Learning Compact Face Representation: Packing a Face into an int32

Haoqiang Fan [*] [†]
Tsinghua University
fhq13@mails.tsinghua.edu.cn

Mu Yang [†]
Tsinghua University
m-yang11@mails.tsinghua.edu.cn

Zhimin Cao
Megvii Inc.
czm@megvii.com

Yuning Jiang
Megvii Inc.
jyn@megvii.com

Qi Yin
Megvii Inc.
yq@megvii.com

ABSTRACT

This paper addresses the problem of producing very compact representation of a face image for large-scale face search and analysis tasks. In tradition, the compactness of face representation is achieved by a dimension reduction step after representation extraction. However, the dimension reduction usually degrades the discriminative ability of the original representation drastically. In this paper, we present a deep learning framework which optimizes the compactness and discriminative ability jointly. The learnt representation can be as compact as 32 bit (same as the int32) and still produce highly discriminative performance (91.4% on LFW benchmark). Based on the extreme compactness, we show that traditional face analysis tasks (e.g. gender analysis) can be effectively solved by a Look-Up-Table approach given a large-scale face data set.

Categories and Subject Descriptors

I.4 [**Image Processing And Computer Vision**]: Image Representation; H.3 [**Information Storage And Retrieval**]: Content Analysis and Indexing

Keywords

face recognition; face search; deep learning

1. INTRODUCTION

Numerous vision tasks benefit from a compact representation of the image data. In face search/analysis tasks, the vector representation of a face image is typically of very high dimension [2] to preserve sufficient discriminative ability.

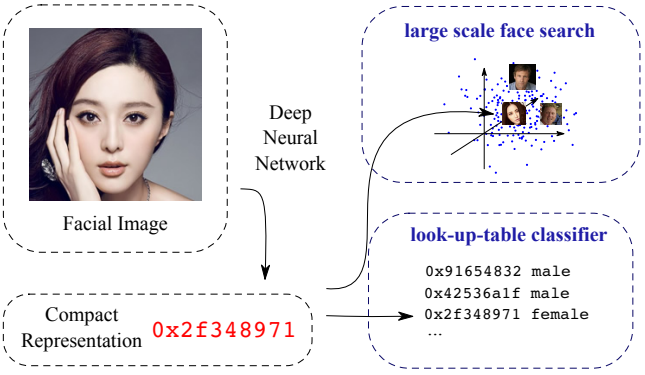

Figure 1: Compact face representation. Our system exploits deep neural networks to produce extremely compact face representation that can fit into a 32-bit integer. The short and low-dimensional representation facilitates efficient large scale face search. In addition, it enables very simple classifiers for face attribute analysis given large scale data set.

In tradition, a face representation of thousands of dimensions has already been considered as "compact" in the community of face recognition. Some dimension reduction or quantization methods may be adopted to compress the representation dimension for the applications of large-scale face systems. However, the discriminative ability is usually decreased due to the compactness requirement. As a result, existing methods make different tradeoffs between compactness and discriminative ability.

To pursue a compact and discriminative representation of a face image, our key observation is that the compactness and discriminative ability should be optimized together. The compactness of the representation is decided by the feature dimension and discreteness strategy. For the feature dimension part, a dimension reduction step is usually adopted after representation extracted. Discreteness is another key issue in the representation learning. Storing quantized or categorical information requires less bits than storing a real-value variable. For example, an int32 data structure can store a single bin of 32-bit feature or four independent bins of 8-bit. Since existing dimension reduction and discreteness [4, 11, 5, 1] methods are all post-processing. The dis-

[*]This work was done when Haoqiang Fan was a visiting student at Megvii Inc..
[†]Ordered lexicographically. Equal contribution.

MM'14, August 24–27, 2014, New York, NY, USA.

ACM 978-1-4503-3063-3/14/11 ...$15.00.
http://dx.doi.org/10.1145/2647868.2654960.

criminative ability will degrade drastically during these two steps.

In this paper, we present a new deep learning framework called Deep Compactness Learning, which can tackle the aforementioned problems in a unified framework. Our contributions are summarized as follows:

1. We propose a deep learning framework which can optimize the compactness and discriminative ability of face representation jointly.

2. We conduct extensive experiments to show our representation learning framework achieves good tradeoff between compactness and discriminative ability (91.4% accuracy on LFW benchmark with 32-bit length).

3. We demonstrate the advantage of our extreme compact representation in the applications of face search and analysis tasks.

2. DEEP COMPACTNESS LEARNING

In this section, we describe the details of our deep representation learning framework. The compactness and discriminative ability of the representation is jointly optimized in a unified CNN framework.

2.1 Joint Optimization Framework

In order to make the representation compact, we have to incorporate the low-dimension and discreteness constraint into our framework. The representation is formulated as a function from image data to a feature space:

$$f : M_{h\times w}(\mathbb{R}) \to \mathbb{R}^n. \tag{1}$$

A loss function is used to learn the function map according to certain criteria. We adopt a pair-based loss function which encourages the identity-preserving property of the learned representation:

$$\mathcal{L} = \sum_{x_1,x_2} L\left(\alpha\left|f(x_1) - f(x_2)\right| - \beta, \delta(x_1, x_2)\right), \tag{2}$$

where $\delta(\cdot,\cdot)$ indicates whether the two images belong to the same person (-1 for same, 1 for different), and

$$L(y, l) = \log\left(1 + e^{-ly}\right). \tag{3}$$

The loss function encourages small distance between the matched (same identity) face pairs and large distance between unmatched pairs. In this way, the feature preserves the discriminative information about the face.

The low-dimension constraint is enforced by setting the model's output dimension n to a small enough number. To produce a 32-bit representation, we can set $n = 32$ and the representation becomes a binary representation. We also allow even smaller n so that more bits can be assigned to each dimension (e.g. 4 bits for each of the 8 dimensions). Using more than one bit has the advantage of forming a hierarchical structure in the feature space so that the data points can be indexed at different levels of granularity. However, some applications explicitly demand binary representation, so we treat the binary and non-binary cases separately.

Another constraint is discreteness, which means each dimension of the model's output has to be rounded:

$$f(x) = \left\lfloor 2^Q f_{\text{model}}(x) \right\rfloor, \tag{4}$$

where Q corresponds to the number of bits available for encoding one dimension, and $f_{\text{model}}(x) \in [0, 1)^n$.

However, the non-differentiable rounding operator posses problem to gradient-based learning algorithms. To overcome this obstacle, we propose three different techniques. The first technique called "rounding error term" is the most straight forward. A "noise" term $\tilde{f}(x)$ is introduced to model the error brought by rounding:

$$f(x) = 2^Q f_{\text{model}}(x) + \tilde{f}(x), \tag{5}$$

where $\tilde{f}(x)$ corresponds to the residual. When computing the gradient of the loss function with respect to model parameters, this term is treated as a constant.

The technique works well for non-binary cases. However, its performance is suboptimal when Q becomes as low as 1. Therefore, we propose two different specialized techniques to handle the binary case.

The first technique associates the model's real valued output with a random n-bit variable. The i-th bit of $f(x)$ has a probability of $f_{\text{model}}(x)_i$ to be 1 and $1 - f_{\text{model}}(x)_i$ probability to be 0. The bits are independent. Then we take the expectation of the loss function:

$$\mathcal{L}' = \sum_{x_1,x_2} \mathrm{E}\left[L(\alpha\left|f(x_1) - f(x_2)\right| - \beta, \delta(x_1, x_2))\right], \tag{6}$$

where the expectation is taken over the random choices of $f(x_1)$ and $f(x_2)$. It is easy to verify that $\mathcal{L}'$ is differentiable with respect to the model's output. Computing the expectation directly is intractable. However, it can be efficiently computed by dynamic programming. For x_1 and x_2, let $D_{i,j}$ be the probability that $f(x_1)$ and $f(x_2)$ differs at j bits in their first i bits. We have

$$\begin{aligned} D_{i,j} = \quad & (1 - p_1 - p_2 + 2p_1p_2)D_{i-1,j} \\ & +(p_1 + p_2 - 2p_1p_2)D_{i-1,j-1}, \end{aligned} \tag{7}$$

where $p_1 = f(x_1)_i, p_2 = f(x_2)_i$. The boundary conditions are $D_{0,0} = 1$ and $D_{i,-1} = 0$.

Another technique aims at minimizing the error introduced by rounding. The idea is to encourage the model to output binarized value by adding a standard deviation term:

$$\mathcal{L}' = \mathcal{L} + \omega \sum_i \mathrm{Std}(f(x)_i), \tag{8}$$

where $\mathrm{Std}(\cdot)$ denotes the standard deviation across the training set.

Our framework is jointly optimized in the sense that both the requirements of compactness and discriminative power are tightly incorporated into the framework. This is in contrast to other methods which use hashing or dimensionality reduction algorithms as a post-processing step.

2.2 Deep Learning Model

We formulate the function family using CNN (Convolutional Neural Network). The CNN is a composition of multiple linear and non-linear operators.

$$f_{\text{cnn}}(x) = f_n\left(f_{n-1}\left(\cdots f_1\left(x\right)\cdots\right)\right). \tag{9}$$

The first type of the operators is the convolution which filters the multi-channel image signal:

$$C_W(x)_{i,j,k} = \sum_{u,v,w} W_{u,v,k,w} x_{i-u,j-v,w} + B_k. \tag{10}$$

Another operator is max-pooling which reduces the size of the image:

$$M_s(x)_{i,j,k} = \max_{0 \le u,v < s} x_{is-u,js-v,k}. \quad (11)$$

Non-linearity is introduced to the network by using element-wise non-linear operators:

$$g(x) = \text{abs}(\tanh(x)). \quad (12)$$

This activation function is inspired by [7] which reveals its advantage in the object recognition task.

The structure of a typical CNN is illustrated in Figure 2.

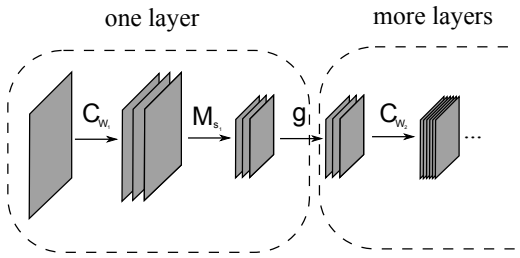

Figure 2: Typical structure of convolutional neural network. Convolution, max-pooling and non-linear operators are three key modules to form a multi-layer convolutional neural network.

3. EXPERIMENT

3.1 Effects of Representation Learning

Firstly we evaluate the proposed framework on face verification task to validate the effectiveness of our method. Our experiment is conducted on the Labelled Faces in the Wild (LFW) [6] data set. We use an outside training set (around 630 thousand faces crawled from the web) which has little overlap with LFW (less than 0.6% pictures in LFW appear as near duplicate in our training set). A 6-layer neural network containing 9 million parameters is employed in our system (inspired by [3]). The parameters in the neural networks are optimized by Stochastic Gradient Descent, and standard deep learning techniques including greedy pre-training are adopted to further accelerate training process.

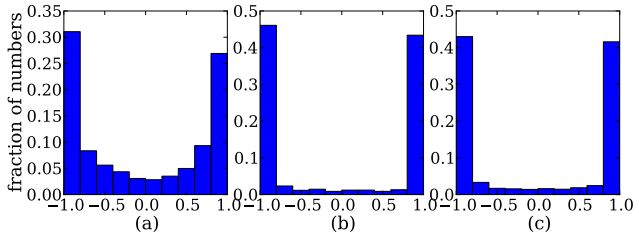

Figure 3: Demonstration of binarized activation value. All the three techniques (a. rounding error term; b. dynamic programming; c. standard deviation) encourage the neurons' output to saturate.

We study two categories of 32-bit representations. The first category is the binary representation in which each bit corresponds to one binarized dimension. The three techniques described in Section 2.1 are implemented and compared. It is observed in the experiment that all these techniques effectively encourage the activation function to saturate (Figure 3). As shown in Table 1, with the dynamic programming technique or the standard deviation technique, our representation achieves 88.5% verification accuracy on LFW. As a comparison, we evaluated a baseline method based on high dimensional LBP and PCA [2]. Though it achieves an accuracy of 96% when more than 1000 dimensions are used, the performance drastically degrades to 81% when the dimension is reduced to 32.

Table 1: Face verification accuracy on the LFW benchmark with the binary representation. The table lists accuracy achieved by different techniques to handle the binarization constraint.

Length	Method	Accuracy
32 bits	rounding error term	87.7%
32 bits	dynamic programming	88.4%
32 bits	standard deviation	88.5%

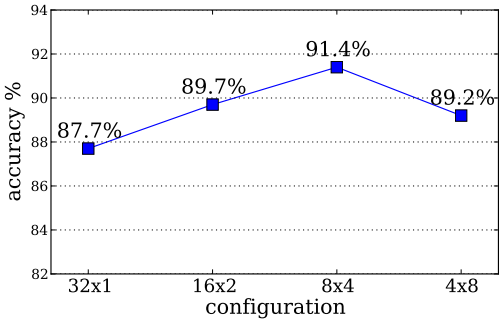

Figure 4: Face verification accuracy obtained by different configurations of the 32 bits. Rounding error term is used to produce quantized features of 32, 16, 8 and 4 dimensions. The best result is achieved by using 4 bits for each of the 8 dimensions.

Conversely, in the second category, each dimension is no longer limited to 1 bit, while the total length of each feature is still 32 bits. We test 4 configurations, from 32×1 to 4×8 bits. The the rounding error term is used to produce discrete features of different dimensions. As shown in Figure 4, the best result is obtained by using 4 bits for each of the 8 dimensions. It achieves an accuracy of 91.4% which is better than the binary representations. It validates the benefits of the flexibility introduced by using multiple bits for one dimension. It is worth mentioning that the 32-bit representation already beats many more complicated and higher-dimensional features [8, 9, 10]. The 8×4 configuration is used throughout remaining face search and classification experiments.

3.2 Applications to Face Search and Analysis

In this subsection we explore the application of the compact representation in face search and analysis.

In large-scale face search, the compact representation is used to build an index for fast nearest neighbour searching. Candidates are quickly generated based on the index, and

finer-grained search and re-ranking steps follow. The 32-bit representation divides the feature space into bins. Each bin contains faces with the same 32-bit representation. We focus on evaluating the recall value when only a small fraction of bins are visited during searching. The bins are visited in ascending order of the distance to the query vector. For fair comparison, the evaluation was also conducted on LFW dataset. We choose persons with more than two pictures in the data set and use their faces as queries to search for matched images. The recall rate when different number of bins are visited is recorded in Figure 5. The figure shows that high recall rate can be obtained by visiting less than one thousandth of the total bins, which greatly accelerates the searching.

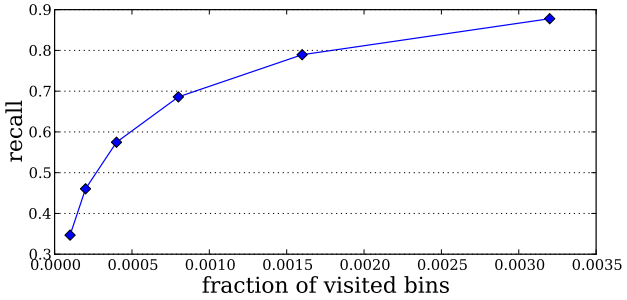

Figure 5: Performance of learnt representation in face search application. This figure shows the recall rate obtained by visiting a small number of bins.

Then we apply the deep face representation in face classification task. The usual way of building classifiers is to learn statistical models based on the feature vectors. As the number of training samples increases, the learning-based methods always suffer from the expensive training cost. Large-scale social network normally maintains a huge image gallery from billions of people, with hundreds of millions of photos uploaded every day. Despite current efforts in online learning and incremental algorithms, it is still difficult to guarantee that the learning process scales well as the training samples continuously come.

When the feature representation is short enough, a scalable and straight forward method of analysis is enabled. The method uses the representation as an entrance and builds a Look Up Table (LUT) for classification. Due to the id-preserving property of the representation, we can hopefully expect the photos belonging to the same bin share common attributes (Figure 6). When the number of training samples is large enough, the accuracy of the classifier approaches the Bayesian error rate. Training and classifying becomes as simple as a table look-up.

Due to limited space, here we construct a prototype system from small-scale data set to illustrate our idea. The task is to predict the gender of the person in a facial image. We gather a training set of 80,000 pictures from existing data sets, and evaluate the performance on the LFW benchmark. As the number of training samples is not large enough, we retrieve data points from nearby bins when doing prediction. We achieve a classification accuracy of 96.8%. We believe as the scale of the data set becomes larger, the performance will be further promoted.

Figure 6: Face samples in different face bins. Based on the representation, the feature space is divided into bins (2^{16} bins in this figure). Clearly, the faces falling in the same bin share a strong correlation with the high-level facial attributes (e.g, gender). This validates our representation bin to be a proper cue for facial attribute analysis.

4. CONCLUSION

We present a new face representation learning approach based on deep learning framework. Our method can jointly optimize the compactness and discriminative ability of the representation in a unified framework. With our extreme compact representation, the face search task can be greatly accelerated and the traditional face analysis task can be effectively solved by large-scale data-driven approach. Extensive experiments justify the effectiveness of our method.

5. REFERENCES

[1] Y. Bengio. Learning deep architectures for ai. *FTML*, 2(1):1–127, 2009.

[2] D. Chen, X. Cao, F. Wen, and J. Sun. Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. In *CVPR*, pages 3025–3032. IEEE, 2013.

[3] H. Fan, Z. Cao, Y. Jiang, Q. Yin, and C. Doudou. Learning deep face representation. Technical report, Megvii. Inc, Beijing, March 2014.

[4] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, pages 817–824. IEEE, 2011.

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

[6] 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 07-49, University of Massachusetts, Amherst, October 2007.

[7] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *ICCV*, pages 2146–2153. IEEE, 2009.

[8] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, pages 365–372. IEEE, 2009.

[9] Y. Taigman and L. Wolf. Leveraging billions of faces to overcome performance barriers in unconstrained face recognition. *arXiv preprint arXiv:1108.1122*, 2011.

[10] Q. Yin, X. Tang, and J. Sun. An associate-predict model for face recognition. In *CVPR*, pages 497–504. IEEE, 2011.

[11] L. Zhang, Y. Zhang, J. Tang, X. Gu, J. Li, and Q. Tian. Topology preserving hashing for similarity search. In *ACMM*, pages 123–132. ACM, 2013.