	72.4	85.7	68.3	86.6	67.7	86.6
truncated SVD	72.9	85.7	68.1	86.6	67.6	86.6
sparse PCA	73.1	85.7	68.3	86.6	68.0	86.6
dictionary learning	73.1	85.6	68.6	86.6	68.1	86.6
LDA	73.5	85.5	70.2	86.4	70.2	86.4
NMF	73.5	85.2	69.6	86.4	69.3	86.6

For 128 dimensions the Rank-1 performance does not show a lot of variation for all of the different reduction methods. It is 86.6% for all methods except for LDA, for which it is 86.4%. The mAP score does show a larger amount of variation, the lowest score being 67.6% for truncated SVD and the highest being 70.2% for LDA. This indicates that the most difference in performance is caused by the harder instances. Exemplary for this is LDA, although it has the lowest rank-1 score it has the highest mAP, meaning that the loss in accuracy on easier instances is made up for by a better ranking of harder instances.

When using 32 dimensions a small upward trend can be observed for the mAP scores. The lowest mAP is now 68.1% for truncated SVD and the highest is still 70.2 for LDA. This trend indicates that the reduction of dimensions is beneficial for the produced rankings.

This trend is continued when the number of dimensions gets reduced to only 8. The lowest mAP score is then 72.3% for kernel PCA and the highest is for NMF and LDA with 73.5%. However, given the standard deviation of the mAP of 0.7, presented in Subsection 4.6.3. This means that the lowest mAP of 72.3% is within two standard deviations of the largest one of 73.5. We deem this not considerable enough to recommend one of these methods over another and therefore leave it by using PCA.

4.10.3 t-SNE

How can t-SNE be used, both for dimension reduction and visualization? As mentioned in Subsection 4.10.2 t-SNE can be used to reduce the dimensionalities of the coarse descriptor to two or three dimensions. Albeit slower than the alternative methods from Subsection 4.10.2 it is expected to be more precise since it takes into account the relative positions of the descriptors both before and after the reduction [45]. Furthermore, t-SNE is known to provide suitable visualizations for inspecting the data from which we might be able to infer something on how the network decides to construct the descriptors.

The same experimental setup is used as in Subsection 4.10.2, but due to the limitations of t-SNE the reduction can only be performed to two and three dimensions. Furthermore, as a comparison results for PCA with two or three dimensions are also provided. On top of that the descriptors with two dimensions are used to plot visualizations of the data. In Table 4.17 the results are listed for this experiment.

Table 4.18: Effect of coarse t-SNE descriptor size for Xception ConvNet+C2F on Market-1501

Reduction method	coarse dim	mAP(%)	Rank-1(%)
t-SNE	3	74.9	85.2
t-SNE	2	73.4	85.6
PCA	3	69.2	75.4
PCA	2	61.8	64.6

In both cases it is observed that the performance of t-SNE is superior to that of PCA. For two dimensions PCA obtains a mAP of 61.8% and a Rank-1 of 64.6%, compared to t-SNE with 73.4% and 85.6% respectively. When reducing to three dimensions this gap is less profound with a mAP of 69.2% and a Rank-1 of 75.4% for PCA and 74.9% and 85.2% for t-SNE.

As the top performing t-SNE mAP of 74.9% for three dimensions is higher than the best result for PCA with a mAP of 72.4% for eight dimensions it might be used in cases where query time is of less importance but the overall retrieval success is. However, the top Rank-1 score for PCA is 86.6% for 32 dimensions or more. This is is better than the top Rank-1 score for t-SNE, which is 85.6% for two dimensions, as is displayed in Table 4.15. This would indicate that t-SNE is especially better retrieving harder instances but PCA is preferred for easier ones. This, combined with the fact that for PCA most of the computation time can be done before actually performing a query, makes that t-SNE is not suitable for replacing it. Only in certain very specific cases could t-SNE be better than PCA. This becomes especially clear in the next section where this is shown for the extended Market-1501 dataset.

4.10.4 Market-1501 extended test data

In the Subsection 4.10.1 and Subsection 4.10.3 it is demonstrated that Xception ConvNet+C2F benefits from using smaller coarse descriptors. This decreases the required average query time and increases the model's accuracy. In this experiment, it is tested if these properties also hold when the model is tested on the larger extended Market-1501 test data.

The setup of this experiment is similar to the setup in Section 4.6.4. Descriptors for the extended Market-1501 datasets are extracted using the best Xception ConvNet from Section 4.6.3. This model is trained for 75 epochs on Market-1501, using the Adadelta optimizer. Coarse descriptors with a dimensionality of 128, 32 and 8 are constructed using PCA. This is similar to the experiment in Subsection 4.10.1. Additionally, to demonstrate the fact that t-SNE can be used as a slower, but more accurate, alternative to PCA, results are shown for 3 dimensional coarse descriptors created using t-SNE. Results are presented in Table 4.19.

Before the results of this experiment are discussed, we want to stress an important difference between the use of PCA and t-SNE as a descriptor-reduction method. In order to perform their reductions, both methods need to find a transformation from the original descriptor space to the new and smaller coarse descriptor space, as is also explained in Section 3. The time this takes greatly varies for these two methods. To perform this calculation PCA needs about 4 minutes for the approximately 520,000 descriptors in the extended dataset. Compare this to the 120 hours t-SNE needs to calculate this transformation. This time difference is already large, but it becomes even larger when new images are added to the database. In case of PCA, the method can readily

Table 4.19: XConvNet+C2F on extended Market-1501. Note that for creating the coarse descriptors PCA needs a one-time investment of 4 minutes where t-SNE needs 120 hours every time new images are added to the database.

	descriptors			accuracy(%)		avg query time(ms)		
Model	method	coarse dims	fine dims	mAP	Rank-1	coarse	fine	total
XConvNet+C2F	PCA	128	8192	58.0	79.4	138	14	152
XConvNet+C2F	PCA	32	8192	61.7	79.2	88	14	102
XConvNet+C2F	PCA	8	8192	63.8	67.9	67	14	81
XConvNet+C2F	t-SNE	3	8192	68.5	78.9	63	14	77

use the previously calculated transformation to transform the descriptor of this new image to the reduced coarse dimensionality space. However, due to the way t-SNE is designed, it is not able to do this and thus needs to repeat the entire transformation calculation process. In practice, this means that for PCA a one-time investment of about 4 minutes is needed, after which new images can be processed in a matter of milliseconds. While for t-SNE, every new set of images that are added to the database require at least an additional 120 hours of computation.

With these important differences noted, we observe the following trend. When reducing the dimensionality of the coarse descriptors, the mAP increases while the Rank-1 scores decrease. The following absolute changes in accuracy are observed when taking the the results for the 128 dimensional coarse descriptors as a baseline. For the 32 dimensional descriptors an increase is observed of 3.7% for the mAP and a decrease of 0.2% for the Rank-1. The 8 dimensional descriptors shows an absolute mAP increase of 5.8% and an absolute decrease in the Rank-1 of 9.5%. Furthermore, using t-SNE to reduce the coarse descriptor size to 3 dimensions increases the mAP with 10.5% (absolute) and decreases the Rank-1 with 0.5% (absolute).

Besides influencing the accuracy the coarse descriptor dimensionality also influences the measured average query times. These times do not include the time needed to construct the respective descriptors, but reflect the time that is needed to rank the gallery w.r.t. a query images as soon as all descriptors are already created. This means that these times are only dependent on the amount of dimensions and independent from the used descriptor-reduction method. As expected, smaller descriptors make for faster query times. Again taking the 128 dimensional descriptors as a baseline, the following is observed. Using 32 dimensional descriptors the system is 50 milliseconds faster on average, which amounts to a speedup of 149%. Then 8 dimensional descriptors make queries 71 milliseconds faster on average, a speedup of 188%. Finally, using the 3 dimensional descriptors decreases the average query time by 75 milliseconds, thereby speeding up the process with 197%.

The results indicate that Xception ConvNet+C2F is better able to handle a gallery containing large amounts of false positive image than the ConvNet+C2F of Yao et al [79]. Using PCA to construct 8 dimensional coarse descriptors, Xception ConvNet+C2F improves the original ConvNet+C2F's mAP of 46.74% with 17.1% and its Rank-1 of 64.58% with and 3.3%, both absolute.

However, it is important to point out that the additional distractor images have a negative influence on the obtained accuracy. Comparing the results of the 8 dimensional coarse descriptors to those on the standard Market-1501 dataset in Table 4.10, the following can be seen. The mAP drops with 8.6% and the Rank-1 drops with 17.8%, both absolute. This shows that it is of importance to keep the gallery small and clean.

Furthermore, the fact that the mAP increases indicates that the system's overall performance improves when the dimensionality of the coarse descriptors is reduced. However, this comes at the cost of the quality of the top results, as the Rank-1 shows a decrease. If only the top results are of importance for a certain use case, then it might be better to use descriptors with a size of 32 instead of 8. However, if the overall accuracy of the system has priority, then 8 dimensional descriptors are preferred. Finally, if the average query time is of no importance, but the accuracy is, then t-SNE is a better alternative then PCA.

Chapter 5

Conclusion, discussion and future work

5.1 Conclusion

To answer the central research question: can we obtain state-of-the-art accuracy and high speed performance? We proposed the new Xception ConvNet which is shown to work well with the coarse-to-fine (C2F) search framework. It uses the capabilities of Xception, a network not yet used for person re-identification.

Additionally the training of Xception ConvNet is recommended to be performed using the Adadelta optimizer, as it has been shown that this optimizer outperforms other modern alternatives. This alleviates a user from having to manage the learning rate during training, as this optimizer is able to determine this by itself.

Additionally, as smaller sizes reduce the model's performance we recommend to perform the training of Xception ConvNet using batch sizes of at least 16, .

Furthermore, we demonstrated that the coarse-to-fine search framework can be improved in both speed and accuracy by using a coarse descriptor dimensionality of 8 instead of 128. This answers the question whether the coarse-to-fine search framework benefits from different sized coarse descriptors.

With our recommendations the Xception ConvNet+C2F achieves state-of-the-art results on Market-1501 and CUHK03. Furthermore, results on the extended Market-1501 dataset containing a large number of distractor images demonstrates that our approach is suitable for use with large galleries that are common for a real world applications.

Finally, we also explored the usage of DCGAN generated images during training, but conclude that this is not beneficial for Xception ConvNet+C2F. This is based on the observation that the improvement in accuracy using generated images could not be shown for other networks than ResNet50. We conclude that the newer GoogLeNet and Xception models have better generalisation capabilities than ResNet50, therefore making the generalisation offered by the DCGAN images not needed.

5.2 Discussion

To repeat the results reported by Yao et al. [79] an implementation of ConvNet+C2F was realised in Caffe. Results produced using this implementation differed from those in the literature. Thereafter we showed that ConvNet+C2F works with a new implementation in Keras. To also perform the original ConvNet experiment using GoogLeNet in this Keras implementation would have been a good addition. But due to the lack of an existing GoogLeNet in Keras we resorted to InceptionV3, the successor of GoogLeNet which is available in Keras.

Furthermore, using the Caffe implementation we showed that the DCGAN generated images did not improve training for either GoogLeNet or ConvNet based on GoogLeNet. Given validity of these results we considered it a low priority to perform this experiment using another network in the Keras implementation, but we do feel that it would make the case stronger.

In the experiments in this work we have focussed on increasing the overall performance of ConvNet by prioritizing the mAP score over the Rank-1 score. We view this as a valid method as the mAP score takes all possible matches into account, whereas the Rank-1 score only takes looks at the first position in the retrieved list. However, there are use cases where one is mainly interested in the first, or first few, result(s). In these cases it would be better to give a higher priority to the Rank-1 score than the mAP score. This is especially important for the experiments where an increase in the mAP corresponds with a decrease in the Rank-1. An example of this is the performance of Xception ConvNet on the extended Market-1501 dataset. A coarse descriptor dimensionality of 8 instead of 32 did improve the mAP score, but reduced the Rank-1 score.

5.3 Future work

Results showed that the proposed method shows state-of-the-art results on Market-1501 with the single query setting and it is expected that the same holds for when the multi query setting is used. It would be interesting to put this assumption to the test in future work.

Furthermore, we demonstrated that the use of DCGAN generated images was not beneficial for ConvNet. However, a recent paper of Yunus et al. [55] claims to produce better images than generated by the DCGAN we used, which might be of interest when further researching this approach.

It was demonstrated that batch sizes of 16 demonstrated better performance than those of 8. However, memory restrictions did not permit us to experiment with larger batch sizes. In the future it would be interesting to find ways to enable this experimentation. This could be done by utilizing multiple GPUs at the same time. Another option might be using smaller images as input, but this might complicate the construction of the fine descriptor. As lower dimensional input images means lower dimensional output for ConvNet. The fine descriptor as it is defined now requires the final output of the ConvNet to be evenly divisible by 4.

The experiments showed that the accuracy of the network improved when the dimensionality of the coarse descriptors is reduced, it would be interesting to see if the same holds for the fine descriptors of C2F. Perhaps it would be even feasible to transform the now two-step process into a single-step one. Additionally, the limited time needed to do the re-ranking with the larger fine descriptors, suggests that it is feasible to use more than 500 images in the fine step and thereby potentially increasing the mAP score.

Another interesting topic for future research is to investigate how the proposed Xception ConvNet+C2F can be trained such that it show good performance on smaller datasets. A viable option might be the use of a Siamese network approach during training as done by Geng et al. [20]. They successfully use such an approach to train a state-of-the-art network on the VIPeR [21] dataset, one of the smallest datasets in literature for person re-identification.

Acknowledgement

Foremost, I would like to express my gratitude to both of my thesis supervisors, Fons Verbeek and Henri Bouma, for their valuable guidance and detailed remarks. Your advise definitely helped me to choose the right direction and successfully complete my thesis. Especially the positive and constructive way in which you provided me with feedback has made working on this thesis an enjoyable experience.

Furthermore, I would like to thank the TNO's Intelligent Imaging group for providing me with the opportunity to perform this research. I have found the discussions with the fellow deep learners and interns particularly helpful and interesting.

Bibliography

[1] M. Abadi, A. Agarwal, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/, 2015.

[2] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *IEEE CVPR*, pages 3908–3916, 2015.

[3] S. Bak and P. Carr. One-shot metric learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[4] I. B. Barbosa, M. Cristani, B. Caputo, A. Rognhaugen, and T. Theoharis. Looking beyond appearances: Synthetic training data for deep CNNs in re-identification. *arXiv:1701.03153*, 2017.

[5] S.-Z. Chen, C.-C. Guo, and J.-H. Lai. Deep ranking for person re-identification via joint representation learning. *IEEE Transactions on Image Processing*, 25(5):2353–2367, 2016.

[6] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. *arXiv:1704.01719*, 2017.

[7] Y.-C. Chen, X. Zhu, W.-S. Zheng, and J.-H. Lai. Person re-identification by camera correlation aware feature augmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[8] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng. Person re-identification by multi-channel parts-based CNN with improved triplet loss function. In *IEEE CVPR*, pages 1335–1344, 2016.

[9] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[10] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2017.

[11] A. Cichocki and A.-H. Phan. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 92(3):708–721, 2009.

[12] A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. Lanckriet. A direct formulation for sparse PCA using semi–definite programming. In *Advances in neural information processing systems*, pages 41–48, 2005.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*, pages 248–255. IEEE, 2009.

[14] S. Ding, L. Lin, G. Wang, and H. Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 48(10):2993–3003, 2015.

[15] T. Dozat. Incorporating Nesterov momentum into Adam. *https://openreview.net*, 2016.

[16] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. 12(Jul):2121–2159, 2011.

[17] H. Fan, L. Zheng, and Y. Yang. Unsupervised person re-identification: clustering and fine-tuning. *arXiv:1705.10444*, 2017.

[18] 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*, 32(9), 2010.

[19] A. Franco and L. Oliveira. Convolutional covariance features: Conception, integration and performance in person re-identification. *Pattern Recognition*, 61:593–609, 2017.

[20] M. Geng, Y. Wang, T. Xiang, and Y. Tian. Deep transfer learning for person re-identification. *arXiv:1611.05244*, 2016.

[21] D. Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, reacquisition, and tracking. Citeseer, 2007.

[22] Y.-F. Guo, L. Wu, H. Lu, Z. Feng, and X. Xue. Null Foley–Sammon transform. *Pattern Recognition*, 39(11):2248–2251, 2006.

[23] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE CCVP*, 2016.

[25] 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*, pages 770–778, 2016.

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

[27] M. Hoffman, F. R. Bach, and D. M. Blei. Online learning for latent Dirichlet allocation. In *advances in neural information processing systems*, pages 856–864, 2010.

[28] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[29] 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. abs/1704.04861, 2017.

[30] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.

[31] H. Jin, X. Wang, S. Liao, and S. Z. Li. Deep person re-identification with improved embedding. *arXiv:1705.03332*, 2017.

[32] I. T. Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. 1986.

[33] T. Kim. DCGAN-tensorflow: A tensorflow implementation of deep convolutional generative adversarial networks. https://github.com/carpedm20/DCGAN-tensorflow. (Accessed on 05/22/2017).

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

[35] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *IEEE CVPR*, pages 2288–2295. IEEE, 2012.

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

[37] S. Li and H. Ma. A siamese inception architecture network for person re-identification. *Machine Vision and Applications*, 2017.

[38] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *IEEE CVPR*, pages 152–159, 2014.

[39] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *IEEE CVPR*, pages 152–159, 2014.

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

[41] W. Lin, Y. Shen, J. Yan, M. Xu, J. Wu, J. Wang, and K. Lu. Learning correspondence structures for person re-identification. *IEEE Transactions on Image Processing*, 26(5):2438–2453, 2017.

[42] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, and Y. Yang. Improving person re-identification by attribute and identity learning. *arXiv:1703.07220*, 2017.

[43] H. Liu, Z. Jie, K. Jayashree, M. Qi, J. Jiang, S. Yan, and J. Feng. Video-based person re-identification with accumulative motion context. *arXiv:1701.00193*, 2017.

[44] X. Ma, X. Zhu, S. Gong, X. Xie, J. Hu, K.-M. Lam, and Y. Zhong. Person re-identification by unsupervised video matching. *Pattern Recognition*, 65:197–210, 2017.

[45] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[46] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009.

[47] S. Mika, B. Schölkopf, A. J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. In *Advances in neural information processing systems*, pages 536–542, 1999.

[48] S. H. Oh, S.-W. Han, B.-S. Choi, and G.-W. Kim. Prototype system design for large-scale person re-identification. In *Advanced Multimedia and Ubiquitous Engineering*, pages 649–651. 2017.

[49] S. Paisitkriangkrai, L. Wu, C. Shen, and A. van den Hengel. Structured learning of metric ensembles with application to person re-identification. *Computer Vision and Image Understanding*, 156:51–65, 2017.

[50] M. Qi, J. Han, J. Jiang, and H. Liu. Deep feature representation and multiple metric ensembles for person re-identification in security surveillance system. *Multimedia Tools and Applications*, 2017.

[51] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, 2015.

[52] H. Robbins and S. Monro. A stochastic approximation method. pages 400–407, 1951.

[53] S. Ruder. An overview of gradient descent optimization algorithms. abs/1609.04747, 2016.

[54] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[55] Y. Saatchi and A. G. Wilson. Bayesian GAN. https://arxiv.org/abs/1705.09558, 2017. (Accessed on 05/30/2017).

[56] A. Schumann, S. Gong, and T. Schuchert. Deep learning prototype domains for person re-identification. *IEEE ICIP*, 2016.

[57] E. Shelhamer. BVLC GoogLeNet implementation of GoogLeNet in Caffe. https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet. (Accessed on 08/24/2017).

[58] E. Shelhamer. Caffe. http://caffe.berkeleyvision.org/. (Accessed on 08/24/2017).

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

[60] M. Sugiyama. Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. *Journal of machine learning research*, 8:1027–1061, 2007.

[61] Y. Sun, L. Zheng, W. Deng, and S. Wang. SVDNet for pedestrian retrieval. *arXiv:1703.05693*, 2017.

[62] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4, inception-resnet.

[63] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE CVPR*, 2015.

[64] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *IEEE CVPR*, pages 2818–2826, 2016.

[65] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[66] T. Tieleman and G. Hinton. RMSprop Gradient Optimization. 2015.

[67] R. R. Varior, M. Haloi, and G. Wang. Gated siamese convolutional neural network architecture for human re-identification. In *ECCV*, pages 791–808. Springer, 2016.

[68] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang. A Siamese Long Short-Term Memory architecture for human re-identification. In *ECCV*, pages 135–153. Springer, 2016.

[69] 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 *IEEE CVPR*, pages 1288–1296, 2016.

[70] L. Wu, C. Shen, and A. v. d. Hengel. Deep recurrent convolutional networks for video-based person re-identification: An end-to-end approach. *arXiv:1606.01609*, 2016.

[71] L. Wu, C. Shen, and A. v. d. Hengel. PersonNet: person re-identification with deep convolutional neural networks. *arXiv:1601.07255*, 2016.

[72] L. Wu, C. Shen, and A. van den Hengel. Deep linear discriminant analysis on Fisher networks: A hybrid architecture for person re-identification. *Pattern Recognition*, 65:238–250, 2017.

[73] L. Wu and Y. Wang. Structured deep hashing with convolutional neural networks for fast person re-identification. *arXiv:1702.04179*, 2017.

[74] Q. Xiao, K. Cao, H. Chen, F. Peng, and C. Zhang. Cross domain knowledge transfer for person re-identification. *arXiv:1611.06026*, 2016.

[75] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *IEEE CVPR*, pages 1249–1258, 2016.

[76] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang. End-to-end deep learning for person search. *arXiv:1604.01850*, 2016.

[77] Y. Yan, B. Ni, Z. Song, C. Ma, Y. Yan, and X. Yang. Person re-identification via recurrent feature aggregation. In *ECCV*, pages 701–716. Springer, 2016.

[78] X. Yang, M. Wang, R. Hong, Q. Tian, and Y. Rui. Enhancing person re-identification in a self-trained subspace. *arXiv:1704.06020*, 2017.

[79] H. Yao, S. Zhang, D. Zhang, Y. Zhang, J. Li, Y. Wang, and Q. Tian. Large-scale person re-identification as retrieval. *ICME*, 2017.

[80] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Deep metric learning for person re-identification. In *IEEE ICPR*, pages 34–39. IEEE, 2014.

[81] M. D. Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[82] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *IEEE CVPR*, pages 1239–1248, 2016.

[83] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE Transactions on Image Processing*, 24(12):4766–4779, 2015.

[84] W. Zhang, S. Hu, and K. Liu. Learning compact appearance representation for video-based person re-identification. *arXiv:1702.06294*, 2017.

[85] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *IEEE ICCV*, pages 1116–1124, 2015.

[86] L. Zheng, Y. Yang, and A. G. Hauptmann. Person re-identification: past, present and future. *arXiv:1610.02984*, 2016.

[87] Z. Zheng, L. Zheng, and Y. Yang. A discriminatively learned CNN embedding for person re-identification. *TOMM*, 2017.

[88] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. *ICCV 2017*, 2017.

[89] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. *arXiv:1701.08398*, 2017.

[90] F. Zhu, X. Kong, L. Zheng, H. Fu, and Q. Tian. Part-based deep hashing for large-scale person re-identification. *IEEE Transactions on Image Processing*, 2017.

[91] J. Zhu, H. Zeng, S. Liao, Z. Lei, C. Cai, and L. Zheng. Deep hybrid similarity learning for person re-identification. *arXiv:1702.04858*, 2017.