Learning Discriminative Features with Class Encoder

Hailin Shi, Xiangyu Zhu, Zhen Lei, Shengcai Liao, Stan Z. Li
Center for Biometrics and Security Research & National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun Donglu, Beijing 100190, China
{hailin.shi, xiangyu.zhu, zlei, scliao, szli}@nlpr.ia.ac.cn

Abstract

Deep neural networks usually benefit from unsupervised pre-training, e.g. auto-encoders. However, the classifier further needs supervised fine-tuning methods for good discrimination. Besides, due to the limits of full-connection, the application of auto-encoders is usually limited to small, well aligned images. In this paper, we incorporate the supervised information to propose a novel formulation, namely class-encoder, whose training objective is to reconstruct a sample from another one of which the labels are identical. Class-encoder aims to minimize the intra-class variations in the feature space, and to learn a good discriminative manifolds on a class scale. We impose the class-encoder as a constraint into the softmax for better supervised training, and extend the reconstruction on feature-level to tackle the parameter size issue and translation issue. The experiments show that the class-encoder helps to improve the performance on benchmarks of classification and face recognition. This could also be a promising direction for fast training of face recognition models.

1. Introduction

In recent years, many learning algorithms, e.g. Restricted Boltzmann Machine (RBM) [6] and auto-encoder (AE) [2], proposed to pre-train the neural network by auto-reconstruction in a layer-wise way and achieved breakthroughs on training problems. This sort of algorithms, to which we refer as *reconstructive* methods, constitute an important subset of deep learning approaches nowadays. More recently, along this direction, certain variants of AE, such as denoising auto-encoder (DAE) [22, 23] and contractive auto-encoder (CAE) [16], referred to as regularized AEs [1], are proposed to estimate data-generating distribution on a local scale and learn compact low-dimensional manifolds, in which better discrimination power can be expected.

On the other hand, convolutional neural networks (CNN) [12] is also a widely-used approach of deep learning towards computer vision. In recent years, the computational resources have been massively improved by GPU implementations [11, 10] and distributed computing clusters [4], and various large-scale data sets have been collected to satisfy the training. Due to these benefits, CNNs demonstrated the power of hierarchical representation by beating the hand-craft features, and won many contests in this field [11, 17, 5, 20].

Problems. Firstly, RBM, AE and their variants are unsupervised methods. To bring about good discrimination, the classifier needs supervised training. In other words, good representation from reconstruction does not guarantee good classification [1]. This suggests to find an objective with both reconstructive and discriminative aspects to improve the training.

Secondly, the auto-encoders are not robust to image translation; in addition, they often keep a large number of parameters that increase explosively according to the data size. As a result, the application of AE is usually limited to small, well aligned images.

Contribution. Firstly, we propose a supervised reconstructive model, referred to as *class-encoder*, whose objective is the reconstruction of one sample from another within the same class. The model minimizes the intra-class variations and learns compact low-dimensional manifolds on a class scale. Although class-encoder method is similar to AE, its application is not in the pre-training. Class-encoder is directly used in the supervised training of network, as it is a supervised method. We further imposed the class-encoder as a constraint into the softmax classifier (namely Class-Encoding Classifier, CEC) and achieve better performance than the pure softmax.

Secondly, we propose a deep hybrid neural network that combines the CNN and the CEC, so to let them benefit from each other. The convolutional layers extract features from data at the bottom level, and the CEC is disposed at the top level. Different from former reconstructive models which directly reconstructs data, in this framework, the intra-class reconstruction is performed on the feature-level. So, the

CEC is robust to translation due to the CNN, and CNN has better generalization thanks to the CEC. Besides, the size of fully-connected (FC) layer and its parameter number are limited in an acceptable range, because the reconstructive target is not images but feature vectors. We use this network to learn robust and discriminative features for face recognition.

2. Related work

Regularized auto-encoders. DAE and CAE locally estimates data-generating distribution and captures local manifold structure. Their pre-training is based on unsupervised method. By contrast, class-encoder extends them to a supervised style.

FIP feature. Zhu et al. [26] proposed to learn face identity-preserving (FIP) features through recovering frontal face images from other views. Another work [27] employed a similar method which trained multiple deep networks on the facial components of recovered frontal face. Comparing with class-encoder, their training objective is strictly fixed by canonical view. Therefore, the selection of canonical view is indispensable. Besides, their reconstruction is performed on data-level, not feature-level. Thus, the performance is very limited by data condition, i.e. facial expression, image cropping (background interference), alignment etc. The feature-level reconstruction of class-encoder is crucial for the elimination of nuisance factors.

3. The proposed method

In this section, we begin with class-encoder. Then, we introduce the CEC model. Finally, we describe the Deep CEC.

3.1. Class-encoder

Class-encoder and auto-encoder share the same architecture (Fig. 1) which includes an input layer, a hidden layer (encoder) and an output layer (decoder) of full-connection. The training objective is the main difference between class-encoder and auto-encoder. Auto-encoder aims to reconstruct a data sample from itself, while class-encoder performs the reconstruction of one sample from another one with the same label.

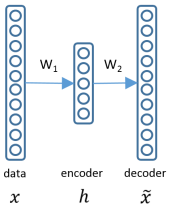

Figure 1. Class-encoder network with single layer of encoder and decoder.

Formulation. Let x be an input data, h be the activation of the hidden layer, $\tilde{x}$ be the reconstruction, W_1 and W_2 be the weight matrices of the FC layers. W_1 and W_2 often take form of tied weights, ie. $W_1^T = W_2$, which is usually employed as an implicit regularization for preventing extremely large and small entries. For the simplicity, we merge the bias term into the weight matrices in this paper. Then, the reconstruction $\tilde{x}$ is calculated as follows:

$$h = f(W_1 x) \tag{1}$$

$$\tilde{x} = f(W_2 h) = f(W_2 f(W_1 x)) \tag{2}$$

where $f(\cdot)$ is the activation function. To achieve intra-class reconstruction, let $\hat{x}$ be any data sample that has the same label with x. Therefore, the objective function of class-encoder is defined as

$$Cost_{ce} = \frac{1}{2N} \sum_{x \in \mathbf{X}} \sum_{\hat{x} \in \mathbf{S}_x} \| \tilde{x} - \hat{x} \|^2 \tag{3}$$

where N denotes the total number of training data, $\mathbf{X}$ denotes the entire training data set, and $\mathbf{S}_x$ denotes the subset of the class in which x is found. Supposing there are $\mathbf{C}$ classes in total, let $c = 1, 2, ..., \mathbf{C}$ be the class labels, and S_c be the subset of c^{th} class with size of N_c. Then, Eq. 3 can be developed as follows:

$$\begin{aligned} Cost_{ce} &= \frac{1}{2} \sum_{c=1}^{\mathbf{C}} \frac{1}{N_c} \sum_{x \in S_c} \sum_{\hat{x} \in S_c} \| \tilde{x} - \hat{x} \|^2 \\ &= \frac{1}{2} \sum_{c=1}^{\mathbf{C}} \sum_{x \in S_c} \frac{1}{N_c} \sum_{\hat{x} \in S_c} (\| \tilde{x} \|^2 + \| \hat{x} \|^2 - 2\tilde{x}^T \hat{x}) \\ &= \frac{1}{2} \sum_{c=1}^{\mathbf{C}} \sum_{x \in S_c} (\frac{1}{N_c} \| \tilde{x} \|^2 + \frac{1}{N_c} \sum_{\hat{x} \in S_c} \| \hat{x} \|^2 \\ &\qquad - 2\tilde{x}^T (\frac{1}{N_c} \sum_{\hat{x} \in S_c} \hat{x})). \end{aligned} \tag{4}$$

In Eq. 4, the first term is regarded as a penalty of magnitude of the reconstruction; the second term is constant; the third term indicates that class-encoder's reconstruction $\tilde{x}$ is prone to have small angle with the mean vector of the corresponding class. Hence, class-encoder tends to maximize a cosine-similarity-like metric between the reconstructions and intra-class means.

It is a supervised learning task which implicitly minimizes the intra-class variation. The model a learns discriminative low-dimensional manifold on a class scale in the decoder space. Data points are projected into a dense distribution within each class, whose center is located at the intra-class mean. Considering Eq. 1, this intra-class convergency also takes place in the hidden layer h (i.e. encoder space). It will be proved empirically in the next section.

3.2. CEC model

To make use of the advantage that class-encoder minimizes the intra-class variation, we impose the class-encoder into the softmax classifier, and train the network with the intra-class reconstruction and softmax regression jointly, in order to potentiate the discrimination.

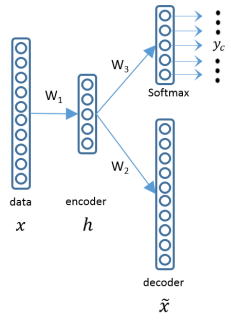

Figure 2. CEC model. We train class-encoder and softmax simultaneously. During the test of classification, we ignore the decoder and only take account of the output of softmax.

Fig. 2 shows the model of CEC. The objective function is the weighted sum of each part,

$$Cost_{cec} = Cost_{softmax} + \lambda Cost_{ce}. \tag{5}$$

The second term in Eq. 5 represents the weighted cost from class-encoder. It has the same definition with Eq. 3

Denote the ground-truth class label y_c corresponding to W_3^c the cth row of W_3. The cost of softmax is formulated as

$$Cost_{softmax} = \frac{exp(W_3^c h)}{\sum_{l=1}^{\mathbf{C}} exp(W_3^l h)}. \tag{6}$$

The softmax aims to maximize the log-likelihood of weight matrix W_3,

$$\begin{aligned}\ell(W_3; h, y) &= \sum_{c=1}^{\mathbf{C}} \frac{1}{N_c} \sum_{x \in S_c} logP(y = y_c, h \mid W_3) \\ &= \sum_{c=1}^{\mathbf{C}} \frac{1}{N_c} \sum_{x \in S_c} logP(h \mid y = y_c, W_3) P(y_c).\end{aligned} \tag{7}$$

We assume that the conditional probability $P(h \mid y = y_c, W_3)$ follows the Gaussian distribution,

$$h \mid y = y_c, W_3 \quad \sim \quad \mathcal{N}(\mu, \sigma). \tag{8}$$

The h is a variable, not fixed. It is also natural to assume the a priori of h in the class y_c follows the Gaussian distribution,

$$h \mid y_c \quad \sim \quad \mathcal{N}(\mu_h, \sigma_h). \tag{9}$$

From an optimized softmax classifier, we can find either $\mu = \mu_h$ or the two mean vectors are very close. In addition, due to the effect of class-encoder, σ_h is small. Thus, softmax has a very large probability to have h close to μ, which leads to a large value of $P(h \mid y = y_c, W_3)$ and so the log-likelihood in Eq. 7. In other words, the class-encoder improves the lower-bound of the likelihood of softmax. Sharper distribution $P(h \mid y_c)$ we sample from, more possibly we obtain large value of likelihood.

3.3. Deep CEC and Feature-level Strategy

Deep CEC (DCEC) is built by cascading CNN module and the CEC (Fig. 3). Like conventional CNNs, the CNN module is composed by convolutional and max-pooling layers.

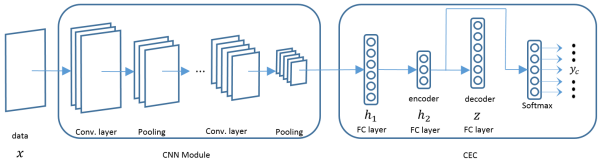

Figure 3. Overview of DCEC. The network is built by cascading the CNN module and the CEC.

The CEC receives the features from the CNN module, and works like the above-mentioned CEC except that the decoder aims to reconstruct the feature rather than the raw data. Here, we notate the input data pair as $\{x', x''\}$. Let h_1, h_2, and z be the activations of the first layer, encoder, and decoder in the CEC, respectively. The training objective is defined as

$$Cost_{ce}^{feature} = \frac{1}{2N} \sum_{x' \in \mathbf{X}} \sum_{x'' \in \mathbf{S}_{x'}} \parallel z' - h_1'' \parallel^2. \tag{10}$$

Note that z' and h_1'' come from the input data pair $\{x', x''\}$, not from a single sample. In the practical training, x' and x'' are sampled from a class, and input to the DCEC in sequence, to compute z' and h_1'', respectively.

Here, the objective of class-encoder is to reconstruct the features (ie. h_1''). We refer to this kind of reconstruction as *feature-level*, in contrast to the data-level reconstruction. There are two reasons behind the feature-level reconstruction.

First, the images may contain not only the target object, but nuisance factors as well, such as background, facial expression, poses etc. Simply reconstructing the intra-class images will introduce substantial noise to the training, whereas the feature-level reconstruction can eliminate the nuisance factors in the feature space. This is because the input of CEC is no longer fixed data, but feature variables. The CEC estimates the joint probability

$$P(z' = h_1'', y_c). \tag{11}$$

Let $z' = \{z'_n, z'_d\}$ and $h''_1 = \{h''_{1n}, h''_{1d}\}$, in which the subscripts n and d denote the nuisance and discriminative factors, respectively. We assume that the nuisance and discriminative factors are independently distributed. Therefore, the joint probability can be written as

$$P(z'_d = h''_{1d} \mid y_c)P(z'_n = h''_{1n} \mid y_c)P(y_c). \quad (12)$$

In Eq. 12, the first conditional probability is maximized by softmax. We marginalize the latter terms

$$\sum_{y_c} P(z'_n = h''_{1n} \mid y_c)P(y_c) = P(z'_n = h''_{1n}) \quad (13)$$

which leads to a very low value, since the nuisance factors are very likely different (eg. background in different images could seldom be the same). Thus, the proportion of nuisance factor is reduced in the feature space. From another point of view (ie. the previous interpretation of convergency), the intra-class features converge to the corresponding discriminative factor.

Second, the target object may present at different locations in images. Without alignment, the data-level reconstruction will introduce the noise too. Owing to the CNN module, the extracted feature is robust to image translation, and so is the feature-level reconstruction.

The objective function of DCEC is the weighted sum of softmax and intra-class, feature-level reconstruction,

$$Cost_{dcec} = Cost_{softmax} + \lambda Cost_{ce}^{feature}. \quad (14)$$

By BP method, the CNN module and the CEC are trained simultaneously.

4. Experiments

In this section, we report the experiments of the proposed methods. We started with the pure class-encoder. Then, we extended the experiment to CEC. Finally, we applied DCEC to learn robust features for human face recognition.

4.1. Inspection of class-encoder

In this subsection, we trained a network of pure class-encoder, in order to give an intuitive show of class-encoder's ability of discrimination in the feature space.

Data. MNIST [12] is a general database of handwritten digits, containing a training set of 50,000 samples, a validation set of 10,000 samples, and a test set of 10,000 samples. The 10 digits own roughly equal number of samples.

Setting. To achieve good convergency, we built a 4-layer encoder and a symmetrical decoder. The number of nodes for encoder were 2000-1000-500-250, determined by referring to the architecture in Hinton et al. [6]. Since the data had been well aligned and keep mono-black background, we let the reconstruction to be on data-level. The network was randomly initialized. We randomly selected 15,000 pairs for each digit. Each pair was fed to the network consequently to calculate the reconstruction cost.

Result. The network was optimized by stochastic gradient descent (SGD) and BP method. We extracted the activation values of the middle layer (250-dimensional) and reduced its dimensionality to 2 by PCA. We show the scatters in Fig 4. Along with the training process, each class converged effectively. In Fig. 5, we show more attempts on different architectures. The scatters suggest that deeper and wider architectures give better results.

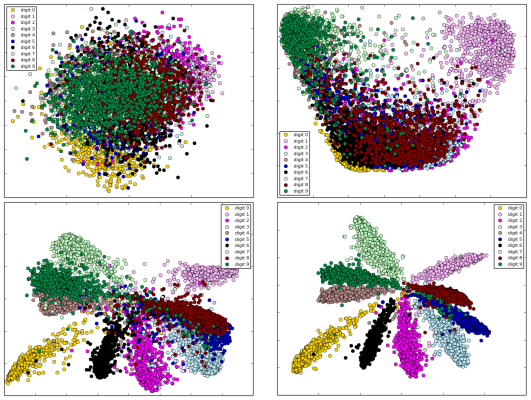

Figure 4. From left to right, top to bottom: scatters of the middle-layer activation of the class-encoder network along with the training epoch 0, 10, 50 and 200. We assign each digit a distinct color. Best view in color.

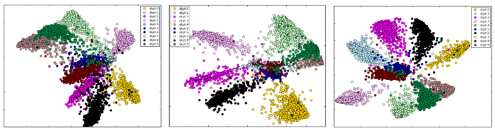

Figure 5. From left to right, the corresponding architectures of encoder are 200-200-200, 1500-1000-500, and 200-200-200-200, respectively.

4.2. CEC for classification

In this subsection, we evaluated the CEC for classification.

Data. We evaluated the classification experiments on MNIST.

Setting. We chose the pure softmax as our baseline model. We compared the pure softmax with CEC for classification task, in order to highlight the advantage of class-encoder. Note that CEC drops into softmax when the weight λ becomes 0 in Eq. 5.

Fig. 6 shows the architecture of CEC. The decoder was a single FC layer since, with a large number of experiments, we found that the one-layer decoder was most suitable for reconstruction.

Initialization \ Training	softmax	CEC (CE+softmax)
AE	1.40±0.23	1.29±0.18
DAE	1.28±0.22	1.16±0.20
CAE	1.26±0.12	1.15±0.09

Table 1. Test error rates (in percentage) on MNIST. In each line, the baseline model was compared with CEC that initialized by the same method.

For the diversity of experiment, we initialized the network in 3 different ways – AE, DAE, and CAE. Then, we took the pre-trained networks for either CEC or softmax.

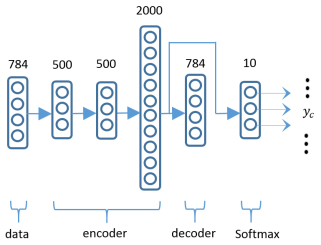

Figure 6. CEC with multi-layer encoder and single-layer decoder. The baseline was the same but without decoder.

Result. Table 1 shows that our CEC outperforms the baselines on MNIST classification. We found that the method of initialization (AE, DAE, or CAE) does not influence the CEC reaching better results.

It should be mentioned that the training error rate reached zero for all the models. Therefore, the class-encoder improved the classifier's generalization.

4.3. DCEC for face recognition

In combination with the advantages of CEC and feature-level strategy, DCEC was employed to learn discriminative representation of human faces.

Data. For training our DCEC, we collected a set of face images from websites, and formed a database called *Webface* (Fig. 7). It contains 156,398 face images of 4,024 identities, most of which are celebrities. Each identity owns quasi-equal number of images. All these images were roughly aligned according to a group of landmarks [25], and normalized to the size of 100×100 with RGB channels. Finally, 3,500 identities were selected to form the training set, and the rest were devoted to the validation. We tested our model on LFW [9] with their official unrestricted protocol. The identities of Webface and LFW are exclusive.

Setting. To build the CNN module, we adopted one convolutional layer and two locally-connected layers, each of which was followed by a max-pooling layer. Locally-connected layer is similar to convolutional layer, while it does not share weights within feature maps. Therefore, it is suitable to extract features from a set of regular images, eg. human faces. As to CEC, the encoder and the decoder were both of single FC layer. The network employed ReLU as activation function. The softmax corresponded to the training identities. See Table. 2 for the details of the parameters.

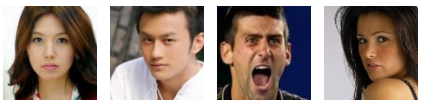

Figure 7. Examples of the Webface database. Through large range of age, expression, pose, and external environment, the database contains eastern and western people of quasi-equal number.

Name	Type	Filter Size/ Stride	Output Size
Conv1	conv.	$3 \times 3/1$	$100 \times 100 \times 32$
Pool1	max pooling	$2 \times 2/2$	$50 \times 50 \times 32$
Local2	local	$3 \times 3/1$	$50 \times 50 \times 64$
Pool2	max pooling	$2 \times 2/2$	$25 \times 25 \times 64$
Local3	local	$3 \times 3/1$	$25 \times 25 \times 128$
Pool3	max pooling	$2 \times 2/2$	$13 \times 13 \times 128$
h_1	FC	N/A	512
h_2 (encoder)	FC	N/A	256
z (decoder)	FC	N/A	512
Softmax	softmax	N/A	3500

Table 2. Parameters of the architecture of DCEC for face representation learning. Both the layer z (decoder) and softmax followed the layer h_2 (encoder).

Each image was horizontally flipped to double the data amount. We generated totally about 25 million intra-person pairs. The CNN module and the CEC were trained together, according to the objective (Eq. 14).

After training, we extracted the feature h_2, which was then processed by PCA and Joint Bayesian (JB) [3] for face verification. We implemented the test under the LFW official unrestricted protocol. Besides, recent studies [14] have noticed the limitations of the original LFW evaluation, eg., limited pairs for verification, high FAR, and no identification experiments. Therefore, we also tried the BLUFR protocol proposed in [14], which included both verification and open-set identification experiments with an exhaustive evaluation of over 40 million of matching scores.

Result. We compared our DCEC with the network that trained by only softmax. We also compared it with contrastive-style DeepID2 and DeepID2+ [18, 19], which used the similar structure (softmax + contrastive cost).

It should be noted that, though increasing higher results have been reported on LFW, it is not clear about the influence of the large private training data they used. To make a fair comparison, we trained all the networks on the same Webface database, respectively.

The results are listed in Table. 3. Our DCEC yielded the best results under all the protocols. The *softmax-only* column shows that the absence of class-encoder leads to sig-

	DeepID2	DeepID2+	Softmax only	DCEC
VR (%) PCA+JB	94.97	95.33	94.21	**95.87**
VR (%) @FAR=0.1%	55.51	57.13	38.61	**57.22**
DIR (%) @FAR=1%, Rank=1	20.19	15.27	12.38	**21.58**

Table 3. The first line shows the accuracies under LFW unrestricted protocol. The second and the bottom lines indicate the two criteria of the BLUFR protocol, respectively.

Method	VR (%)
LM3L [8]	81.3 $\pm$1.2
DDML (LBP) [7]	81.3 $\pm$1.6
DDML (combined) [7]	82.3 $\pm$ 1.5
EigenPEP [13]	84.8 $\pm$ 1.4
DeepFace-single [21]	**91.4** $\pm$ **1.1**
DCEC (fusion)	**90.2** $\pm$ **0.4**

Table 4. Comparison on the YTF database, with the first two accuracies in bold.

nificant depravity of performance. Hence, the improvement of DCEC was mainly attributed to the class-encoder.

The BLUFR evaluation indicated that the proposed method performed better under practical scenarios like verification at low FARs and the watch-list task in surveillance.

To eliminate the background, we cropped the face images according to 7 patches used in Sun et al. [18], and trained 7 DCECs with them. We fused the 7 models and tested them on the YouTube Faces (YTF) database [24]. This gave a competitive performance (Table. 4). Note that DeepFace [21] used much more data (4.4 million images) and deeper architecture than ours.

Analysis. Our DCEC used only intra-class pairs for training, and obtained better results than DeepID2 and DeepID2+ which used both intra- and inter-class pairs. It implies that inter-class pairs contribute very little for training. In addition, rather than the penalty by feature distance (contrastive cost), intra-class reconstruction gives better regularization for learning robust and discriminative face representation. There are two reasons for this. First, the L_2 contrastive cost gives limited effect in the high-dimensional feature space, whereas the class-encoder minimizes the intra-class variation implicitly. Second, in the high-dimensional space, the discriminative methods often allocate much larger partition than the proper class, leading to false positives with high confidence [15]. By contrast, the generative method, involved in CEC, eliminates the nuisance factors in the feature space with their low marginal probability.

Negative pairs. DCEC does not require inter-class pairs (the negatives). This can accelerate the training process comparing with the contrastive-style methods or the margin-style methods (often with time-consuming hard-negative-mining).

5. Conclusion

In this paper, we have two main contributions.

Firstly, we propose a novel class-encoder model, which minimizes the intra-class variations and learns discriminative manifolds of data at a class scale. The experiment on MNIST shows that, if data is well aligned and with mono-background, the mere data-level reconstruction is able to bring about discrimination in not only the decoder, but the encoder as well. We further imposed the class-encoder into the softmax classifier and improves the ability of generalization. The intra-class convergency leads to a sharp priori distribution, from which we obtain high value of conditional probability given the trained weight matrix and the yielded label. As a result, the lower-bound of likelihood raises.

Secondly, we generalize the class-encoder to the feature-level, and combine the convolutional network and the CEC to learn discriminative features (Fig. 8). Our DCEC obtained competitive results with much less training data regarding to state-of-the-art on face recognition. The feature-level strategy has well coped with size issue and translation issue of FC networks; and CNNs have gained better generalization from class-encoder.

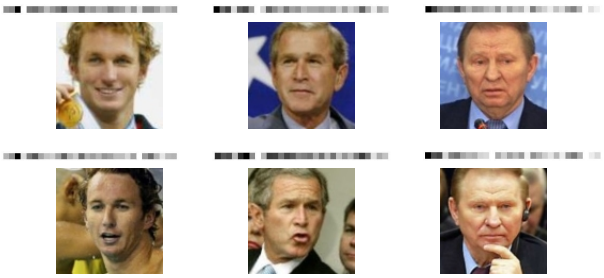

Figure 8. Instances in LFW and the corresponding feature vectors learned by DCEC. Each column belongs to an identity.

6. Acknowledgments

This work was supported by the Chinese National Natural Science Foundation Projects #61105023, #61103156, #61105037, #61203267, #61375037, #61473291, National Science and Technology Support Program Project #2013BAK02B01, Chinese Academy of Sciences Project No.KGZD-EW-102-2, and AuthenMetric R&D Funds. The Tesla K40 used for this research was donated by the NVIDIA Corporation.

References

[1] G. Alain and Y. Bengio. What regularized auto-encoders learn from the data generating distribution. *arXiv preprint arXiv:1211.4246*, 2012.

[2] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[3] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun. Bayesian face revisited: A joint formulation. In *Computer Vision–ECCV 2012*, pages 566–579. Springer, 2012.

[4] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[6] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[7] J. Hu, J. Lu, and Y.-P. Tan. Discriminative deep metric learning for face verification in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1875–1882. IEEE, 2014.

[8] J. Hu, J. Lu, J. Yuan, and Y.-P. Tan. Large margin multimetric learning for face and kinship verification in the wild. In *Proc. ACCV*, 2014.

[9] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

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

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt. Eigen-pep for video face recognition. *Submitted to ECCV and under review*, 2104, 2014.

[14] S. Liao, Z. Lei, D. Yi, and S. Z. Li. A benchmark study of large-scale unconstrained face recognition. In *Biometrics (IJCB), 2014 IEEE International Joint Conference on*, pages 1–8. IEEE, 2014.

[15] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *arXiv preprint arXiv:1412.1897*, 2014.

[16] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 833–840, 2011.

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[18] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems*, pages 1988–1996, 2014.

[19] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. *arXiv preprint arXiv:1412.1265*, 2014.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[21] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1701–1708. IEEE, 2014.

[22] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[23] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

[24] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 529–534. IEEE, 2011.

[25] J. Yan, Z. Lei, D. Yi, and S. Z. Li. Learn to combine multiple hypotheses for accurate face alignment. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 392–396. IEEE, 2013.

[26] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 113–120. IEEE, 2013.

[27] Z. Zhu, P. Luo, X. Wang, and X. Tang. Recover canonical-view faces in the wild with deep neural networks. *arXiv preprint arXiv:1404.3543*, 2014.