Deep Cosine Metric Learning for Person Re-Identification

Nicolai Wojke
German Aerospace Center (DLR)
nicolai.wojke@dlr.de

Alex Bewley
University of Oxford
bewley@robots.ox.ac.uk

Abstract

Metric learning aims to construct an embedding where two extracted features corresponding to the same identity are likely to be closer than features from different identities. This paper presents a method for learning such a feature space where the cosine similarity is effectively optimized through a simple re-parametrization of the conventional softmax classification regime. At test time, the final classification layer can be stripped from the network to facilitate nearest neighbor queries on unseen individuals using the cosine similarity metric. This approach presents a simple alternative to direct metric learning objectives such as siamese networks that have required sophisticated pair or triplet sampling strategies in the past. The method is evaluated on two large-scale pedestrian re-identification datasets where competitive results are achieved overall. In particular, we achieve better generalization on the test set compared to a network trained with triplet loss.

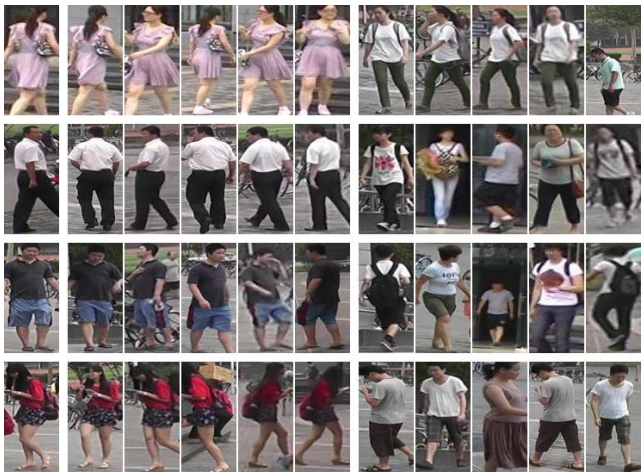


Figure 1: The proposed classifier successfully learns a metric representation space that is robust to articulation, lighting, and background variation. For each query image the five most similar and dissimilar images are shown.

1. Introduction

Person re-identification is a common task in video surveillance where a given query image is used to search a large gallery of images potentially containing the same person. As gallery images are usually taken from different cameras at different points in time, the system must deal with pose variations, different lighting conditions, and changing background. Furthermore, direct identity classification is prohibited in this scenario because individuals in the gallery collected at test time are not contained in the training set. Instead, the re-identification problem is usually addressed within a metric learning framework. Here the goal is to learn a feature representation – from a set of separate training identities – suitable for performing nearest neighbor queries on images and identities provided at test time. Ideally, the learnt feature representation should be invariant to the aforementioned nuisance conditions while at the same time follow a predefined metric where feature similarity corresponds to person identity.

Due to the annotation effort that is necessary to set up a person re-identification dataset, until recently only a limited amount of labeled images were available. This has changed with publication of the Market 1501 [36] and MARS [35] datasets. MARS contains over one million images that have been annotated in a semi-supervised fashion. The data has been generated using a multi-target tracker that extracts short, reliable trajectory fragments that were subsequently annotated to consistent object trajectories. This annotation procedure not only leads to larger amount of data, but also puts the dataset closer to real-world applications where people are more likely extracted by application of a person detector rather than manual cropping.

Much like in other vision tasks, deep learning has become the predominant paradigm to person re-identification since the advent of larger datasets. Yet, the problem remains challenging and far from solved. In particular, there is an ongoing discourse over the performance of direct metric learning objectives compared to approaching the training procedure indirectly in a classification framework. Whereas metric learning objectives encode the similarity metric directly into the training objective, classification-based meth-

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00087

ods train a classifier on the set of identities in the training set and then use the underlying feature representation of the network to perform nearest neighbor queries at test time. On the one hand, in the past direct metric learning objectives have suffered from undesirable properties that can hinder optimization, such as non-smoothness or missing contextual information about the neighborhood structure [19]. On the other hand, these problems have been approached with success in more recent publications [18, 8]. Nevertheless, with similarity defined solely based on class membership, it remains arguable if direct metric learning has a clear advantage over training in a classification regime. In this setting, metric learning is often reduced to minimizing the distance between samples of the same class and forcing a margin between samples of different classes [3, 8]. A classifier that is set up with care might decrease intra-class variance and increase inter-class variance in a similar way to direct metric learning objectives.

Inspired by this discussion, the main contribution of this paper is the unification of metric learning and classification. More specifically, we present a careful but simple re-parametrization of the softmax classifier that encodes the metric learning objective directly into the classification task. Finally, we demonstrate how our proposed cosine softmax training extends the effectiveness of the learnt embedding to unseen identities at test time within the context of person re-identification. Source code of this method is provided in a GitHub repository[1].

2. Related Work

Metric Learning Convolutional neural networks (CNNs) have shown impressive performance on large scale computer vision problems and the representation space underlying these models can be successfully transferred to tasks that are different from the original training objective [5, 22]. Therefore, in classification applications with few training examples a task-specific classifier is often trained on top of a general purpose feature representation that was learned beforehand on ImageNet [11] or MS COCO [16]. There is no guarantee that the representation of a network which has been trained with a softmax classifier can directly be used in an image retrieval task such as person re-identification, because the representation does not necessarily follow a certain (known) metric to be used for nearest-neighbor queries. Nevertheless, several successful applications in face verification and person re-identification exist [24, 31, 37]. In this case, a softmax classifier is trained to discriminate the identities in the training set. When training is finished, the classifier is stripped of the network and distance queries are made using cosine similarity or Euclidean distance on the final layer of the network. If, however, the feature representation cannot be used directly, an alternative is to find a metric subspace in a post processing step [10, 15].

Deep metric learning approaches encode notion of similarity directly into the training objective. The most prominent formulations are siamese networks with contrastive [3] and triplet [28] loss. The contrastive loss minimizes the distance between samples of the same class and forces a margin between samples of different classes. Effectively, this loss pushes all samples of the same class towards a single point in representation space and penalizes overlap between different classes. The triplet loss relaxes the contrastive formulation to allow samples to move more freely as long as the margin is kept. Given an anchor point, a point of the same class, and a point of a different class, the triplet loss forces the distance to the point of the same class to be smaller than the distance to the point of the different class plus a margin.

Both the contrastive and triplet losses have been applied successfully to metric learning problems (e.g., [21, 26, 8]), but the success has long been dependent on an intelligent pair/triplet sampling strategy. Many of the possible choices of pairs and triplets that one can generate from a given dataset contain little information about the relevant structures by which identities can be discriminated. If the wrong amount of hard to distinguish pairs/triplets are incorporated into each batch, the optimizer either fails to learn anything meaningful or does not converge at all. Development of an effective sampling strategy can be a complex and time consuming task, thus limiting the practical applicability of siamese networks.

A second issue related to the contrastive and triplet loss stems from the hard margin that is enforced between samples of different classes. The hard margin leads to a non-smooth objective function that is harder to optimize, because only few examples are presented to the optimizer at each iteration and there can be strong disagreement between different batches [19]. These problems have been addressed recently. For example, Song *et al*. [18] formulate a smooth upper bound of the original triplet loss formulation that can be implemented by drawing informative samples from each batch directly on a GPU. A similar formulation of the triplet loss where the hard margin is replaced by a soft margin has shown to perform well on a person re-identification problem [8].

Apart from siamese network formulations, the magnet loss [19] has been formulated as an alternative to overcoming many of the related issues. The loss is formulated as a negative log-likelihood ratio between the correct class and all other classes, but also forces a margin between samples of different classes. By operating on entire class distributions instead of individual pairs or triplets, the magnet loss potentially converges faster and leads to overall better solutions. The center loss [29] has been developed in an attempt

[1] `github.com/nwojke/cosine_metric_learning`

to combine classification and metric learning. The formulation utilizes a combination of a softmax classifier with an additional term that forces compact classes by penalizing the distance of samples to their class mean. A scalar hyperparameter balances the two losses. Experiments suggest that this joint formulation of classification and metric learning produces state of the art results.

Person Re-Identification With the availability of larger datasets, person re-identification has become an application domain of deep metric learning and several CNN architectures have been designed specifically for this task. Most of them focus on mid-level features and try to deal with pose variations and viewpoint changes explicitly by introducing special units into the architecture. For example, Li *et al*. [13] propose a CNN with a special patch matching layer that captures the displacement between mid-level features. Ahmed *et al*. [1] capture feature displacements similarly by application of special convolutions that compute the difference between neighborhoods in the feature map of two input images. The gating functions in the network of Varior *et al*. [26] compare features along a horizontal stripe and output a gating mask to indicate how much emphasis should be paid to the local patterns. Finally, in [27] a recurrent siamese neural network architecture is proposed that processes images in rows. The idea behind the recurrent architecture is to increase contextual information through sequential processing.

More recent work on person re-identification suggests that baseline CNN architectures can compete with their specialized counter parts. In particular, the current best performing method on the MARS [35] is a conventional residual network [8]. Application of baseline CNN architectures can be beneficial if pre-trained models are available for fine-tuning to the person re-identification task. Influence of pre-training on overall performance is studied in [35]. They report between 9.5% and 10.2% recognition rate is due to pre-training on ImageNet [11].

3. Standard Softmax Classifier

Given a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ of N training images $\boldsymbol{x}_i \in \mathbb{R}^D$ and associated class labels $y_i \in \{1, \dots, C\}$, the standard approach to classification in the deep learning setting is to process input images by a CNN and place a softmax classifier on top of the network to obtain probability scores for each of the C classes. The softmax classifier chooses the class with maximum probability according to a parametric function

$$p(y = k \mid \boldsymbol{r}) = \frac{\exp\left({\boldsymbol{w}_k}^{\mathrm{T}} \boldsymbol{r} + b_k\right)}{\sum_{n=1}^{C} \exp\left({\boldsymbol{w}_n}^{\mathrm{T}} \boldsymbol{r} + b_n\right)} \tag{1}$$

where $\boldsymbol{r} = f(\boldsymbol{x}), \boldsymbol{r} \in \mathbb{R}^d$ is the underlying feature representation of a parametrized encoder network that is trained jointly with the classifier. For the special case of $C = 2$ classes this formulation is equivalent to logistic regression. Further, the specific choice of functional form can be motivated from a generative perspective on the classification problem. If the class-conditional densities are Gaussian

$$p(\boldsymbol{r} \mid y = k) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{r} - \boldsymbol{\mu}_k)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{r} - \boldsymbol{\mu}_k)\right) \tag{2}$$

with shared covariance $\boldsymbol{\Sigma}$, then the posterior class probability can be computed by Bayes' rule

$$p(y = k \mid \boldsymbol{r}) = \frac{p(\boldsymbol{r} \mid y = k) p(y = k)}{\sum_{n=1}^{C} p(\boldsymbol{r} \mid y = n) p(y = n)} \tag{3}$$

$$= \frac{\exp\left(\boldsymbol{w}_k^{\mathrm{T}} \boldsymbol{r} + b_k\right)}{\sum_{n=1}^{C} \exp\left(\boldsymbol{w}_n^{\mathrm{T}} \boldsymbol{r} + b_n\right)} \tag{4}$$

with $\boldsymbol{w}_k = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$ and $b_k = -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \log p(y_i = k)$ [2]. However, the softmax classifier is trained in a discriminative regime. Instead of determining the parameters of the class-conditional densities and prior class probabilities, the parameters $\{\boldsymbol{w}_1, b_1, \dots, \boldsymbol{w}_C, b_C\}$ of the conditional class probabilities are obtained directly by minimization of a classification loss. Let $\mathbb{1}_{y=k}$ denote the indicator function that evaluates to 1 if y is equal to k and 0 otherwise. Then, the corresponding loss

$$\mathcal{L}(\mathcal{D}) = -\sum_{i=1}^{N} \sum_{k=1}^{C} \mathbb{1}_{y_i=k} \cdot \log\, p(y_i = k \mid \boldsymbol{r}_i) \tag{5}$$

minimizes the cross-entropy between the *true* label distribution $p(y = k) = \mathbb{1}_{y=k}$ and estimated probabilities of the softmax classifier $p(y = k \mid \boldsymbol{r})$. By minimizing the cross-entropy loss, parameters are chosen such that the estimated probability is close to 1 for the correct class and close to 0 for all other classes.

Figure 2a shows three Gaussian densities $p(\boldsymbol{r} \mid y)$ together with the corresponding decision boundary. The posterior class probabilities of this scenario are shown in Figure 2b together with a set of hypothesized training examples. Whereas the Gaussian densities peak around a class mean, the posterior class probability is a function of the distance to the decision boundary. When the feature encoder is trained with the classifier jointly by minimization of the cross-entropy loss, the parameters of the encoder network are adapted to push samples away from the decision boundary as far as possible, but not necessarily towards the class mean that has been taken to motivate the specific functional form. This behavior is problematic for metric learning because similarity in terms of class membership is encoded in the orientation of the decision boundary rather than in the feature representation itself.

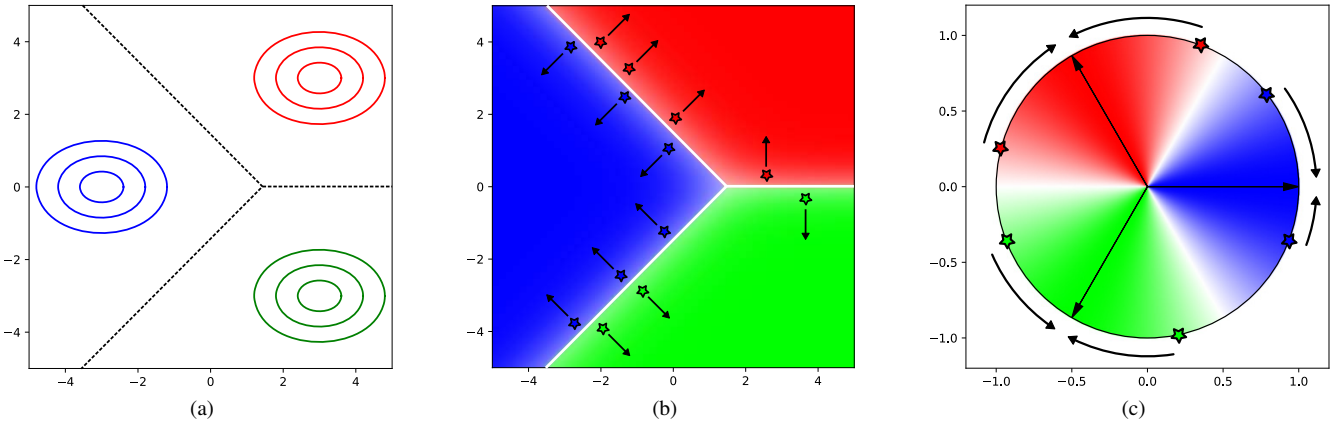


Figure 2: Plot (a) shows three Gaussian class-conditional densities (iso-contours) and the corresponding decision boundary (dashed lines). Plot (b) shows the conditional class probabilities (color coded) and a set of hypothesized training examples. The softmax classifier models the posterior class probabilities directly, without construction of Gaussian densities. By training with the cross-entropy loss, samples are pushed away from the decision boundary, but not necessarily towards a class mean. Plot (c) illustrates the posterior class probabilities (color coded) and decision boundary (white line) of the cosine softmax classifier for three classes. During training, all samples are pushed away from the decision boundary towards their parametrized class mean direction (indicated by an arrow).

4. Cosine Softmax Classifier

With few adaptations the standard softmax classifier can be modified to produce compact clusters in representation space. First, ℓ_2 normalization must be applied to the final layer of the encoder network to ensure the representation is unit length $\|f_\Theta(\boldsymbol{x})\|_2 = 1, \forall \boldsymbol{x} \in \mathbb{R}^D$. Second, the weights must be normalized to unit-length as well, i.e., $\tilde{\boldsymbol{w}}_k = \boldsymbol{w}_k / \|\boldsymbol{w}_k\|_2, \forall k = 1, \ldots, C$. Then, the cosine softmax classifier can be stated by

$$p(y_i = k \mid \boldsymbol{r}_i) = \frac{\exp\left(\kappa \cdot \tilde{\boldsymbol{w}}_k^{\mathrm{T}} \boldsymbol{r}_i\right)}{\sum_{n=1}^{C} \exp\left(\kappa \cdot \tilde{\boldsymbol{w}}_n^{\mathrm{T}}\right)}, \tag{6}$$

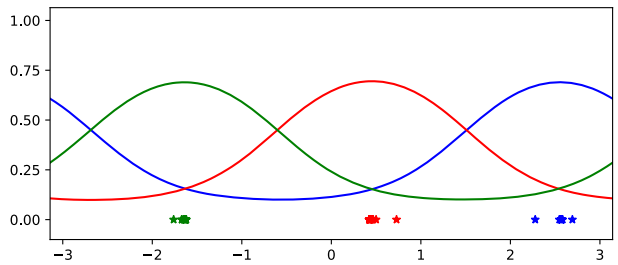
where κ is a free scaling parameter. This parametrization has $C - 1$ fewer parameters compared to the standard formulation (1) because the bias terms b_k have been removed $\{\kappa, \tilde{\boldsymbol{w}}_1, \ldots, \tilde{\boldsymbol{w}}_C\}$. Otherwise, the functional form resembles strong similarity to the standard parametrization and implementation is straight-forward. In particular, decoupling the length of the weight vector κ from its direction has been proposed before [20] as a way to accelerate convergence of stochastic gradient descent. Training itself can be carried out using the cross-entropy loss as usual since the cosine softmax classifier is merely a change of parametrization compared to the standard formulation.

The functional modeling of log-probabilities by $\kappa \cdot \tilde{\boldsymbol{w}}_k^{\mathrm{T}} \boldsymbol{r}$ can be motivated from a generative perspective as well. If the class-conditional likelihoods follow a von Mises-Fisher (vMF) distribution

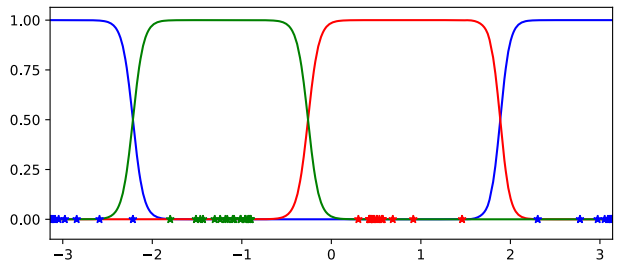
$$p(\boldsymbol{r} \mid y = k) = c_d(\kappa) \exp\left(\kappa \cdot \tilde{\boldsymbol{w}}_k^{\mathrm{T}} \boldsymbol{r}\right) \tag{7}$$

with shared concentration parameter κ and normalizer $c_d(\kappa)$, then Equation 6 is the posterior class probability under an equal prior assumption $p(y = k) = p(y = l), \forall k, l \in \{1, \ldots, C\}$. The vMF distribution is an isotropic probability distribution on the $d - 1$ dimensional sphere in $\mathbb{R}^d$ that peaks around mean direction $\tilde{\boldsymbol{w}}_k$ and decays as the cosine similarity decreases.

To understand why this parametrization enforces a cosine similarity on the representation space, observe that the log-probabilities are directly proportional to the cosine similarity between training examples and a parametrized class mean direction. By minimizing the cross-entropy loss, examples are pushed away from the decision boundary towards their parametrized mean as illustrated in Figure 2c. In consequence, parameter vector $\tilde{\boldsymbol{w}}_k$ becomes a surrogate for all samples in cases k. The scaling parameter κ controls the shape of the conditional class probabilities as illustrated in Figure 3. A low value corresponds to smoother functions with wider support. A high κ value leads to conditional class probabilities that are box-like shaped around the decision boundary. This places a larger penalty on misclassified examples, but at the same time leaves more room for samples to move freely in the region of representation space that is occupied by its corresponding class. In this regard, the scale takes on a similar role to margin parameters in direct metric learning objectives. When the scale is left as a free parameter, the optimizer gradually increases its value as the overlap between classes reduces. A margin between samples of different classes can be enforced by regularizing the scale with weight decay.



(a) $\kappa = 1$



(b) $\kappa = 10$

Figure 3: Illustration of the free scaling parameter κ in a one dimensional problem with three classes. The conditional class probabilities are shown as colored functions. Optimized sample locations are visualized as stars at $y = 0$. A low κ value (a) leads to smoother functions with wider support, such that samples are pushed into tight clusters. The shape becomes box-like for high values (b), allowing samples to move more freely within a region that is occupied by the class.

5. Evaluation

The first part of the evaluation compares both the training behavior and validation error between our loss formulation and common metric learning losses using a network trained from scratch. In the second part, overall system performance is established against existing re-identification systems on the same datasets.

5.1. Network Architecture

The network architecture used in our experiments is relatively shallow to allow for fast training and inference, e.g., for application in the related task of appearance based object tracking [30]. The architecture is summarized in Table 1. Input images are rescaled to 128×64 and presented to the network in RGB color space. A series of convolutional layers reduces the size of the feature map to 16×8 before a global feature vector of length 128 is extracted by layer *Dense 10*. The final ℓ_2 normalization projects features onto the unit hypersphere for application of the cosine softmax classifier. The network contains several residual blocks that follow the pre-activation layout proposed by He *et al*. [7]. The design follows the ideas of wide residual networks [33]: All convolutions are of size 3×3 and max pooling is replaced by convolutions of stride 2. When

Name	Patch Size/Stride	Output Size
Conv 1	$3 \times 3/1$	$32 \times 128 \times 64$
Conv 2	$3 \times 3/1$	$32 \times 128 \times 64$
Max Pool 3	$3 \times 3/2$	$32 \times 64 \times 32$
Residual 4	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 5	$3 \times 3/1$	$32 \times 64 \times 32$
Residual 6	$3 \times 3/2$	$64 \times 32 \times 16$
Residual 7	$3 \times 3/1$	$64 \times 32 \times 16$
Residual 8	$3 \times 3/2$	$128 \times 16 \times 8$
Residual 9	$3 \times 3/1$	$128 \times 16 \times 8$
Dense 10		128
ℓ_2 normalization		128

Table 1: Overview of the CNN architecture. The final ℓ_2 normalization projects features onto the unit hypersphere.

the spatial resolution of the feature map is reduced, then the number of channels is increased accordingly to avoid a bottleneck. Dropout and batch normalization are used as means of regularization. Exponential linear units [4] are used as activation function in all layers.

Note that with in total 15 layers (including two convolutional layers in each residual block) the network is relatively shallow when compared to the current trend of ever deeper architectures [7]. This decision was made for the following two reasons. First, the network architecture has been designed for the application of both person re-identification and online people tracking [30], where the latter requires fast computation of appearance features. In total, the network has 2,800,864 parameters and one forward pass of 32 bounding boxes takes approximately 30 ms on an Nvidia GeForce GTX 1050 mobile GPU. Thus, this network is well suited for online tracking even on low-cost hardware. Second, architectures that have been designed for person re-identification specifically [13, 1] put special emphasis on mid-level features. Therefore, the dense layer is added at a point where the feature map still provides *enough* spatial resolution.

5.2. Datasets and Evaluation Protocols

Evaluation is carried out on the Market 1501 [36] and MARS [35]. Market 1501 contains 1,501 identities and roughly 30,000 images taken from six cameras. MARS is an extension of Market 1501 that contains 1,261 identities and over 1,100,000 images. The data has been generated using a multi-target tracker that generates tracklets, i.e. short-term track fragments, which have then been manually annotated to consistent identities. Both datasets contain considerate bounding box misalignment and labeling inaccuracies. For all experiments a single-shot, cross-view evaluation protocol is adopted, i.e. a single query image from one camera is matched against a gallery of images taken

from different cameras. The gallery image ranking is established using cosine similarity or Euclidean distance, if appropriate. Training and test data splits are provided by the authors. Additionally, 10% of the training data is split for hyperparameter tuning and early stopping. On both datasets cumulative matching characteristics (CMC) at rank 1 and 5 as well as mean average precision (mAP) are reported. The scores are computed with evaluation software provided by the corresponding dataset authors.

5.3. Baseline Methods

In order to assess the performance of the joint classification and metric learning framework on overall performance, the network architecture is repeatedly trained with two baseline direct metric learning objectives.

Triplet loss The triplet loss [28] is defined over tuples of three examples $\boldsymbol{r}_a$, $\boldsymbol{r}_p$, and $\boldsymbol{r}_n$ that include a positive pair $y_a = y_p$ and a negative pair $y_a \neq y_n$. For each such triplet the loss demands that the difference of the distance between the negative and positive pair is larger than a predefined margin $m \in \mathbb{R}$:

$$\mathcal{L}_{\mathrm{t}}(\boldsymbol{r}_a, \boldsymbol{r}_p, \boldsymbol{r}_n) = \left\{ \|\boldsymbol{r}_a - \boldsymbol{r}_p\|_2 - \|\boldsymbol{r}_a - \boldsymbol{r}_n\|_2 + m \right\}_+, \quad (8)$$

where $\{\}_+$ denotes the hinge function that evaluates to 0 for negative values and identity otherwise. In this experiment, a soft-margin version of the original triplet loss [8] is used where the hinge is replaced by a soft plus function $\{x+m\}_+ = \log(1+\exp(x))$ to avoid issues with non-smoothness [19]. Further, the triplets are generated directly on GPU as proposed by [8] to avoid potential issues in the sampling strategy. Note that this particular triplet loss formulation has been used to train the current best performing model on the MARS dataset.

Magnet loss The magnet loss has been proposed as an alternative to siamese loss formulations that works on entire class distribution rather than individual samples. The loss is a likelihood ratio measure that forces separation in terms of each sample's distance away from the means of other classes. In its original proposition [19] the loss takes on a multi-modal form. Here, a simpler, unimodal variation of this loss is employed as it better fits the single-shot person re-identification task:

$$\mathcal{L}_{\mathrm{m}}(y, \boldsymbol{r}) = \left\{ -\log \frac{\mathrm{e}^{-\frac{1}{2\hat{\sigma}^2}\|\boldsymbol{r}-\hat{\boldsymbol{\mu}}_y\|_2^2 - m}}{\sum_{k \in \bar{\mathcal{C}}(y)} \mathrm{e}^{-\frac{1}{2\hat{\sigma}^2}\|\boldsymbol{r}-\hat{\boldsymbol{\mu}}_k\|_2^2}} \right\}_+, \quad (9)$$

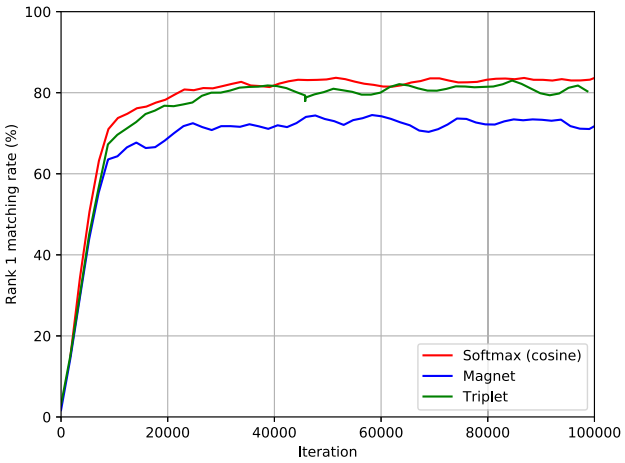
where $\bar{\mathcal{C}}(y) = \{1, \ldots, C\} \setminus \{y\}$, m is again a margin parameter, $\hat{\boldsymbol{\mu}}_y$ is the sample mean of class y, and $\hat{\sigma}^2$ is the variance of all samples away from their class mean. These parameters are computed on GPU for each batch individually.

5.4. Results

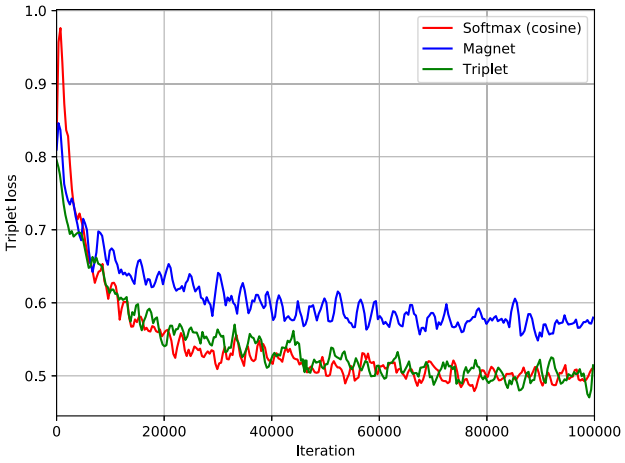
The results reported in this section have been established by training the network for a fixed number of $100,000$ iterations using Adam [9]. The learning rate was set to 1×10^{-3}. As can be seen in Figure 4 all configurations have fully converged at this point. The network was regularized with a weight decay of 1×10^{-8} and dropout inside the residual units with probability 0.4. The margin of the magnet loss has been set to $m = 1$ and the cosine softmax scale κ was left as a free parameter for the optimizer to tune, but regularized with a weight decay of 1×10^{-1}. The batch size was fixed to 128 images. Gallery rankings are established using Euclidean distance in case of magnet and triplet loss, while cosine similarity is used for the softmax classifier. To increase variability in the training set, input images have been randomly flipped, but no random resizing or cropping has been performed.

Training Behavior Figure 4a shows the rank 1 matching rate on the validation set of MARS as a function of training iterations. The results obtained on Market 1501 are omitted here since the training behavior is similar. The network trained with cosine softmax classifier achieves overall best performance, followed by the network trained with soft-margin triplet loss. The best validation performance of the softmax network is reached at iteration $49\,760$ with rank 1 matching rate 84.92%. The best performance of the triplet loss network is reached at iteration $86\,329$ with rank 1 matching rate 83.23%. The magnet loss network reaches its best performance at iteration $47\,677$ with rank 1 matching rate 77.34%. Overall, the convergence behavior of the three losses is similar, but the magnet loss falls behind on final model performance. In its original implementation [19] the authors sample batches such that similar classes appear in the same batch. For practical reasons such more informative sample mining has not been implemented. Instead, a fixed number of images per individual was randomly selected for each batch. Potentially, the magnet loss suffers from this less informative sampling strategy more than the other two losses.

During all runs the triplet loss has been monitored as an additional information source on training behavior. Figure 4b plots the triplet loss as a function of training iterations. Note that the triplet loss has not been used as a training objective in runs softmax (cosine) and magnet. Nevertheless, both minimize the triplet loss indirectly. In particular the softmax classifier is quite efficient at minimizing the triplet loss. During iterations 20,000 to 40,000 the triplet loss drops even slightly faster when optimization is carried out with the softmax classifier rather than optimizing the triplet loss directly. Therefore, the cosine softmax classifier effectively enforces a similarity metric onto the representation space.



(a) Evolution of validation set accuracy



(b) Evolution of triplet loss on training set

Figure 4: Plot (a) shows the rank 1 matching accuracy on the validation set as a function of training iterations. Plot (b) shows how the triplet loss evolves on the training set. Note that the triplet loss is only used as training objective for the triplet network. For the other two methods the loss is only monitored to obtain insight into the training behavior.

Re-Identification Performance All three networks have been evaluated on the provided test splits of the Market 1501 and MARS datasets. Table 2 and 3 summarize the results and provide a comparison against the state of the art. The training behavior and rank 1 matching rates that have been observed on the validation set manifest in the final performance on the provided test splits. Of our own networks, on both datasets the cosine softmax network achieves the best results, followed by the siamese network. The gain in mAP due to the softmax loss is 3.64 on the Market 1501 dataset and 2.58 on the MARS dataset. This is a relative gain of 6.8% and 4.7% respectively. The state of the art contains several alternative siamese architectures that have been trained with a contrastive or triplet loss, marked by [b] in Table 2 and 3. The performance of these networks is not always directly comparable, because the models have varying capacity. However, the LuNet of Hermans *et al*. [8] is a residual network with roughly double the capacity of the proposed architecture. The reported numbers have been generated with test-time data augmentation that accounts for approximately 3 mAP points according to the corresponding authors. Thus, the proposed network comes in close range at much lower capacity. Further, the method of [35] refers to a CaffeNet that has been trained with the conventional softmax classifier and the metric subspace has been obtained in a separate post processing step. The results suggest that the proposed joint classification and metric learning framework not only enforces a metric onto the representation space, but also that encoding the metric directly into the classifier works better than treating it in a subsequent post processing step.

Method	Market 1501		
	Rank 1	Rank 5	mAP
TriNet [8][a,b]	84.92	94.21	69.14
LuNet [8][b]	81.38	92.34	60.71
IDE + XQDA [35][a,†]	73.60	-	49.05
DaF [32][a]	82.30	-	72.42
JLML [14][a]	85.10	-	65.50
GoogLeNet [34][a]	81.00	-	63.40
SVDNet [23][a]	82.30	-	62.10
Gated CNN [26][b]	65.88	-	39.55
Recurrent CNN [27][b]	61.60	-	35.30
Ours (triplet)[b]	74.88	88.72	53.04
Ours (magnet)	61.10	81.03	40.12
Ours (cosine softmax)	79.10	91.06	56.68

Table 2: Performance comparison on Market 1501 [36]. [†]: Numbers taken from [8]. Methods below the line show our network architecture trained with different losses. [a]: Pre-trained on ImageNet. [b]: Siamese network.

The best performing method on Market 1501 has a 15.84 points higher mAP score than the cosine softmax network. On MARS, the best performing method achieves a 10.82 higher mAP. This is a large-margin improvement over the proposed network, which shows that considerate improvement is possible by application of larger capacity architectures with additional pre-training. Note that, for example, the TriNet [8] is a ResNet-50 [6] with 25.74 million parameters that has been pre-trained on ImageNet [11]. With roughly a tenth of the parameters, our network has much lower capacity. The best performing network that has been trained from scratch, i.e., without pre-training on ImageNet, is the LuNet of Hermans *et al*. [8]. With approximately 5 million parameters the network is still roughly double the

Method	MARS		
	Rank 1	Rank 5	mAP
TriNet [8][a,b]	79.80	91.36	67.70
LuNet [8][b]	75.56	89.70	60.48
IDE + XQDA [35][a,†]	65.30	82.00	47.60
MSCAN [12]	71.77	86.57	56.06
P-QAN [17]	73.73	84.90	51.70
CaffeNet [38]	70.60	90.00	50.70
Ours (triplet)[b]	71.31	85.55	54.30
Ours (magnet)	63.13	81.16	45.45
Ours (cosine softmax)	72.93	86.46	56.88

Table 3: Performance comparison on MARS [35]. Methods below the line show our network architecture trained with different losses. †: Numbers taken from [8]. [a]: Pre-trained on ImageNet. [b]: Siamese network.

size, but the final model performance in terms of mAP is only 4.03 and 3.6 points higher (including test-time augmentation). Therefore, the proposed architecture provides a good trade off between computational efficiency and re-identification performance.

Learned Embedding Figure 1 and 5 show a series of exemplary queries computed from the Market 1501 test gallery. The queries shown in Figure 1 represent a selection of many identities that the network successfully identifies by nearest neighbor search. In many cases, the feature representation is robust to varying poses as well as changing background and image quality. Figure 5 shows some challenging queries and interesting failure cases. For example, in the second row the network seems to focus on the bright handbag in a low-resolution capture of a woman. The top five results returned by the network contain four women with colorful clothing. In the third row the network fails to correctly identify the gender of the queried identity. In the last example, the network successfully re-identifies a person that is first sitting on a scooter and later walks (rank 4 and 5), but also returns a wrong identity with similarly striped sweater (rank 3). A visualization of the learned embedding on the MARS test split is shown in Figure 6.

Figure 5: Failure cases on example queries generated from Market 1501 [36] test gallery. For each query image the five most similar and dissimilar images are shown.

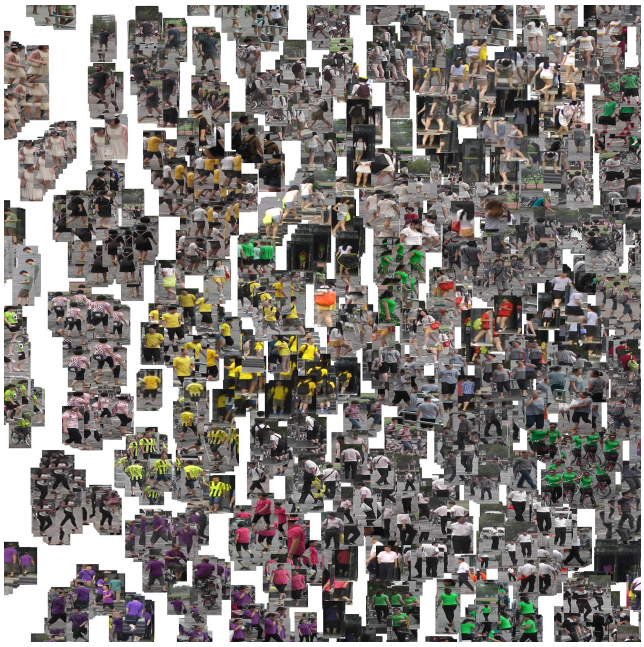


Figure 6: Excerpt of the learned embedding on the MARS test split generated with t-SNE [25].

6. Conclusion

We have presented a re-parametrization of the conventional softmax classifier that enforces a cosine similarity on the representation space when trained to identify the individuals in the training set. Due to this property, the classifier can be stripped of the network after training and queries for unseen identities can be performed using nearest-neighbor search. Thus, the presented approach offers a simple, easily applicable alternative for metric learning that does not require sophisticated sampling strategies. In our experiments, training in this regime provided a modest gain in test performance. While the method itself is general, our evaluation was limited to a very specific application using a single light-weight CNN architecture. In future work, the approach should be further validated on more datasets and application domains. Such an evaluation should also include larger capacity architectures and pre-training on ImageNet.

References

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

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, 2006.

[3] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pages 539–546, 2005.

[4] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *ICLR*, pages 1–14, 2015.

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655, 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645, 2016.

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

[9] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, pages 1–15, 2015.

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

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[12] D. Li, X. Chen, Z. Zhang, and K. Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *CVPR*, pages 384–393, 2017.

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

[14] W. Li, X. Zhu, and S. Gong. Person re-identification by deep joint learning of multi-loss classification. In *IJCAI*, pages 2194–2200, 2017.

[15] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, pages 2197–2206, 2015.

[16] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014.

[17] Y. Liu, J. Yan, and W. Ouyang. Quality aware network for set to set recognition. In *CVPR*, pages 1–10, 2017.

[18] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, pages 4004–4012, 2016.

[19] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev. Metric learning with adaptive density discrimination. *ICLR*, pages 1–15, 2016.

[20] T. Salimans and D. P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NIPS*, pages 901–909, 2016.

[21] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015.

[22] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshops*, pages 806–813, 2014.

[23] Y. Sun, L. Zheng, W. Deng, and S. Wang. Svdnet for pedestrian retrieval. In *ICCV*, pages 3800–3808, 2017.

[24] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014.

[25] L. Van Der Maaten. Accelerating t-sne using tree-based algorithms. *JMLR*, 15(1):3221–3245, 2014.

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

[27] 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, 2016.

[28] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *JMLR*, 10(Feb):207–244, 2009.

[29] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, pages 499–515, 2016.

[30] N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric. In *ICIP*, pages 3645–3649, 2017.

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

[32] R. Yu, Z. Zhou, S. Bai, and X. Bai. Divide and fuse: A re-ranking approach for person re-identification. In *BMVC*, pages 1–13, 2017.

[33] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, pages 1–12, 2016.

[34] L. Zhao, X. Li, J. Wang, and Y. Zhuang. Deeply-learned part-aligned representations for person re-identification. In *ICCV*, pages 3219–3228, 2017.

[35] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian. MARS: A video benchmark for large-scale person re-identification. In *ECCV*, pages 868–884, 2016.

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

[37] L. Zheng, H. Zhang, S. Sun, M. Chandraker, and Q. Tian. Person re-identification in the wild. In *CVPR*, 2017.

[38] Z. Zhou, Y. Huang, W. Wang, L. Wang, and T. Tan. See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification. In *CVPR*, pages 4747–4756, 2017.