SVDNet for Pedestrian Retrieval

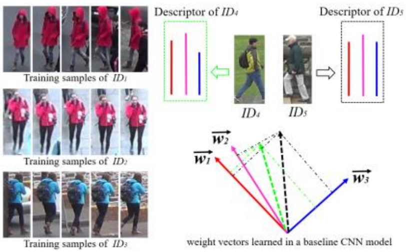
Yifan Sun[1] Liang Zheng[2] Weijian Deng[3] Shengjin Wang[1]

[1]Tsinghua University [2]University of Technology Sydney [3]University of Chinese Academy of Sciences

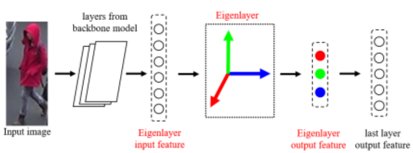
◆ An attempt to interpret CNN

➢ **An interpretation of CNN weight vectors:** we view each weight vector within a layer as a projection basis, as well as an exemplar in the embedded feature space of input layer.

➢ **Motivation**: correlation among weight vectors indicates redundancy between exemplars, and may compromise the learned feature representation for pedestrian retrieval (as well as for some other tasks).



➢ **Goal:** uncorrelated weight vectors
independent exemplars
enhanced discriminative ability



➢ SVDNet 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.

◆ SVD helps understanding CNN

➢ **SVD says Yes** to the question: Once CNN learns a set of projection basis, i.e., weight vectors for a certain layer, which projection direction does CNN consider most important? Can this set of weight vectors be replaced by an orthogonal weight matrix with the learned discriminative ability maintained?

$$D_{ij} = \|\vec{f}_i - \vec{f}_j\|_2 = \sqrt{(\vec{f}_i - \vec{f}_j)^T(\vec{f}_i - \vec{f}_j)}$$

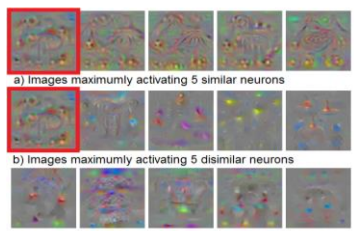
$$= \sqrt{(\vec{h}_i - \vec{h}_j)^T WW^T(\vec{h}_i - \vec{h}_j)}$$

$$= \sqrt{(\vec{h}_i - \vec{h}_j)^T USV^TVS^TU^T(\vec{h}_i - \vec{h}_j)}$$

$$\downarrow$$

$$\sqrt{(\vec{h}_i - \vec{h}_j)^T USS^TU^T(\vec{h}_i - \vec{h}_j)}$$

➢ **Decorrelated weight vectors, dissimilar exemplars**

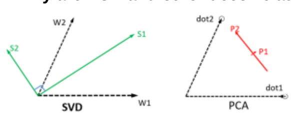


a) Images maximumly activating 5 similar neurons

b) Images maximumly activating 5 disimilar neurons

c) Images maximumly activating neurons after our method

➢ **Why are PCA and other decorrelating methods inferior?**



PCA ignores the most important P1 projection direction.

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

Other methods degrade the performance.

◆ Train CNN like educating children

➢ **Restraint and Relaxation Iteration training**

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**

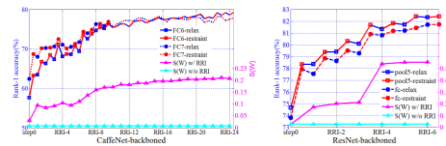
  1. Decorrelation: Decompose W with SVD decomposition, and then update it: $W \leftarrow US$

  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.

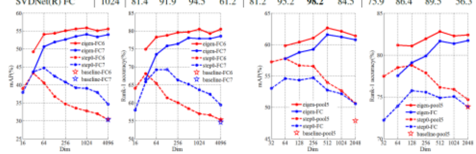


“increase – stagnate” echoing “Restraint – Relaxation”.

When educating children, a similar rhythm is encouraged!

➢ **Performance**

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



(a) CaffeNet-backboned SVDNet

(b) ResNet-backboned SVDNet

➢ Higher performance is achieved with data augmentation, e.g., 87% Rank-1 on market-1501

➢ SVDNet may be extended to some other computer vision tasks. SVDNet based on resnet-20 achieves 93.5% (+1.7%) top-1 accuracy on Cifar-10 dataset.