DEEP COSINE METRIC LEARNING FOR PERSON RE-IDENTIFICATION

LIDOR BURSHTEIN

GIL BEN DAVID

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.

ABSTRACT

- 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.

INTRODUCTION

- Classification:
  - Many samples
  - K classes
  - Centroid-based
- One-Shot Learning
  - Few samples
  - Unknown/Big amount of classes

RELATED WORK – METRIC LEARNING

- CNNs - trained on top of a general-purpose feature representation that was learned beforehand on ImageNet or MS COCO
- Feature Representation – Stripping the final layer of the classifier
- NN queries – by Euclidean distance or cosine similarity

RELATED WORK – METRIC LEARNING

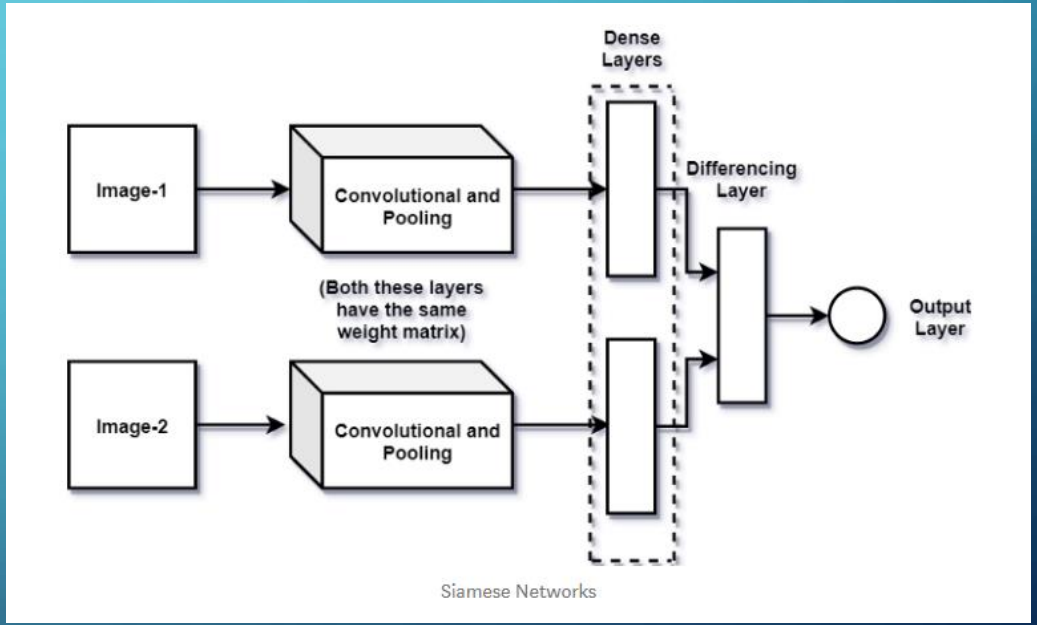
- Siamese Networks
  - Contrastive Loss
  - Triplet Loss
- Sampling Strategies
  - Magnet loss
- Classification + Metric Learning
  - Center Loss

KEYWORDS

- Metric Learning
- Cosine Similarity
- Siamese Networks
- Contrastive Loss
- Triplet Loss
- Magnet Loss
- Center Loss
- Sampling Strategy

SIAMESE NETWORKS

- One-Shot learning method
- N-way one-shot learning



RELATED WORK – METRIC LEARNING

- Siamese Networks
  - Contrastive Loss
  - Triplet Loss
- Sampling Strategies
  - Magnet loss
- Classification + Metric Learning
  - Center Loss

TRIPLET LOSS

- Given 3 samples: $r_a, r_p, r_n,$ s.t. $\mathrm{y_a} = \mathrm{y_p}$ AND $\mathrm{y_a} \neq y_n$ (triplet)
- $\mathcal{L}_t(r_a, r_p, r_n) = \left\{\|r_a - r_n\|_2 - \|r_a - r_p\|_2 + m\right\}_+$
- $\{\}_+$ = hinge function -> $h(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$
- Here, the hinge replaced with soft-plus:

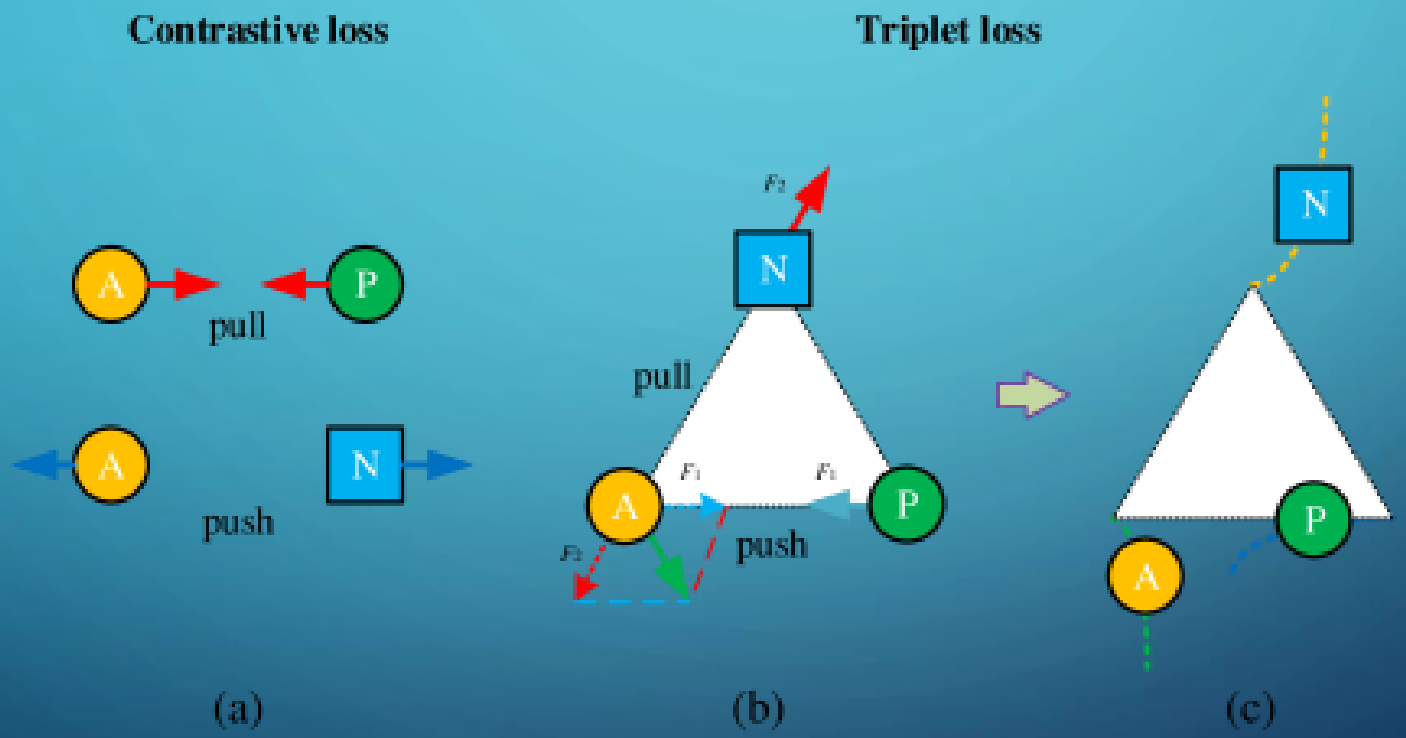
$$\{x + m\}_+ = \log(1 + \exp(x + m))$$

CONTRASTIVE LOSS

- $\mathcal{L}_c\ (x_0, x_1, y)\ =\ y\|d(x_0, x_1)\|\ +\ (1-y)\ max(0, m\ -\|d(x_0, x_1)\|)$
- m – margin

- $y = \begin{cases} 1, & y(x_1) = y(x_2) \\ 0, & y(x_1) \neq y(x_2) \end{cases}$

TRIPLET VS CONTRASTIVE



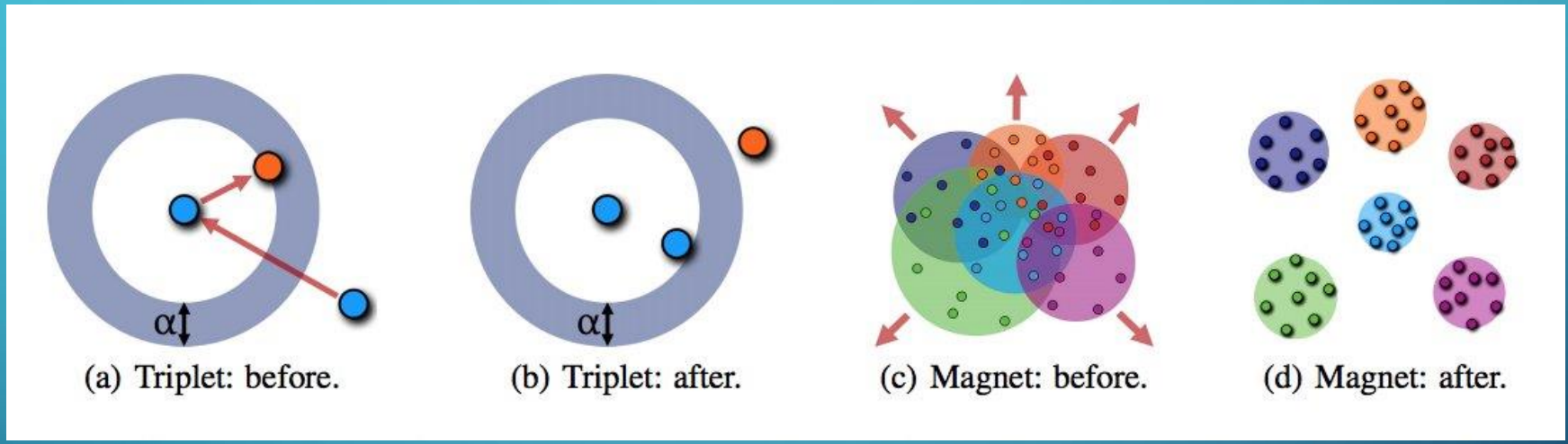
RELATED WORK – METRIC LEARNING

- Siamese Networks
  - Contrastive Loss
  - Triplet Loss
- Sampling Strategies
  - Magnet loss
- Classification + Metric Learning
  - Center Loss

MAGNET LOSS

- $\mathcal{L}_m(y, r) = \left\{ -\log \left(\frac{e^{-\frac{1}{2\hat{\sigma}^2}\|r-\hat{\mu}_y\|_2^2 - m}}{\sum_{k \in \bar{\mathcal{C}}(y)} e^{-\frac{1}{2\hat{\sigma}^2}\|r-\hat{\mu}_y\|_2^2}} \right) \right\}_+$
- Where $\bar{\mathcal{C}}(y) = \{1, \ldots, C\} \setminus \{y\}$, m – margin, $\hat{\mu}_y$ class mean and $\hat{\sigma}^2$ variance of all samples away from their class mean.

TRIPLET VS MAGNET



(a) Triplet: before. (b) Triplet: after. (c) Magnet: before. (d) Magnet: after.

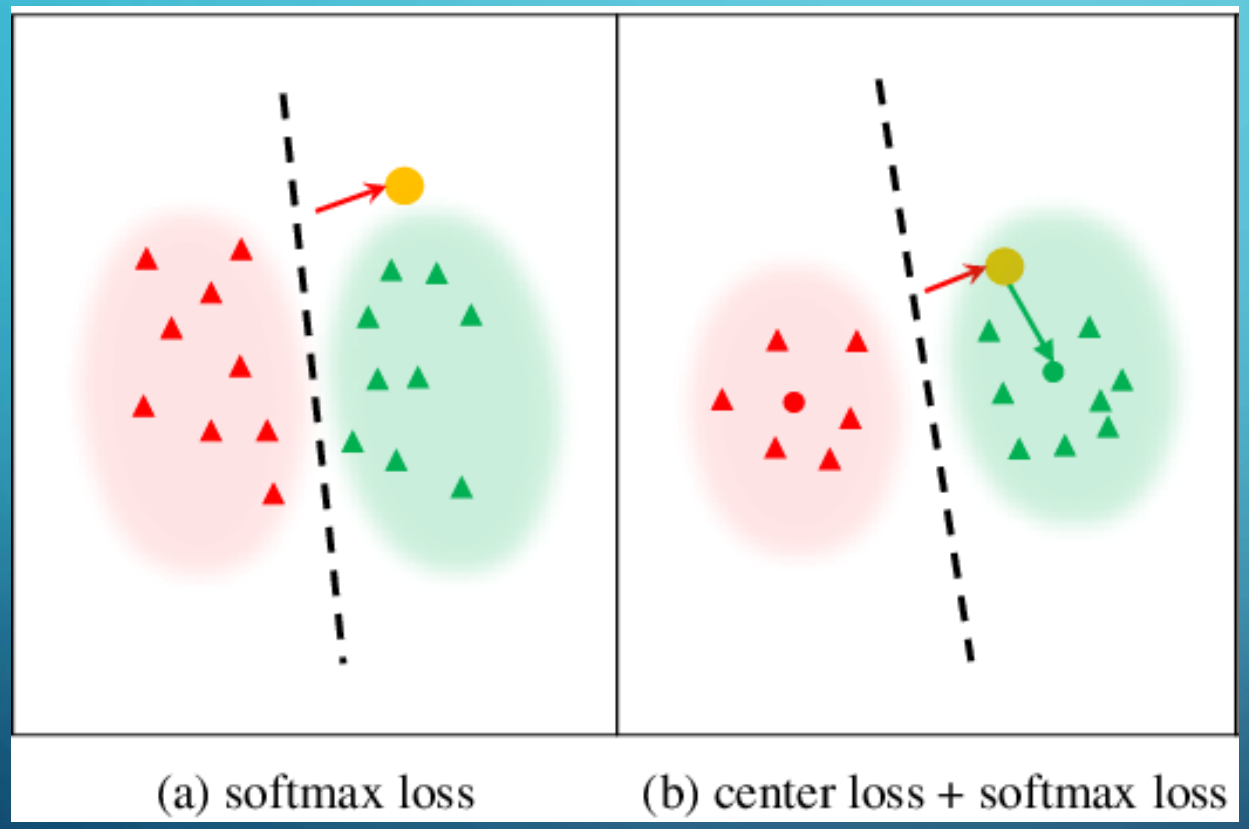
RELATED WORK – METRIC LEARNING

- Siamese Networks
  - Contrastive Loss
  - Triplet Loss
- Sampling Strategies
  - Magnet loss
- Classification + Metric Learning
  - Center Loss

CENTER LOSS

- $\mathcal{L}_{center}(x, c_y) = \frac{1}{2}\sum_{i=1}^{m}\|x_i - c_{y_i}\|_2^2$
- Where c_y is the class center.
- Joint with CE loss:
  - $L = L_{CE} + \lambda L_{Center}$
  - λ is the scalar used to balance between the 2 losses

TRIPLET VS MAGNET

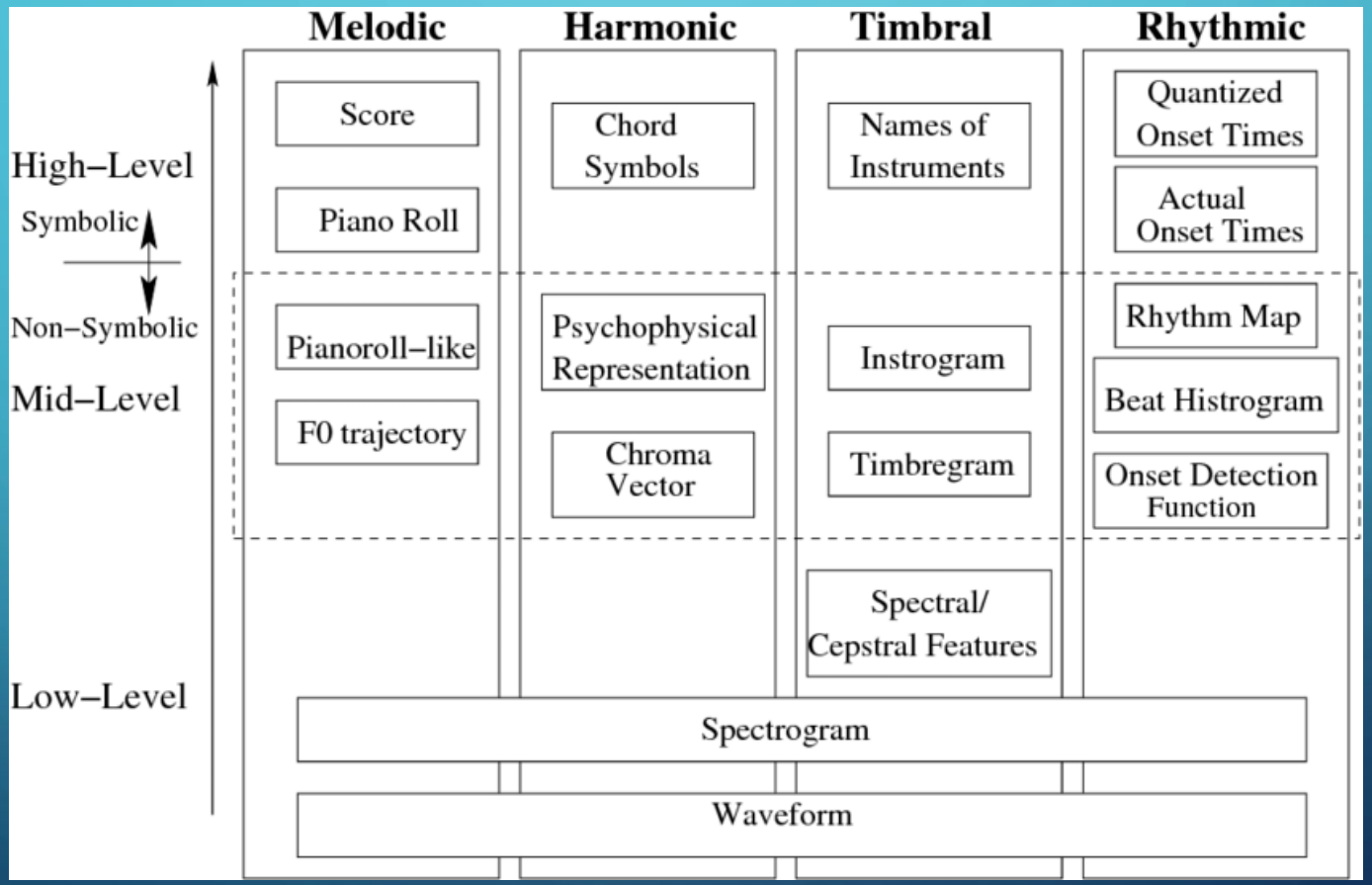


(a) softmax loss (b) center loss + softmax loss

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.

FEATURE LEVELS



PERSON RE-IDENTIFICATION

- 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 is a conventional residual network (ResNet).
- Application of baseline CNN architectures can be beneficial if pre-trained models are available for finetuning to the person re-identification task.

STANDARD SOFTMAX CLASSIFIER

- Given a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ of N training images $x_i \in \mathbb{R}^D$ 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.

$$p(y = k|r) = \frac{\exp(w_k^T r + b_k)}{\sum_{n=1}^{C} \exp(w_n^T r + b_n)}$$

- Where $\mathrm{r} = \mathrm{f}(\mathrm{x}),\ r \in \mathbb{R}^d$ is the underlying feature representation of a parametrized encoder network that is trained jointly with the classifier.
- The parameters $\{w_1, b_1, \dots w_c, b_c\}$ are obtained directly by minimization of a classification loss.

STANDARD SOFTMAX CLASSIFIER

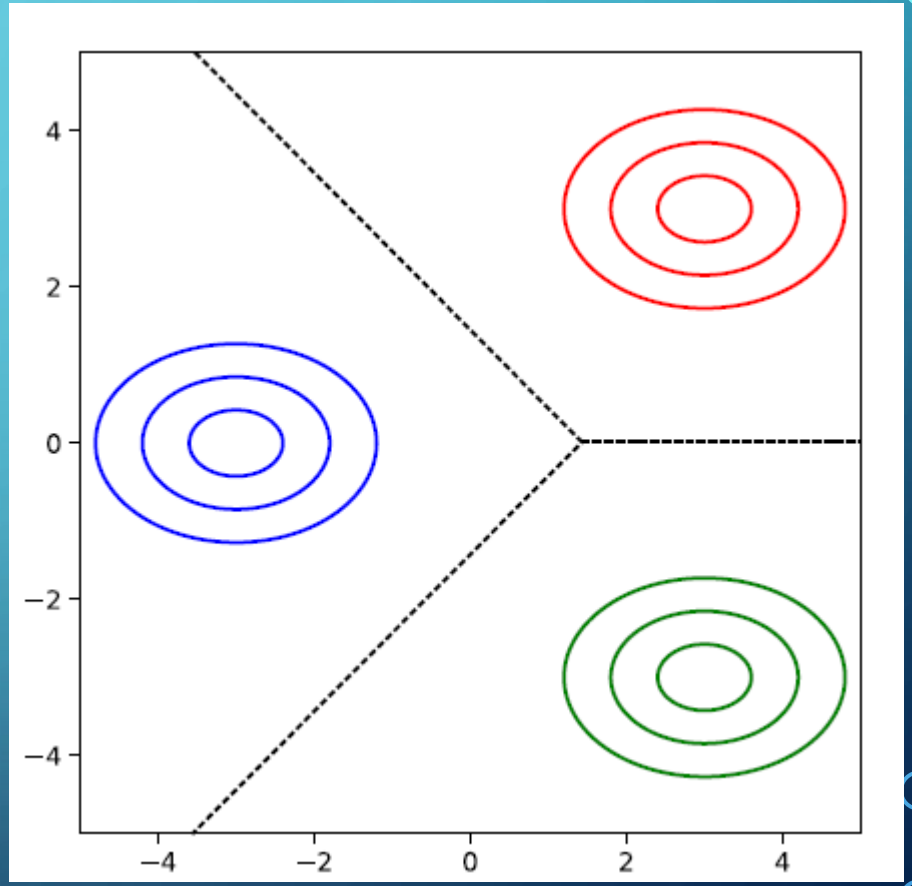
- Then, the cross-entropy loss is defined as:

$$L(D) = -\sum_{i=1}^{N}\sum_{k=1}^{C} 1_{y_i=k} \cdot \log p(y_i = k|r_i)$$

- 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.

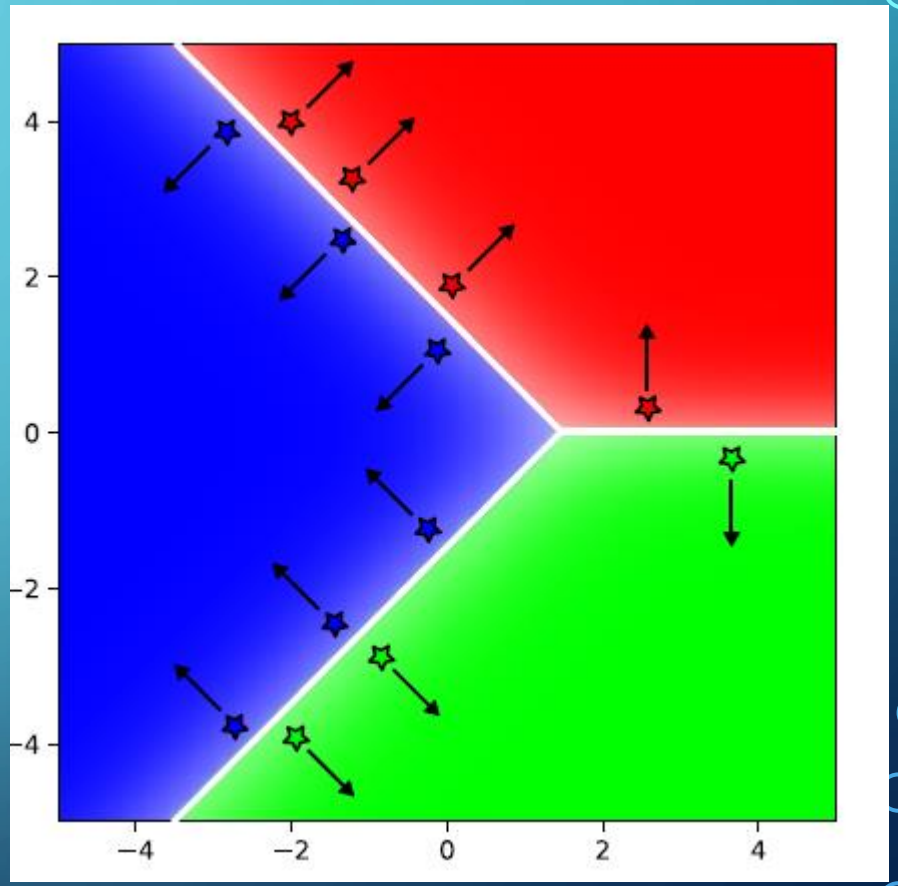
STANDARD SOFTMAX CLASSIFIER

- The following plot shows three Gaussian class-conditional densities (iso-contours) and the corresponding decision boundary (dashed lines).



STANDARD SOFTMAX CLASSIFIER

- 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.



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:

$$\|f_\theta(x)\|_2 = 1, \qquad \forall x \in R^D$$

- Second, the weights must be normalized to unit-length as well, i.e.,

$$\widetilde{w_k} = \frac{w_k}{\|w_k\|_2}$$

COSINE SOFTMAX CLASSIFIER

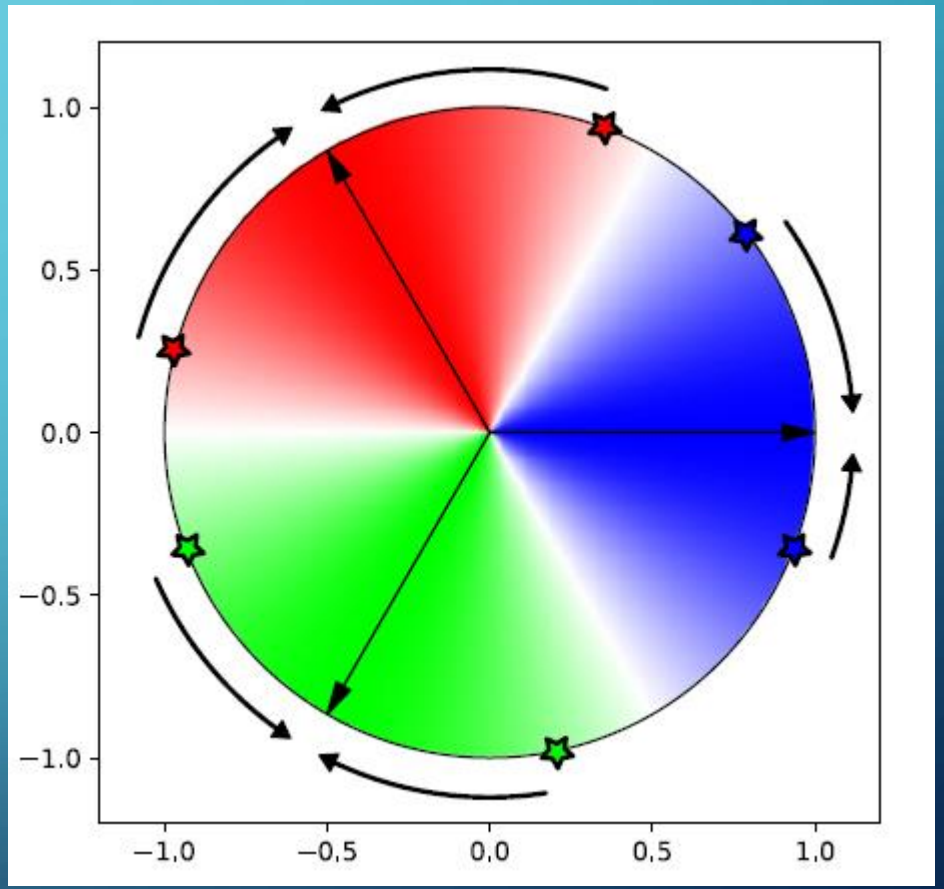
- Then, the cosine softmax classifier can be stated by:

$$p(y = k|r) = \frac{\exp(\kappa \cdot \widetilde{w}_k^T r)}{\sum_{n=1}^{C} \exp(\kappa \cdot \widetilde{w}_n^T r)}$$

- Where κ is a free scaling parameter.
- This parametrization has C-1 fewer parameters compared to the standard formulation because the bias terms b_k have been removed.

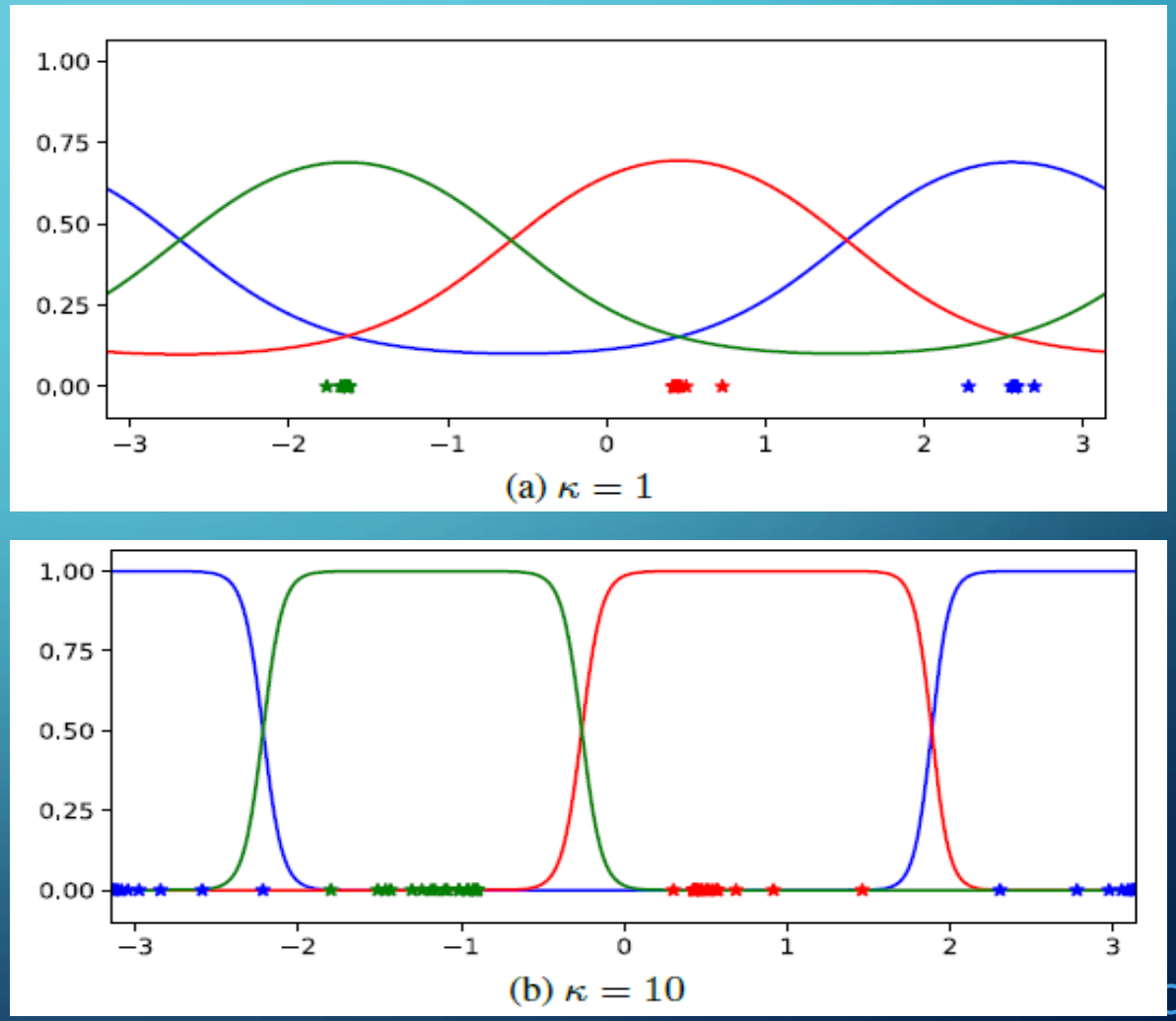
COSINE SOFTMAX CLASSIFIER

- During training, all samples are pushed away from the decision boundary towards their parametrized class mean direction.



COSINE SOFTMAX CLASSIFIER

- The scaling parameter κ controls the shape of the conditional class probabilities.
- A low value corresponds to smoother functions with wider support.
- A high value leads to conditional class probabilities that are box-like shaped around the decision boundary.



NETWORK ARCHITECTURE

- The network architecture used in the experiments is relatively shallow to allow for fast training and inference.
- Input images are rescaled to 128X64 and presented to the network in RGB color space.
- A series of convolutional layers reduces the size of the feature map to 16X8 before a global feature vector of length 128 is extracted.

NETWORK ARCHITECTURE

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.

In total, the network has **2,800,864 parameters.**

DATASETS

- Evaluation is carried out on the Market 1501 and MARS.
- 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.
- Both datasets contain considerate bounding box misalignment and labelling inaccuracies.

EVALUATION PROTOCOLS

- For all experiments 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.

EVALUATION PROTOCOLS

- 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.

CUMULATIVE MATCHING CHARACTERISTICS

- Cumulative Matching Characteristics (CMC) curves are the most popular evaluation metrics for person re-identification methods.
- Consider a simple single-gallery-shot setting, where each gallery identity has only one instance.
- For each query, an algorithm will rank all the gallery samples according to their distances to the query from small to large, and the CMC top-k accuracy is defined by the following shifted step function:

CUMULATIVE MATCHING CHARACTERISTICS

- A_{CMC_k}

$$= \begin{cases} 1, & \textit{if top}-k \textit{ ranked gallery samples contain the query identity} \\ 0, & \textit{otherwise} \end{cases}$$

- The final CMC curve is computed by averaging the shifted step functions over all the queries.

MEAN AVERAGE PRECISION

- For a set of Q queries, the mean of the average precision scores for each query.

$$mAP = \sum_{q=1}^{Q} \frac{AveP(q)}{Q}$$

RESULTS

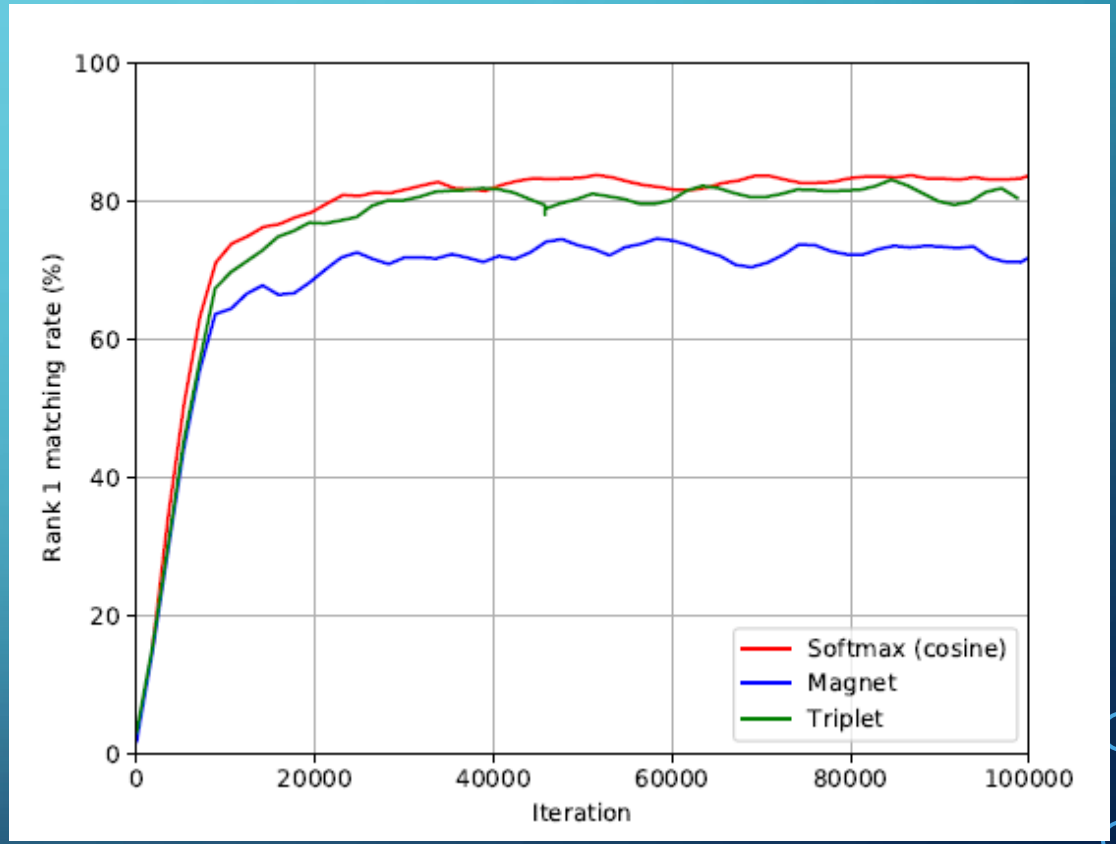
- The results reported in this section have been established by training the network for a fixed number of **100,000** iterations using Adam.
- The learning rate was set to $\boldsymbol{1X10^{-3}}$.
- The dropout probability inside the residual units is **0.4**.
- The batch size was fixed to **128 images**.

RESULTS

- The margin of the magnet loss has been set to $\mathbf{m = 1}$.
- The cosine softmax scale κ was left as a free parameter for the optimizer to tune.
- To increase variability in the training set, input images have been randomly flipped, but no random resizing or cropping has been performed.

RESULTS - TRAINING BEHAVIOR

The network trained with cosine softmax classifier achieves overall best performance, followed by the network trained with soft-margin triplet loss.



RESULTS - TRAINING BEHAVIOR

- 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.

RESULTS - TRAINING BEHAVIOR

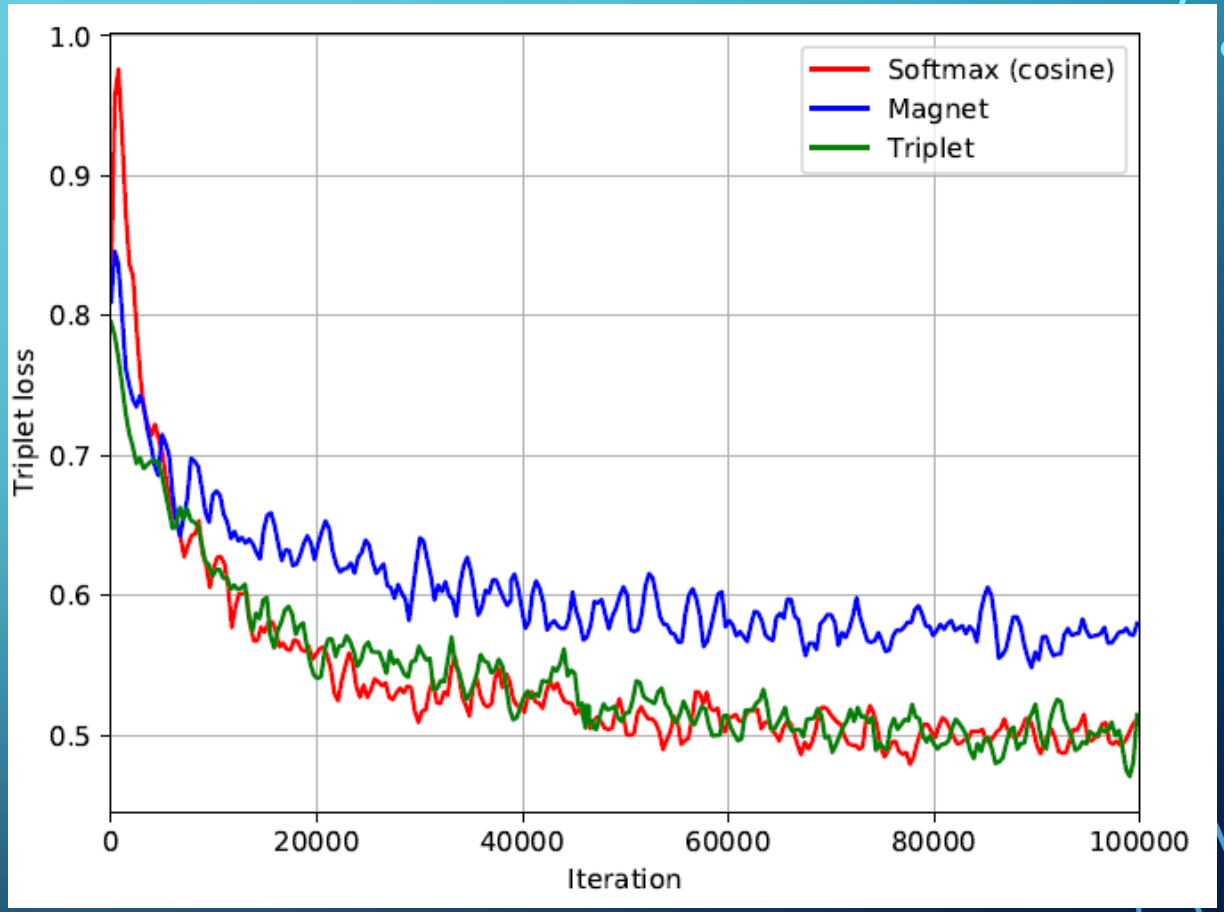
- In the magnet network original implementation, the batches were sampled such that similar classes appear in the same batch.
- For practical reasons such more informative sample mining has not been implemented. Instead, a fixed number of images per individual was randomly selected for each batch.
- Potentially, the magnet loss suffers from this less informative sampling strategy more than the other two losses.

RESULTS - TRAINING BEHAVIOR

- During all runs the **triplet loss** has been monitored as an additional information source on training behavior.
- 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.

RESULTS - TRAINING BEHAVIOR

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.



RESULTS - RE-IDENTIFICATION

- All three networks have been evaluated on the provided test splits of the Market 1501 and MARS datasets.
- Among the authors' networks, on both datasets, the cosine softmax network achieves the best results, followed by the siamese network.
- The proposed architecture provides a good trade-off between computational efficiency and reidentification performance.

RESULTS - RE-IDENTIFICATION

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.

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.

LEARNED EMBEDDING

- In many cases, the feature representation is robust to varying poses as well as changing background and image quality.

LEARNED EMBEDDING

- The next figure shows some challenging queries and interesting failure cases from Market 1501.

LEARNED EMBEDDING

- 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.

LEARNED EMBEDDING

- In the third row the network fails to correctly identify the gender of the queried identity.

LEARNED EMBEDDING

- In the last example, the network successfully re-identifies a person that is first sitting on a scooter and later walks, but also returns a wrong identity with similarly striped sweater.

LEARNED EMBEDDING

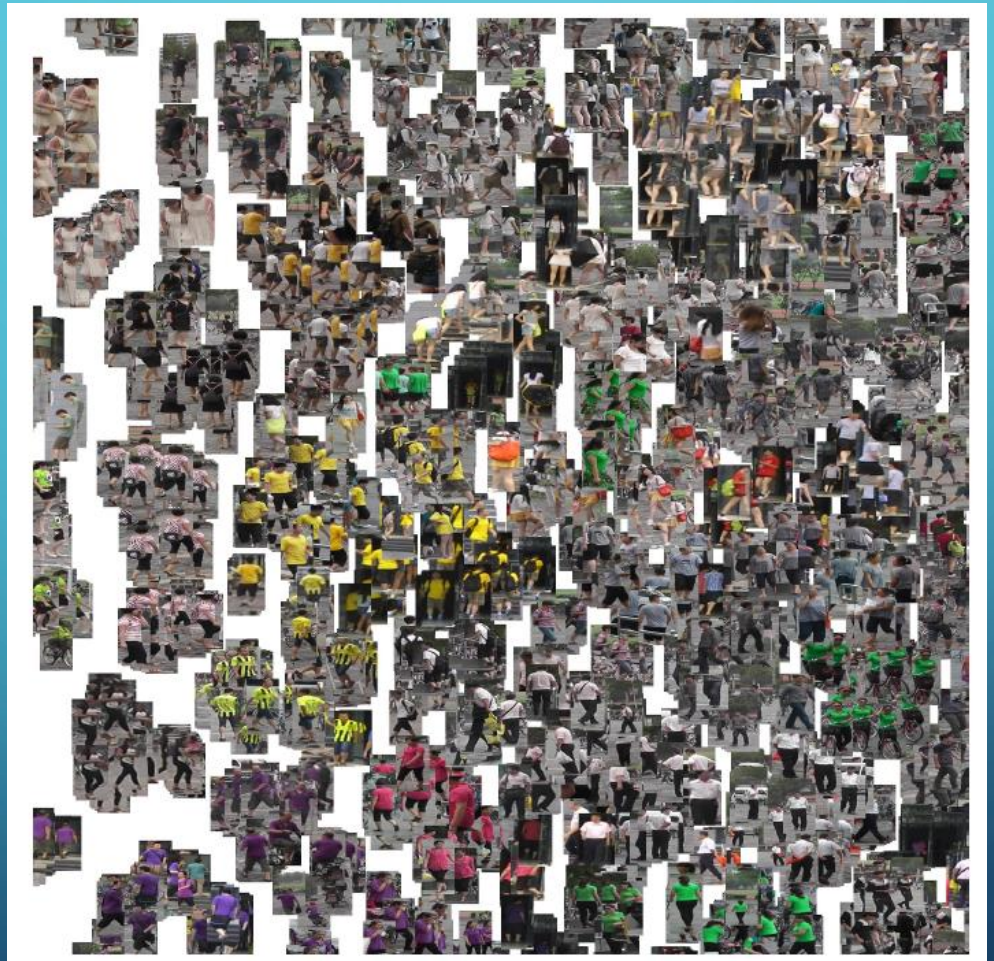


Figure 6: Excerpt of the learned embedding on the MARS test split generated with t-SNE [25].

CODE

- We were unable to run the code on BI's servers due to overload
- The net runs since last week on Gil's computer
- The pretrain-model which introduced in GitHub require permissions, so we sent an email to the author:

CODE

Automatic reply: Deep Cosine Metric-Learning for Person Re-Identification

דואר נכנס ×

יום א', 15ביולי, 21:26

Nicolai.Wojke@dlr.de

אני

Hello and thank you for your message. I am no longer with the DLR. Please address project related communication to Franz Andert (franz.andert@dlr.de).

You may find updated contact information on my Google scholar page [1].

Kind regards,
Nicolai Wojke

[1] https://scholar.google.com/citations?user=i8BCLogAAAAJ

SUMMARY

- The paper presents a re-parametrization of the conventional softmax classifier that enforces a cosine similarity on the representation space.
- 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.

SUMMARY

- Thus, the presented approach offers a simple, easily applicable alternative for metric learning that does not require sophisticated sampling strategies and achieve relatively very satisfied performances.