SVDNet for Pedestrian Retrieval

Yifan Sun[†], Liang Zheng[‡], Weijian Deng[§], Shengjin Wang[†*]
[†]Tsinghua University [‡]University of Technology Sydney
[§]University of Chinese Academy of Sciences
sunyf15@mails.tsinghua.edu.cn, {liangzheng06, dengwj16}@gmail.com, wgsgj@tsinghua.edu.cn

Abstract

This paper proposes the SVDNet for retrieval problems, with focus on the application of person re-identification (re-ID). We view each weight vector within a fully connected (FC) layer in a convolutional neuron network (CNN) as a projection basis. It is observed that the weight vectors are usually highly correlated. This problem leads to correlations among entries of the FC descriptor, and compromises the retrieval performance based on the Euclidean distance. To address the problem, this paper proposes to optimize the deep representation learning process with Singular Vector Decomposition (SVD). Specifically, with the restraint and relaxation iteration (RRI) training scheme, we are able to iteratively integrate the orthogonality constraint in CNN training, yielding the so-called SVDNet. We conduct experiments on the Market-1501, CUHK03, and DukeMTMC-reID datasets, and show that RRI effectively reduces the correlation among the projection vectors, produces more discriminative FC descriptors, and significantly improves the re-ID accuracy. On the Market-1501 dataset, for instance, rank-1 accuracy is improved from 55.3% to 80.5% for CaffeNet, and from 73.8% to 82.3% for ResNet-50.

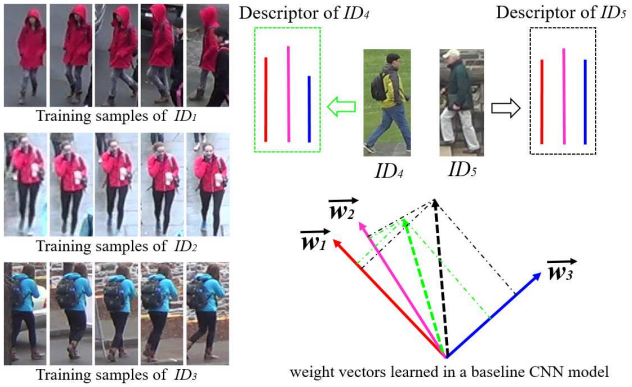


Figure 1: A cartoon illustration of the correlation among weight vectors and its negative effect. The weight vectors are contained in the last fully connected layer, *e.g.*, FC8 layer of CaffeNet [12] or FC layer of ResNet-50 [11]. There are three training IDs in red, pink and blue clothes from the DukeMTMC-reID dataset [17]. The dotted green and black vectors denote feature vectors of two testing samples before the last FC layer. Under the baseline setting, the red and the pink weight vectors are highly correlated and introduce redundancy to the descriptors.

1. Introduction

This paper considers the problem of pedestrian retrieval, also called person re-identification (re-ID). This task aims at retrieving images containing the same person to the query.

Person re-ID is different from image classification in that the training and testing sets contain entirely different classes. So a popular deep learning method for re-ID consists of 1) training a classification deep model on the training set, 2) extracting image descriptors using the fully-connected (FC) layer for the query and gallery images, and 3) computing similarities based on Euclidean distance before returning the sorted list [33, 31, 26, 10].

Our work is motivated by the observation that after training a convolutional neural network (CNN) for classification, the weight vectors within a fully-connected layer (FC) are usually highly correlated. This problem can be attributed to two major reasons. The first reason is related to the non-uniform distribution of training samples. This problem is especially obvious when focusing on the last FC layer. The output of each neuron in the last FC layer represents the similarity between the input image and a corresponding identity. After training, neurons corresponding to similar persons (*i.e.*, the persons who wear red and pink clothes) learns highly correlated weight vectors, as shown in Fig. 1. The second is that during the training of CNN, there exists few, if any, constraints for learning orthogonalization. Thus the learned weight vectors may be naturally correlated.

Correlation among weight vectors of the FC layer com-

*Corresponding Author

promises the descriptor significantly when we consider the retrieval task under the Euclidean distance. In fact, a critical assumption of using Euclidean distance (or equivalently the cosine distance after ℓ_2-normalization) for retrieval is that the entries in the feature vector should be possibly independent. However, when the weight vectors are correlated, the FC descriptor – the projection on these weight vectors of the output of a previous CNN layer – will have correlated entries. This might finally lead to some entries of the descriptor dominating the Euclidean distance, and cause poor ranking results. For example, during testing, the images of two different persons are passed through the network to generate the green and black dotted feature vectors and then projected onto the red, pink and blue weight vectors to form the descriptors, as shown in Fig. 1. The projection values on both red and pink vectors are close, making the two descriptors appear similar despite of the difference projected on the blue vector. As a consequence, it is of vital importance to reduce the redundancy in the FC descriptor to make it work under the Euclidean distance.

To address the correlation problem, we proposes SVDNet, which is featured by an FC layer containing decorrelated weight vectors. We also introduce a novel three-step training scheme. In the first step, the weight matrix undergoes the singular vector decomposition (SVD) and is replaced by the product of the left unitary matrix and the singular value matrix. Second, we keep the orthogonalized weight matrix fixed and only fine-tune the remaining layers. Third, the weight matrix is unfixed and the network is trained for overall optimization. The three steps are iterated to approximate orthogonality on the weight matrix. Experimental results on three large-scale re-ID datasets demonstrate significant improvement over the baseline network, and our results are on par with the state of the art.

2. Related Work

Deep learning for person re-ID. In person re-ID task, deep learning methods can be classified into two classes: similarity learning and representation learning. The former is also called deep metric learning, in which image pairs or triplets are used as input to the network [25, 24, 1, 13, 5, 19]. In the two early works, Yi *et al*. [29] and Li *et al*. [13] use image pairs and inject part priors into the learning process. In later works, Varior *et al*. [25] incorporate long short-term memory (LSTM) modules into a siamese network. LSTMs process image parts sequentially so that the spatial connections can be memorized to enhance the discriminative ability of the deep features. Varior *et al*. [24] insert a gating function after each convolutional layer to capture effective subtle patterns between image pairs. The above-mentioned methods are effective in learning image similarities in an adaptive manner, but may have efficiency problems under large-scale galleries.

The second type of CNN-based re-ID methods focuses on feature learning, which categorizes the training samples into pre-defined classes and the FC descriptor is used for retrieval [33, 21, 26]. In [33, 34], the classification CNN model is fine-tuned using either the video frames or image bounding boxes to learn a discriminative embedding for pedestrian retrieval. Xiao *et al*. [26] propose learning generic feature representations from multiple re-ID datasets jointly. To deal with spatial misalignment, Zheng *et al*. [31] propose the PoseBox structure similar to the pictorial structure [6] to learn pose invariant embeddings. To take advantage of both the feature learning and similarity learning, Zheng *et al*. [35] and Geng *et al*. [10] combine the contrastive loss and the identification loss to improve the discriminative ability of the learned feature embedding, following the success in face verification [22]. This paper adopts the classification mode, which is shown to produce competitive accuracy without losing efficiency potentials.

PCANet and truncated SVD for CNN. We clarify the difference between SVDNet and several "look-alike" works. The PCANet [3] is proposed for image classification. It is featured by cascaded principal component analysis (PCA) filters. PCANet is related to SVDNet in that it also learns orthogonal projection directions to produce the filters. The proposed SVDNet differs from PCANet in two major aspects. First, SVDNet performs SVD on the weight matrix of CNN, while PCANet performs PCA on the raw data and feature. Second, the filters in PCANet are learned in an unsupervised manner, which does not rely on back propagation as in the case of SVDNet. In fact, SVDNet manages a stronger connection between CNN and SVD. SVDNet's parameters are learned through back propagation and decorrelated iteratively using SVD.

Truncated SVD [8, 28] is widely used for CNN model compression. SVDNet departs from it in two aspects. First, truncated SVD decomposes the weight matrix in FC layers and reconstructs it with several dominant singular vectors and values. SVDNet does not reconstruct the weight matrix but replaces it with an orthogonal matrix, which is the product of the left unitary matrix and the singular value matrix. Second, Truncated SVD reduces the model size and testing time at the cost of acceptable precision loss, while SVDNet significantly improves the retrieval accuracy without impact on the model size.

Orthogonality in the weight matrix. We note a concurrent work [27] which also aims to orthogonalize the CNN filters, yet our work is different from [27]. In [27], the regularization effect of orthogonalization benefits the back-propagation of very deep networks, thus improving the classification accuracy. The regularization proposed in [27] may not directly benefit the embedding learning process. But in this paper, orthogonalization is used to generate decorrelated descriptors suitable for retrieval. Our network

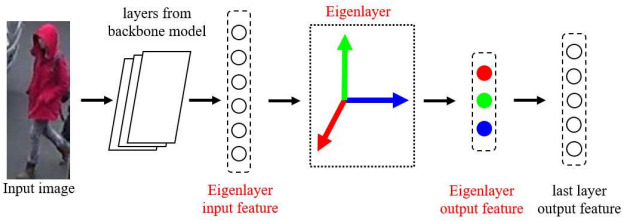


Figure 2: The architecture of SVDNet. It contains an Eigenlayer before the last FC layer of the backbone model. The weight vectors of the Eigenlayer are expected to be orthogonal. In testing, either the *Eigenlayer input feature* or the *Eigenlayer output feature* is employed for retrieval.

may not be suitable for improving classification.

3. Proposed Method

This section describes the structure of SVDNet, its training strategy, and its working mechanism.

3.1. Architecture

SVDNet mostly follows the backbone networks, *e.g.*, CaffeNet and ResNet-50. The only difference is that SVDNet uses the Eigenlayer as the second last FC layer, as shown in Fig. 2, the Eigenlayer contains an orthogonal weight matrix and is a linear layer without bias. The reason for not using bias is that the bias will disrupt the learned orthogonality. In fact, our preliminary experiments indicate that adding the ReLU activation and the bias term slightly compromises the re-ID performance, so we choose to implement the Eigenlayer based on a linear layer. The reason for positioning Eigenlayer at the second last FC layer, rather than the last one is that the model fails to converge when orthogonality is enforced on the last FC layer, which might be due to that the correlation of weight vectors in the last FC layer is determined by the training sample distribution, as explained in the introduction. During training, the input feature from a previous layer is passed through the Eigenlayer. Its inner products with the weight vectors of the Eigenlayer form the output feature, which is fully connected to the last layer of c-dim, where c denotes the number of training classes.

During testing, we extract the learned embeddings for the query and gallery images. In this step, we can use either the input or the output of Eigenlayer for feature representation, as shown in Fig. 2. Our experiment shows that using the two features can achieve similar performance, indicating that the orthogonality of Eigenlayer improves the performance of not only output but also input. The reason is a bit implicit, and we believe it originates from the back-propagation training of CNN, during which the orthogonal characteristic of weight matrix within the Eigenlayer will directly impact the characteristic of its input feature.

Algorithm 1: Training SVDNet

Input: a pre-trained CNN model, re-ID training data.
0. Add the Eigenlayer and fine-tune the network.
for $t \leftarrow 1$ *to* T **do**
  1. Decorrelation: Decompose W with SVD decomposition, and then update it: $W \leftarrow US$
  2. Restraint: Fine-tune the network with the Eigenlayer fixed
  3. Relaxation: Fine-tune the network with the Eigenlayer unfixed
end
Output: a fine-tuned CNN model, *i.e.*, SVDNet.

3.2. Training SVDNet

The algorithm of training SVDNet is presented in Alg. 1. We first briefly introduce Step 0 and then describe the restraint and relaxation Iteration (RRI) (Step 1, 2, 3).

Step 0. We first add a linear layer to the network. Then the network is fine-tuned till convergence. Note that after Step 0, the weight vectors in the linear layer are still highly correlated. In the experiment, we will present the re-ID performance of the CNN model after Step 0. Various output dimensions of the linear layer will be evaluated.

Restraint and Relaxation Iteration (RRI). It is the key procedure in training SVDNet. Three steps are involved.

- Decorrelation. We perform SVD on the weight matrix as follows:

$$W = USV^{\mathrm{T}}, \tag{1}$$

  where W is the weight matrix of the linear layer, U is the left-unitary matrix, S is the singular value matrix, and V is the right-unitary matrix. After the decomposition, we replace W with US. Then the linear layer uses all the eigenvectors of WW^{T} as weight vectors and is named as Eigenlayer.

- Restraint. The backbone model is fine-tuned till convergence, but the Eigenlayer is *fixed*.

- Relaxation. The fine-tuning goes on for some more epochs with Eigenlayer *unfixed*.

After Step 1 and Step 2, the weight vectors are orthogonal, *i.e.*, in an eigen state. But after Step 3, *i.e.*, relaxation training, W shifts away from the eigen state. So the training procedure enters another iteration t $(t = 1, \ldots, T)$ of "restraint and relaxation".

Albeit simple, the mechanism behind the method is interesting. We will try to provide insight into the mechanism in Section 3.3. During all the analysis involved, CaffeNet pre-trained on ImageNet is chosen as the backbone.

3.3. Mechanism Study

Why is SVD employed? Our key idea is to find a set of orthogonal projection directions based on what CNN has already learned from training set. Basically, for a linear layer, a set of basis in the range space of W (*i.e.*, linear subspace spanned by column vectors of W) is a potential solution. In fact, there exists numerous sets of orthogonal basis. So we decide to use the singular vectors of W as new projection directions and to weight the projection results with the corresponding singular values. That is, we replace $W = USV^{\mathrm{T}}$ with US. By doing this, the discriminative ability of feature representation over the whole sample space will be maintained. We make a mathematical proof as follows:

Given two images x_i and x_j, we denote $\vec{h_i}$ and $\vec{h_j}$ as the corresponding features before the Eigenlayer, respectively. $\vec{f_i}$ and $\vec{f_j}$ are their output features from the Eigenlayer. The Euclidean distance D_{ij} between the features of x_i and x_j is calculated by:

$$\begin{aligned} D_{ij} &= \|\vec{f_i} - \vec{f_j}\|_2 = \sqrt{(\vec{f_i} - \vec{f_j})^{\mathrm{T}}(\vec{f_i} - \vec{f_j})} \\ &= \sqrt{(\vec{h_i} - \vec{h_j})^{\mathrm{T}} W W^{\mathrm{T}} (\vec{h_i} - \vec{h_j})} \\ &= \sqrt{(\vec{h_i} - \vec{h_j})^{\mathrm{T}} U S V^{\mathrm{T}} V S^{\mathrm{T}} U^{\mathrm{T}} (\vec{h_i} - \vec{h_j})}, \end{aligned} \quad (2)$$

where U, S and V are defined in Eq. 1. Since V is a unit orthogonal matrix, Eq. 2 is equal to:

$$D_{ij} = \sqrt{(\vec{h_i} - \vec{h_j})^{\mathrm{T}} U S S^{\mathrm{T}} U^{\mathrm{T}} (\vec{h_i} - \vec{h_j})} \quad (3)$$

Eq. 3 suggests that when changing $W = USV^{\mathrm{T}}$ to US, D_{ij} remains unchanged. **Therefore, in Step 1 of Alg. 1, the discriminative ability (re-ID accuracy) of the fine-tuned CNN model is 100% preserved.**

There are some other decorrelation methods in addition to SVD. But these methods do not preserve the discriminative ability of the CNN model. To illustrate this point, we compare SVD with several competitors below.

1. Use the originally learned W (denoted by $Orig$).
2. Replace W with US (denoted by US).
3. Replace W with U (denoted by U).
4. Replace W with UV^{T} (denoted by UV^{T}).
5. Replace $W = QR$ (Q-R decomposition) with QD, where D is the diagonal matrix extracted from the upper triangle matrix R (denoted by QD).

Comparisons on Market-1501 [32] are provided in Table 1. We replace the FC layer with a 1,024-dim linear layer and fine-tune the model till convergence (Step 0 in Alg. 1). We then replace the fine-tuned W with methods 2 - 5. All the four decorrelation methods 2 - 5 update W to be an orthogonal matrix, but Table 1 indicates that only replacing W with US retains the re-ID accuracy, while the others degrade the performance.

Methods	$Orig$	US	U	UV^{T}	QD
rank-1	63.6	63.6	61.7	61.7	61.6
mAP	39.0	39.0	37.1	37.1	37.3

Table 1: Comparison of decorrelation methods in Step 1 of Alg. 1. Market-1501 and CaffeNet are used. We replace FC7 with a 1,024-dim linear layer. Rank-1 (%) and mAP (%) are shown.

When does performance improvement happen? As proven above, Step 1 in Alg. 1, *i.e.*, replacing $W = USV^{\mathrm{T}}$ with US, does not bring an immediate accuracy improvement, but keeps it unchanged. Nevertheless, after this operation, the model has been pulled away from the original fine-tuned solution, and the classification loss on the training set will increase by a certain extent. Therefore, Step 2 and Step 3 in Alg. 1 aim to fix this problem. The major effect of these two steps is to improve the discriminative ability of the input feature as well as the output feature of the Eigenlayer (Fig. 2). On the one hand, the restraint step learns the upstream and downstream layers of the Eigenlayer, which still preserves the orthogonal property. We show in Fig. 5 that this step improves the accuracy. On the other hand, the relaxation step will make the model deviate from orthogonality again, but it reaches closer to convergence. This step, as shown in Fig. 5, deteriorates the performance. But within an RRI, the overall performance improves. Interestingly, when educating children, an alternating rhythm of relaxation and restraint is also encouraged.

Correlation diagnosing. Till now, we have not provided a metric how to evaluate vector correlations. In fact, the correlation between two vectors can be estimated by the correlation coefficient. However, to the best of our knowledge, it lacks an evaluation protocol for diagnosing the *overall* correlation of a vector set. In this paper, we propose to evaluate the overall correlation as below. Given a weight matrix W, we define the gram matrix of W as,

$$\begin{aligned} G = W^{\mathrm{T}}W &= \begin{bmatrix} \vec{w_1}^{\mathrm{T}}\vec{w_1} & \vec{w_1}^{\mathrm{T}}\vec{w_2} & \cdots & \vec{w_1}^{\mathrm{T}}\vec{w_k} \\ \vec{w_2}^{\mathrm{T}}\vec{w_1} & \vec{w_2}^{\mathrm{T}}\vec{w_2} & \cdots & \vec{w_2}^{\mathrm{T}}\vec{w_k} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{w_k}^{\mathrm{T}}\vec{w_1} & \vec{w_k}^{\mathrm{T}}\vec{w_2} & \cdots & \vec{w_k}^{\mathrm{T}}\vec{w_k} \end{bmatrix} \\ &= \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1k} \\ g_{21} & g_{22} & \cdots & g_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kk} \end{bmatrix}, \end{aligned} \quad (4)$$

where k is the number of weight vectors in W (k = 4,096 in FC7 of CaffeNet), g_{ij} $(i, j = 1, ..., k)$ are the entries in W, and w_i $(i = 1, ..., k)$ are the weight vectors in W. Given W, we define $S(\cdot)$ as a metric to denote the extent of correlation between all the column vectors of W:

$$S(W) = \frac{\sum_{i=1}^{k} g_{ii}}{\sum_{i=1}^{k} \sum_{j=1}^{k} |g_{ij}|}. \quad (5)$$

From Eq. 5, we can see that the value of $S(W)$ falls within $[\frac{1}{k}, 1]$. $S(W)$ achieves the largest value 1 only when W is an orthogonal matrix, *i.e.*, $g_{ij} = 0, \text{if } i \neq j$. $S(W)$ has the smallest value $\frac{1}{k}$ when all the weight vectors are totally the same, *i.e.*, $g_{ij} = 1, \forall i, j$. So when $S(W)$ is close to $1/k$ or is very small, the weight matrix has a high correlation extent. For example, in our baseline, when directly fine-tuning a CNN model (without SVDNet training) using CaffeNet, $S(W_{\text{FC7}}) = 0.0072$, indicating that the weight vectors in the FC7 layer are highly correlated. As we will show in Section 4.5, S is an effective indicator to the convergence of SVDNet training.

Convergence Criteria for RRI. When to stop RRI is a non-trivial problem, especially in application. We employ Eq. 5 to evaluate the orthogonality of W after the relaxation step and find that $S(W)$ increases as the iteration goes on. It indicates that the correlation among the weight vectors in W is reduced step-by-step with RRI. So when $S(W)$ becomes stable, the model converges, and RRI stops. Detailed observations can be accessed in Fig. 5.

4. Experiment

4.1. Datasets and Settings

Datasets. This paper uses three datasets for evaluation, *i.e.*, **Market-1501** [32], **CUHK03** [13] and **DukeMTMC-reID** [18, 37]. The Market-1501 dataset contains 1,501 identities, 19,732 gallery images and 12,936 training images captured by 6 cameras. All the bounding boxes are generated by the DPM detector [9]. Most experiments relevant to mechanism study are carried out on Market-1501. The CUHK03 dataset contains 13,164 images of 1,467 identities. Each identity is observed by 2 cameras. CUHK03 offers both hand-labeled and DPM-detected bounding boxes, and we use the latter in this paper. For CUHK03, 20 random train/test splits are performed, and the averaged results are reported. The DukeMTMC-reID dataset is collected with 8 cameras and used for cross-camera tracking. We adopt its re-ID version benchmarked in [37]. It contains 1,404 identities (one half for training, and the other for testing), 16,522 training images, 2,228 queries, and 17,661 gallery images. For Market-1501 and DukeMTMC-reID, we use the evaluation packages provided by [32] and [37], respectively.

For performance evaluation on all the 3 datasets, we use both the Cumulative Matching Characteristics (CMC) curve and the mean Average Precision (mAP).

Backbones. We mainly use two networks pre-trained on ImageNet [7] as backbones, *i.e.*, CaffeNet [12] and ResNet-50 [11]. When using CaffeNet as the backbone, we directly replace the original FC7 layer with the Eigenlayer, in case that one might argue that the performance gain is brought by deeper architecture. When using ResNet-50 as the backbone, we have to insert the Eigenlayer before the last FC layer because ResNet has no hidden FC layer and the influence of adding a layer into a 50-layer architecture can be neglected. In several experiments on Market-1501, we additionally use VGGNet [20] and a Tiny CaffeNet as backbones to demonstrate the effectiveness of SVDNet on different architectures. The Tiny CaffeNet is generated by reducing the FC6 and FC7 layers of CaffeNet to containing 1024 and 512 dimensions, respectively.

4.2. Implementation Details

Baseline. Following the practice in [33], baselines using CaffeNet and ResNet-50 are fine-tuned with the default parameter settings except that the output dimension of the last FC layer is set to the number of training identities. The CaffeNet Baseline is trained for 60 epochs with a learning rate of 0.001 and then for another 20 epochs with a learning rate of 0.0001. The ResNet Baseline is trained for 60 epochs with learning rate initialized at 0.001 and reduced by 10 on 25 and 50 epochs. During testing, the FC6 or FC7 descriptor of CaffeNet and the Pool5 or FC descriptor of ResNet-50 are used for feature representation.

On Market-1501, CaffeNet and Resnet-50 achieves rank-1 accuracy of 55.3% (73.8%) with the FC6 (Pool5) descriptor, which is consistent with the results in [33].

Detailed settings. CaffeNet-backboned SVDNet takes 25 RRIs to reach final convergence. For both the restraint stage and the relaxation stage within each RRI except the last one, we use 2000 iterations and fix the learning rate at 0.001. For the last restraint training, we use 5000 iterations (learning rate 0.001) + 3000 iterations (learning rate 0.0001). The batch size is set to 64. ResNet-backboned SVDNet takes 7 RRIs to reach final convergence. For both the restraint stage and the relaxation stage within each RRI, we use 8000 iterations and divide the learning rate by 10 after 5000 iterations. The initial learning rate for the 1st to the 3rd RRI is set to 0.001, and the initial learning rate for the rest RRIs is set to 0.0001. The batch size is set to 32.

The output dimension of Eigenlayer is set to be 1024 in all models, yet the influence of this hyper-parameter is to be analyzed in Section 4.4. The reason of using different times of RRIs for different backbones is to be illustrated in Section 4.5.

Models & Features	dim	Market-1501				CUHK03				DukeMTMC-reID			
		R-1	R-5	R-10	mAP	R-1	R-5	R-10	mAP	R-1	R-5	R-10	mAP
Baseline(C) FC6	4096	55.3	75.8	81.9	30.4	38.6	66.4	76.8	45.0	46.9	63.2	69.2	28.3
Baseline(C) FC7	4096	54.6	75.5	81.3	30.3	42.2	70.2	80.4	48.6	45.9	62.0	69.7	27.1
SVDNet(C) FC6	4096	**80.5**	**91.7**	**94.7**	**55.9**	**68.5**	**90.2**	**95.0**	**73.3**	**67.6**	**80.5**	**85.7**	**45.8**
SVDNet(C) FC7	1024	79.0	91.3	94.2	54.6	66.0	89.4	93.8	71.1	66.7	80.5	85.1	44.4
Baseline(R) Pool5	2048	73.8	87.6	91.3	47.9	66.2	87.2	93.2	71.1	65.5	78.5	82.5	44.1
Baseline(R) FC	N	71.1	85.0	90.0	46.0	64.6	89.4	95.0	70.0	60.6	76.0	80.9	40.4
SVDNet(R) Pool5	2048	**82.3**	**92.3**	**95.2**	**62.1**	**81.8**	**95.2**	97.2	**84.8**	**76.7**	**86.4**	**89.9**	**56.8**
SVDNet(R) FC	1024	81.4	91.9	94.5	61.2	81.2	95.2	**98.2**	84.5	75.9	86.4	89.5	56.3

Table 2: Comparison of the proposed method with baselines. C: CaffeNet. R: ResNet-50. In ResNet Baseline, "FC" denotes the last FC layer, and its output dimension N changes with the number of training identities, *i.e.*, 751 on Market-1501, 1,160 on CUHK03 and 702 on DukeMTMC-reID. For SVDNet based on ResNet, the Eigenlayer is denoted by "FC", and its output dimension is set to 1,024.

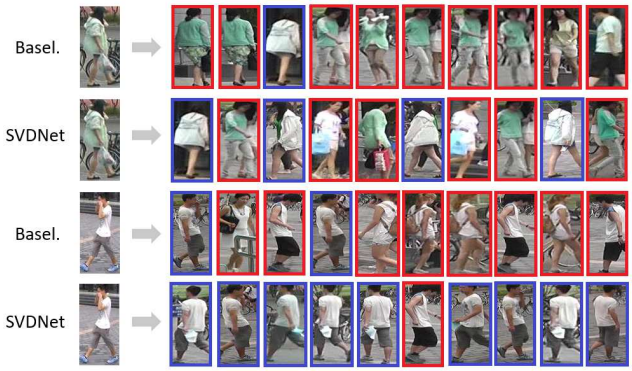


Figure 3: Sample retrieval results on Market-1501. In each row, images are arranged in descending order according to their similarities with the query on the left. The true and false matches are in the blue and red boxes, respectively.

4.3. Performance Evaluation

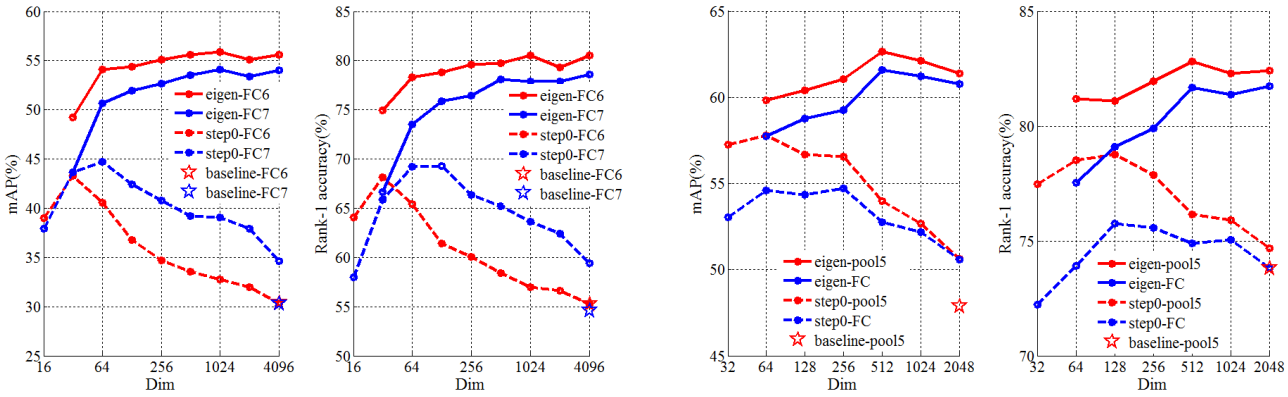
The effectiveness of SVDNet. We comprehensively evaluate the proposed SVDNet on all the three re-ID benchmarks. The overall results are shown in Table 2.

The improvements achieved on both backbones are significant: When using CaffeNet as the backbone, the Rank-1 accuracy on Market-1501 rises from 55.3% to 80.5%, and the mAP rises from 30.4% to 55.9%. On CUHK03 (DukeMTMC-reID) dataset, the Rank-1 accuracy rises by +26.3% (+20.7%), and the mAP rises by +24.7% (+17.5%). When using ResNet as the backbone, the Rank-1 accuracy rises by +8.4%, +15.6% and +11.2% respectively on Market-1501, CUHK03 and DukeMTMC-reID dataset. The mAP rises by +14.2%, +13.7% and +12.7% correspondingly. Some retrieval examples on Market-1501 are shown in Fig. 3.

Comparison with state of the art. We compare SVDNet with the state-of-the-art methods. Comparisons on Market-1501 and CUHK03 are shown in Table 3. Comparing with already published papers, SVDNet achieves competitive performance. We report **rank-1 = 82.3%, mAP = 62.1% on Market-1501, and rank-1 = 81.8%, mAP = 84.8% on CUHK03**. The re-ranking method [38] is higher than ours in mAP on Market-1501, because re-ranking exploits the relationship among the gallery images and results in a high recall. We speculate that this re-ranking method will also bring improvement for SVDNet. Comparing with

Methods	Market-1501		CUHK03	
	rank-1	mAP	rank-1	mAP
LOMO+XQDA[14]	43.8	22.2	44.6	51.5
CAN[16]	48.2	24.4	63.1	-
SCSP[4]	51.9	26.4	-	-
Null Space[30]	55.4	29.9	54.7	-
DNS[30]	61.0	35.6	54.7	-
LSTM Siamese[25]	61.6	35.3	57.3	46.3
MLAPG[15]	-	-	58.0	-
Gated SCNN[24]	65.9	39.6	61.8	51.3
ReRank (C) [38]	61.3	46.8	58.5	64.7
ReRank (R) [38]	77.1	63.6	64.0	69.3
PIE (A)* [31]	65.7	41.1	62.6	67.9
PIE (R)* [31]	79.3	56.0	67.1	71.3
SOMAnet (VGG)* [2]	73.9	47.9	72.4	-
DLCE (C)* [35]	62.1	39.6	59.8	65.8
DLCE (R)* [35]	79.5	59.9	83.4	86.4
Transfer (G)* [10]	83.7	65.5	84.1	-
SVDNet(C)	80.5	55.9	68.5	73.3
SVDNet(R,1024-dim)	82.3	62.1	81.8	84.8

Table 3: Comparison with state of the art on Market-1501 (single query) and CUHK03. * denotes unpublished papers. Base networks are annotated. C: CaffeNet, R: ResNet-50, A: AlexNet, G: GoogleNet [23]. The best, second and third highest results are in blue, red and green, respectively.



(a) CaffeNet-backboned SVDNet (b) ResNet-backboned SVDNet

Figure 4: Dimension comparison on (a) CaffeNet-backboned and (b) ResNet-backboned. The marker prefixed by "step0" denotes that the corresponding model is trained without any RRI. The marker prefixed by "eigen" denotes that the corresponding model is trained with sufficient RRIs to final convergence. For (a), the output dimension of Eigenlayer is set to 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096. For (b), the output dimension of Eigenlayer is set to 32, 64, 128, 256, 512, 1024 and 2048.

Methods	DukeMTMC-reID		CUHK03-NP	
	rank-1	mAP	rank-1	mAP
BoW+kissme [32]	25.1	12.2	6.4	6.4
LOMO+XQDA [14]	30.8	17.0	12.8	11.5
Baseline (R)	65.5	44.1	21.3	19.7
GAN (R) [37]	67.7	47.1	-	-
PAN (R) [36]	71.6	51.5	36.3	34.0
SVDNet (C)	67.6	45.8	27.7	24.9
SVDNet (R)	**76.7**	**56.8**	**41.5**	**37.3**

Table 4: Comparison with the state of the art on DukeMTMC-reID and CUHK03-NP. Rank-1 accuracy (%) and mAP (%) are shown. For fair comparison, all the results are maintained without post-processing methods.

the unpublished Arxiv papers, (some of) our numbers are slightly lower than [10] and [35]. Both works [10] and [35] combine the verification and classification losses, and we will investigate into integrating this strategy into SVDNet.

Moreover, the performance of SVDNet based on relatively simple CNN architecture is impressive. On Market-1501, CaffeNet-backboned SVDNet achieves 80.5% rank-1 accuracy and 55.9% mAP, exceeding other CaffeNet-based methods by a large margin. Additionally, using VGGNet and Tiny CaffeNet as backbone achieves 79.7% and 77.4% rank-1 accuracy respectively. On CUHK03, CaffeNet-backboned SVDNet even exceeds some ResNet-based competing methods except DLCE(R). This observation suggests that our method can achieve acceptable performance with high computing effectiveness.

In Table 4, comparisons on DukeMTMC-reID and CUHK03 under a new training/testing protocol (denoted as CUHK03-NP) raised by [38] are summarized. Relatively fewer results are reported because both DukeMTMC-reID and CUHK03-NP have only been recently benchmarked. On DukeMTMC-reID, this paper reports **rank-1 = 76.7%, mAP = 56.8%**, which is higher than the several competing methods including a recent GAN approach [37]. On CUHK03-NP, this paper reports **rank-1 = 41.5%, mAP = 37.3%**, which is also the highest among all the methods.

4.4. Impact of Output Dimension

We vary the dimension of the output of Eigenlayer. Results of CaffeNet and ResNet-50 are drawn in Fig. 4.

When trained without RRI, the model has no intrinsic difference with a baseline model. It can be observed that the output dimension of the penultimate layer significantly influences the performance. As the output dimension increases, the re-ID performance first increases, reaches a peak and then drops quickly. In this scenario, we find that lowering the dimension is usually beneficial, probably due to the reduced redundancy in filters of FC layer.

The influence of the output dimension on the final performance of SVDNet presents another trend. As the output dimension increases, the performance gradually increases until reaching a stable level, which suggests that our method is immune to harmful redundancy.

4.5. RRI Boosting Procedure

This experiment reveals how the re-ID performance changes after each restraint step and each relaxation step, and how SVDNet reaches the stable performance step by step. In our experiment, we use 25 epochs for both the re-

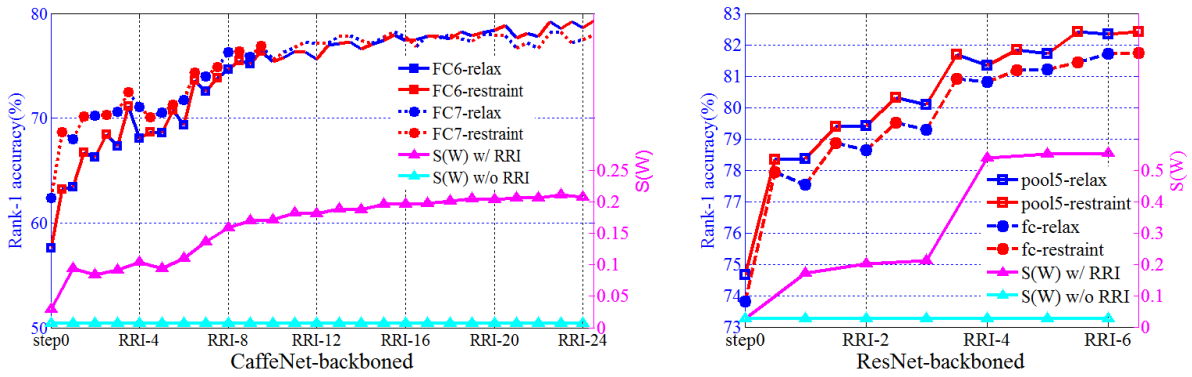


Figure 5: Rank-1 accuracy and $S(W)$ (Eq. 5) of each intermediate model during RRI. Numbers on the horizontal axis denote the end of each RRI. SVDNet based on CaffeNet and ResNet-50 take about 25 and 7 RRIs to converge, respectively. Results before the $11th$ RRI is marked. $S(W)$ of models trained without RRI is also plotted for comparison.

Methods	$Orig$	US	U	UV^{T}	QD
FC6(C)	57.0	80.5	76.2	57.4	58.8
FC7(C)	63.6	79.0	75.8	62.7	63.2
Pool5(R)	75.9	82.3	80.9	76.5	77.9
FC(R)	75.1	81.4	80.2	74.8	77.3

Table 5: Comparison of the decorrelation methods specified in Section 3.3. Rank-1 accuracy (%) on Market-1501 is shown. Dimension of output feature of Eigenlayer is set to 1024. We run sufficient RRIs for each method.

straint phase and the relaxation phase in one RRI. The output dimension of Eigenlayer is set to 2,048. Exhaustively, we test re-ID performance and $S(W)$ values of all the intermediate CNN models. We also increase the training epochs of baseline models to be equivalent of training SVDNet, to compare $S(W)$ of models trained with and without RRI. Results are shown in Fig. 5, from which four conclusions can be drawn.

First, within each RRI, rank-1 accuracy takes on a pattern of "increase and decrease" echoing the restraint and relaxation steps: When W is fixed to maintain orthogonality during restraint training, the performance increases, implying a boosting in the discriminative ability of the learned feature. Then during relaxation training, W is unfixed, and the performance stagnates or even decreases slightly. Second, as the RRI goes, the overall accuracy increases, and reaches a stable level when the model converges. Third, it is reliable to use $S(W)$ – the degree of orthogonality – as the convergence criteria for RRI. During RRI training, $S(W)$ gradually increases until reaching stability, while without RRI training, $S(W)$ fluctuates slightly around a relatively low value, indicating high correlation among weight vectors. Fourth, ResNet-backboned SVDNet needs much fewer RRIs to converge than CaffeNet-backboned SVDNet.

4.6. Comparison of Decorrelation Methods

In Section 3.3, several decorrelation methods are introduced. We show that only the proposed method of replacing W with US maintains the discriminative ability of the output feature of Eigenlayer, while all the other three methods lead to performance degradation to some extent. Here, we report their final performance when RRI training is used.

Results on Market-1501 are shown in Table 5. It can be observed that the proposed decorrelating method, *i.e.*, replacing W with US, achieves the highest performance, followed by the "U", "QD" and "UV^{T}" methods. In fact, the "UV^{T}" method does not bring about observable improvement compared with "$Orig$". This experiment demonstrates that not only the orthogonality itself, but also the decorrelation approach, are vital for SVDNet.

5. Conclusions

In this paper, SVDNet is proposed for representation learning in pedestrian retrieval, or re-identification. Decorrelation is enforced among the projection vectors in the weight matrix of the FC layer. Through iterations of "restraint and relaxation", the extent of vector correlation is gradually reduced. In this process, the re-ID performance undergoes iterative "increase and decrease", and finally reaches a stable accuracy. Due to elimination of correlation of the weight vectors, the learned embedding better suits the retrieval task under the Euclidean distance. Significant performance improvement is achieved on the Market-1501, CUHK03, and DukeMTMC-reID datasets, and the re-ID accuracy is competitive with the state of the art.

In the future study, we will investigate more extensions of SVDNet to find out more about its working mechanism. We will also apply SVDNet on the generic instance retrieval problem.

References

[1] E. Ahmed, M. J. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *CVPR*, 2015. 2

[2] 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 preprint arXiv:1701.03153*, 2017. 6

[3] T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. Pcanet: A simple deep learning baseline for image classification? *IEEE Trans. Image Processing*, 24(12):5017–5032, 2015. 2

[4] D. Chen, Z. Yuan, B. Chen, and N. Zheng. Similarity learning with spatial constraints for person re-identification. In *CVPR*, 2016. 6

[5] 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 *CVPR*, 2016. 2

[6] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom pictorial structures for re-identification. In *BMVC*, 2011. 2

[7] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[8] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*, 2014. 2

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 5

[10] M. Geng, Y. Wang, T. Xiang, and Y. Tian. Deep transfer learning for person re-identification. *arXiv preprint arXiv:1611.05244*, 2016. 1, 2, 6, 7

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 5

[13] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014. 2, 5

[14] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, 2015. 6, 7

[15] S. Liao and S. Z. Li. Efficient PSD constrained asymmetric metric learning for person re-identification. In *ICCV*, 2015. 6

[16] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan. End-to-end comparative attention networks for person re-identification. *arXiv preprint arXiv:1606.04404*, 2016. 6

[17] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV*, 2016. 1

[18] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016. 5

[19] H. Shi, Y. Yang, X. Zhu, S. Liao, Z. Lei, W. Zheng, and S. Z. Li. Embedding deep metric for person re-identification: A study against large variations. In *ECCV*, 2016. 2

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[21] C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian. Deep attributes driven multi-camera person re-identification. In *ECCV*, 2016. 2

[22] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *NIPS*, 2014. 2

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 6

[24] R. R. Varior, M. Haloi, and G. Wang. Gated siamese convolutional neural network architecture for human re-identification. In *ECCV*, 2016. 2, 6

[25] 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*, 2016. 2, 6

[26] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016. 1, 2

[27] D. Xie, J. Xiong, and S. Pu. All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation. In *CVPR*, 2017. 2

[28] J. Xue, J. Li, and Y. Gong. Restructuring of deep neural network acoustic models with singular value decomposition. In *Interspeech*, 2013. 2

[29] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Deep metric learning for person re-identification. In *ICPR*, 2014. 2

[30] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *CVPR*, 2016. 6

[31] L. Zheng, Y. Huang, H. Lu, and Y. Yang. Pose invariant embedding for deep person re-identification. *arXiv preprint arXiv:1701.07732*, 2017. 1, 2, 6

[32] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015. 4, 5, 7

[33] L. Zheng, Y. Yang, and A. G. Hauptmann. Person re-identification: Past, present and future. *arXiv preprint arXiv:1610.02984*, 2016. 1, 2, 5

[34] L. Zheng, H. Zhang, S. Sun, M. Chandraker, and Q. Tian. Person re-identification in the wild. In *CVPR*, 2017. 2

[35] Z. Zheng, L. Zheng, and Y. Yang. A discriminatively learned CNN embedding for person re-identification. *arXiv preprint arXiv:1611.05666*, 2016. 2, 6, 7

[36] Z. Zheng, L. Zheng, and Y. Yang. Pedestrian alignment network for large-scale person re-identification. *arXiv preprint arXiv:1707.00408*, 2017. 7

[37] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. *arXiv preprint arXiv:1701.07717*, 2017. 5, 7

[38] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*, 2017. 6, 7