Face Recognition via Deep Sparse Graph Neural Networks

Renjie WU[1]
wurj-sjtu-waseda@ruri.waseda.jp
Sei-ichiro KAMATA[1]
kam@waseda.jp
Toby Breckon[2]
toby.breckon@durham.ac.uk

[1] Graduate School of Information, Production and Systems
Waseda University
Kitakyushu-shi, Japan

[2] Engineering and Computing Sciences
Durham University, Durham, UK

Abstract

In recent years, deep learning based approaches have substantially improved the performance of face recognition. Most existing deep learning techniques work well, but neglect effective utilization of face correlation information. The resulting performance loss is noteworthy for personal appearance variations caused by factors such as illumination, pose, occlusion, and misalignment. We believe that face correlation information should be introduced to solve this network performance problem originating from by intra-personal variations. Obviously, different poses of same person have similar feature structure, and even different people may have some similar facial sub-regions. We propose a graph representation based on the face correlation information that is embedded via the sparse reconstruction and deep learning within an irregular domain. The proposed method achieves high recognition rates of 99.58% on the benchmark LFW facial evaluation database.

1 Introduction

Face recognition has become one of the hottest topics in the area of computer vision and pattern recognition. It has been extensively applied in identity validation and recognition. Many researchers have been studying new face recognition algorithms [1, 4, 7, 17, 23, 24] for decades. The visual signature of the human face has clear advantages over other biometric information because it is natural and easy to handle face images. However, that main problems in face recognition include highly overlapping intra and inter identity distributions due to naturally occurring variations in pose, age, expression, occlusion, and external imaging factors such as variations of scene illumination.

A decade ago, some face recognition approaches focus on the sparse coding. Olshausen and Field [9] have indicated that neural networks in the human vision system perform sparse coding of the learned features, qualitatively which is similar to the receptive fields of simple cells in V1 (V1 is the primary visual cortex). Subsequently, a generation of facial recognition algorithms are enabled via the sparse coding for finding the succinct representations of the stimuli. Given unlabeled input data only, sparse coding learns basis functions that capture the high-level features. For face recognition, Shan [5] developed a hierarchical model

called Recursive ICA (RICA), which captures nonlinear statistical structures of the visual inputs, that cannot be captured by a single layer of ICA. Shan [12] also performed variational recognition tasks by sparse coding learnt from natural images. However, the sparse coding model has a problem of lower recognition accuracy in general. Now, with the advent of deep learning, the recognition accuracy has been significantly improved. We think the concept of sparse has not been completely abandoned; in fact, it can be embedded to the deep learning strategy which still achieves good performance in some cases such as occlusion.

Deep learning networks have attracted more and more attention in face recognition [10]. Face recognition accuracy has been incredibly boosted with better deep network architectures and supervisory learning methods in recent years. Sun [18] proposed a supervised learning method of deep face representation. His approach greatly reduced the intra-personal variations in the face representation. Subsequently, DeepID [19] and DeepFace [21, 22] are proposed to learn a discriminative deep face representation in large-scale face identification. In DeepID2+ [20], Sun has developed a learning method of deep face representation through joint face identification-verification (adding verification supervisory signals). The goal of this approach is to absorb the significant intra-personal variations in face representation. GoogLeNet [16] based FaceNet [11] is proposed to train a deep network by triplet loss. The learnt features are mapped into a compact Euclidean space for evaluating face similarity. Among the state-of-the-art deep neural networks, ResNet [6], GoogLeNet and VGG [13] are ranked in the 3-top in general image classification competition. However, the DeepID series of networks is less accurate than these 3-top networks, because depth of DeepID series is much shallower. However, in this paper, we optimize DeepID2+ to learn the features for face recognition because we just need to verify the ability of deep sparse graph.

Although a large-scale face dataset is used to improve the performance, intra- and extra-correlations of face parts (*e.g.* eyes, noses, etc.,) are not taken into consideration. We think this kind of face correlation information should be implicit within optimising neural network for the performance gain. This poses a new problem of convolutional neural network with irregular spatial domain.

As discussed above, we need to verify the performance gain by using a sparse graph. At first each face image is divided into several feature blocks. Each vertex in the graph corresponds to the feature block respectively. Face correlation information is used to link any two vertices. In order to optimize the deep neural network, we should introduce some structural information from local to global. For example, each face image is divided into I blocks as Fig. 1. The graph is constructed to describe the sparse constraints with local and global structures as shown in Fig. 2. For different images, the maximum degree vertex of each subgraph is related to face correlation of the corresponding feature blocks. We find that the maximum degree vertex is the most representative feature block. Finally, we obtain the whole graph of all face images for each individual.

A novel approach named Deep Sparse Graph Neural Networks (DSGNN) is proposed. Each feature block is sparsely reconstructed by the sparse coding based on the graph. On learning the sparse coding using universal nature images, the common dictionary is extracted. Subsequently, the sparse coding is guided by graph constraints for smoother expression of face information. The resulting reconstructed face sparse graph is an input to the deep neural network. As shown in Fig. 3, the similar feature blocks are connected by checking the common weights corresponding to face correlation.

From the experiments, the proposed DSGNN is significantly better than the previous DeepID or DeepFace for face recognition. Compared to other state-of-the-art methods [11, 14, 15, 20, 21, 22], our method is more accurate than most of the other methods, and is

Figure 1: Face images are divided into I blocks. In this case, the input image is first resized to 120×120 pixels, next divided into 15×15 blocks. Size of each block is 8×8 pixels, where $I = 225$.

comparable to FaceNet [21].

Taken together, our contributions are as follows:

1. We construct a graph using face correlation information and obtain the reconstructed face sparse graph.

2. We construct deep sparse graph neural networks, which introduces the concept of face sparse graph, and improves the recognition accuracy.

The rest of the paper is organized as follows. Section 2 describes the structure of the face graph. Subsequently, Section 3 shows the sparse reconstruction of face graph. Detail of our proposed method is described in the Section 4. Simulation and the results are shown in Section 5. Finally, conclusions and future work are discussed in Section 6.

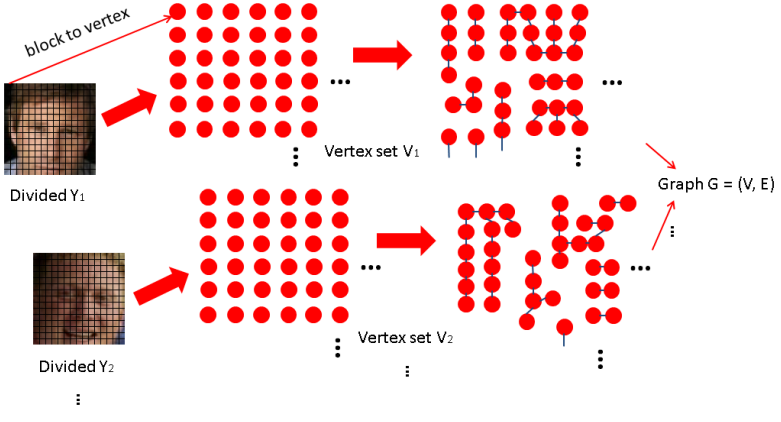

Figure 2: Face graph $G = (V, E)$. In this example, each vertex is connected to a 1-nearest neighbor related vertex.

2 Graph Construction

In this section, we show our graph representation of face images. Here we use the Euclidean distance to calculate the correlation between two different feature blocks.

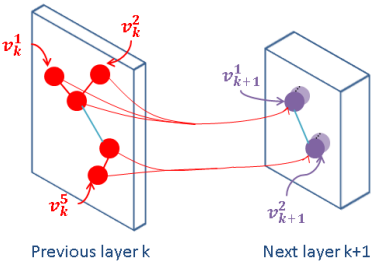

Figure 3: Depiction of relation between graph and convolution layer. The intra- graph (show as green graph) is shared convolute to the next layer. The graph of the next layer is the abstraction of the previous layer graph.

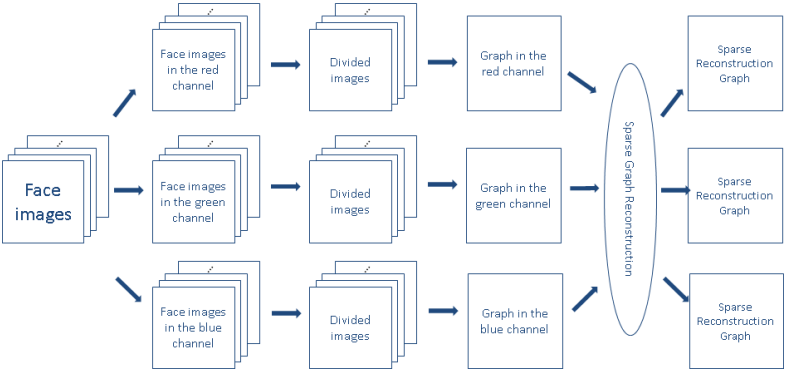

Figure 4: This view shows the graph sparse reconstruction with different color channels. Richer image information can be obtained under the different color channels, compared with the previous works.

2.1 Graph Definition

Let us denote a face image dataset $\{Y_1, Y_2, ..., Y_L\}$ with L images. The graph construction produces $G = (V, E)$ consisting of vertex set $V = V_1 \cup V_2 \cup ... \cup V_L$ where each vertex subset V_l is associated with the sample Y_l. As mentioned before, in order to construct the V_l, we divide each face image into I blocks as shown in Fig. 1. As a result, the vertex set V_l can be expressed as $V_l = \{v_l^1, v_l^2, ..., v_l^I\}$. Obviously, for i-th block of l-th face image, the vertex v_l^i corresponds to an M-dimensional vector $y_l^i \in R^M$. Take E to be a set of undirected edges as $E = \{e_{\alpha\beta} | v_\alpha, v_\beta \in V\}$, where α and β are used to assign two different feature blocks, that is $\alpha, \beta \in \{(i,l) | i = 1, 2, ..., I; l = 1, 2, ..., I\}$. In G, the graph should contain global information across all face images. The similar vertices (blocks) are connected by edges.

2.2 Graph Structure

Each edge e of an undirected graph G is associated to a weight w. In V, we suppose two different vertices v_α and v_β correspond to two vectors y_α and y_β. A weight $w_{\alpha\beta}$ means a

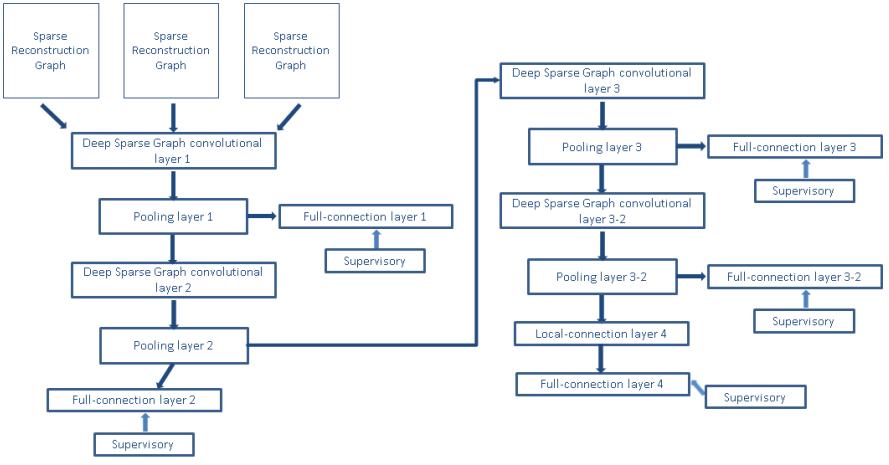

Figure 5: Deep sparse graph neural networks. In this case, the framework was designed with a simple supervised learning network such as DeepID2+. Of course, other state-of-the-art deep neural networks (VGG, GoogLeNet and ResNet) can also be applied and optimized. DSGNN have 5 convolutional layers, and the filters are utilized by cubic B-spline fast filter with vector of control points. In each deep sparse graph convolutional layer, the input of previous layer is sparse graph. And, supervisory signals are connected to convolutional layer (after pooling layer), while the lower convolutional layers can trained with back-propagated from higher layers. The final full-connection feature extraction is used for face recognition.

correlation between two different vertices v_α and v_β. If $w_{\alpha\beta} > 0$, then we suppose there is an edge $e_{\alpha\beta}$ between two vertices. In other words the corresponding feature vectors belong to the same category. The edge set should be redefined by $w_{\alpha\beta}$. Next, we have to define the weight $w_{\alpha\beta}$.

Here, the mutual k-NN [2] is simply used to estimate the weight in the graph. Due to the fact that in the mutual k-NN, all vertices have a degree upper-bounded by k. This property helps to produce no vertices with extremely high degree in the graph. In V, if v_α (or v_β) is among the mutual k-NN of v_β (or v_α), the weight $w_{\alpha\beta}$ is introduced by a Gaussian kernel function as follows:

$$w_{\alpha\beta} = \begin{cases} e^{-\frac{\|y_\alpha - y_\beta\|_2^2}{2\sigma^2}}, & if\ v_\alpha \in N_k(v_\beta)\ or\ v_\beta \in N_k(v_\alpha), \\ 0, & otherwise, \end{cases} \tag{1}$$

where σ is the bandwidth parameter of the kernel.

$N_k(v_\alpha) \subset V$ and $N_k(v_\beta) \subset V$ denote the vertex subsets for the k-nearest neighbors of the v_α and v_β in V. Edge set is defined as $E^* = \{e^*_{\alpha\beta} | v_\alpha \in N_k(v_\beta) \vee v_\beta \in N_k(v_\alpha), v_\alpha, v_\beta \in V\}$. After this setting, the graph is undirected and the adjacent relations between two vectors is symmetric.

3 Sparse Graph of Feature Blocks

In this section, we firstly set up the common feature hypothesis to focus on the sparse coding of the universal natural images and then extract the basis functions in common. The basis

functions are taken into the sparse coding so that the face sparse graph can be reconstructed. The graph-guided sparse reconstruction can be developed by a concise representation of the facial features.

Given a data matrix $\mathbf{Y} \in \mathbb{R}^{M\times(L\times I)}$, where each column represents a data vertex $\mathbf{y}_l^i \in \mathbb{R}^M$ on the graph. Let $\mathbf{D} = (\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_N) \in \mathbb{R}^{M\times N}$ be a common dictionary. Here $\mathbf{d}_n$ is the basic function with $N \gg M$ (obviously, solution of $\mathbf{Y} = \mathbf{DS}$ is not unique. $N \gg M$ is the basic condition to obtain the optimal solution). The target of the sparse coding is to find an sparse coefficient matrix $\mathbf{S} = (\mathbf{s}_1^1, \mathbf{s}_1^2, ..., \mathbf{s}_L^I) \in \mathbb{R}^{N\times(L\times I)}$. This optimization problem can be written as follows:

$$\min_{\mathbf{s},\lambda} \frac{1}{2} \|\mathbf{Y} - \mathbf{DS}\|_F^2 + \lambda \|\mathbf{S}\|_1 , \tag{2}$$

where the parameter λ is a scalar regularization parameter that balances the trade off between reconstruction error and sparsity, and $\|.\|_F^2$ represents the Frobenius norm.

The sparse coefficient $\mathbf{S}$ is regularized by the l_1-norm, sparsity is estimated by blocks in $\mathbf{D}$. The structure of sparse graph is missing in Eq. (2). The structured regularization term can realize better performance for the final goal. Therefore, we use a feature fusion which simply means that some similar features are connected by using block correlation. Here, we impose the correlation between columns in $\mathbf{Y}$, which is reflected in the correlation between the rows in $\mathbf{S}$ by embedding function $\frac{1}{2}\Sigma_{e^*_{\alpha\beta}\in E^*, \alpha<\beta} w_{\alpha\beta} \|\mathbf{s}_\alpha - \mathbf{s}_\beta\|_2^2 = \|\mathbf{S\Delta S}^T\|_{Tr}$, where $\mathbf{\Delta} \in \mathbb{R}^{(L\times I)\times(L\times I)}$ is an unnormalized graph Laplacian matrix and $\|.\|_{Tr}$ is trace norm. Obviously, $\mathbf{\Delta} = \mathbf{U} - \mathbf{W}$ is an symmetric positive-semidefinite matrix, where $\mathbf{U} = diag(\Sigma_{\alpha\neq\beta} w_{\alpha\beta})$ is the degree matrix; $\mathbf{W} = (w_{\alpha\beta})$ is the symmetric weight matrix.

In order to integrate the graph structure into our sparse coding, the loss function Eq. (2) can be rewritten as follows:

$$\min_{\mathbf{s},\gamma,\lambda} \frac{1}{2} \|\mathbf{Y} - \mathbf{DS}\|_F^2 + \lambda \|\mathbf{S}\|_{2,1} + \gamma \|\mathbf{S\Delta S}^T\|_{Tr}, \tag{3}$$

where γ is the regularization parameter and $\|.\|_{2,1}$ is the $l_{2,1}$-norm regularizer that measure the distance in feature space via the l_2-norm regularizer. The summation over different vertices of graph is performed via the l_1-norm.

Next we introduce a common feature hypothesis into the basis functions of the sparse coding in common from the universal natural images (used for training). For the human visual system, one notable advantage is that human beings can recognize one person at a simple glance of one face image, while most computer vision face recognition techniques depend on a huge number of face images for initial training. Therefore, the concept of the common feature hypothesis suggests that all visual stimuli share common characteristics such that the knowledge from one set of visual stimuli can be applied to a completely different problem. Finally, we get the graph of sparse reconstruction, each vertex v_l^i corresponds to $\hat{\mathbf{y}}_l^i = \mathbf{Ds}_l^i$.

4 Deep Sparse Graph Neural Network

In this section, we describe the proposed DSGNN which can optimize the high-level features. Our network is more robust to pose, illumination variations and occlusions. The state-of-the-art deep neural networks (*e.g.* VGG, GoogLeNet, ResNet) are very powerful, but the computational cost is very high if the network is deep, and they have the risk of over-fitting. For the above networks, we consider the optimal method of a fundamental framework - Convolutional Neural Networks (CNNs). CNNs are a biologically inspired class of deep learning

models that is trained end to end from raw pixel values to classifier outputs through restricted connectivity between layers (local filters), parameter sharing (convolutions) and special local invariance-building neurons (max pooling). Here, we consider that the convolution of a filter across the spatial domain is non-trivial within the irregular spatial domain [8]. Here, we use the graph to describe the spatial correlation between the vertices and perform convolution by the multiplication in the spectral graph domain.

As we discussed, graph $G^* = (V, E^*)$ consists of vertices V and the edges E^*, $w_{\alpha\beta}$ is the weight of an edge $e^*_{\alpha\beta}$ between two vertices v_α *and* v_β and each vertex v^i_l corresponds to vector $\hat{\mathbf{y}}^i_l$. $\acute{\mathbf{\Delta}} = \acute{\mathbf{U}} - \acute{\mathbf{W}}$ is an unnormalized Laplacian matrix; $\acute{\mathbf{D}} = diag(\Sigma_{\alpha\neq\beta} w_{\alpha\beta})$ is the degree matrix; $\acute{\mathbf{W}} = (w_{\alpha\beta})$ is the symmetric weight matrix. Since $\acute{\mathbf{\Delta}}$ is a symmetric positive-semidefinite matrix that admits an eigenvalue decomposition $\acute{\mathbf{\Delta}} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^T$, where the orthonormal eigenvalues $\mathbf{\Phi} = (\phi^1_l, \phi^2_l, ..., \phi^I_l)$, and $\mathbf{\Lambda} = diag(\lambda^1_l, \lambda^2_l, ..., \lambda^I_l)$ is the diagonal matrix of the corresponding non-negative eigenvalues. Now, we can describe a convolution construction from layer k to layer $k+1$, without pooling layer:

$$\hat{\mathbf{y}}^q_{k+1,l} = \delta(\Sigma^P_{p=1} \mathbf{\Phi}\mathbf{F}_{k,p,q}\mathbf{\Phi}^T \hat{\mathbf{y}}^p_{k,l}), \tag{4}$$

where, $p = 1, 2, ..., P$ is an index of k-layer vector $\hat{\mathbf{y}}^p_{k,l}$, $q = 1, 2, ..., Q$ is an index of $k+1$-layer vector $\hat{\mathbf{y}}^q_{k+1,l}$, $\mathbf{F}_{k,p,q}$ is a diagonal matrix of spectral multipliers representing a learnt filter in the frequency domain, and δ is a nonlinearity applied to the vertex-wise function values.

Here, we can minimize the loss function to learn the optimal diagonal matrix $\mathbf{F}_k$ for each layer in the following:

$$\min_F \Sigma^L_{l=1} \ell(f(\mathbf{Y}_l), O_l) + \kappa \Sigma^{K-1}_{k=1} \|\mathbf{F}_k\|^2_F, \tag{5}$$

where $f(\mathbf{Y}_l)$ is an output of the network, O_l is the ground-truth label defined as $O_l = (o_{l1}, o_{l2}, ..., o_{lC}$ $\mathbb{B}^c$ and $o_{lc} = 1$ identify the l-th sample face belongs to the c-th person class.

Furthermore, if we use low-rank sparse to replace mutual k-NN, we can get a more generalized deep sparse graph. The regularized optimization problem is formulated as following as:

$$\min_F \Sigma^L_{l=1} \ell(f(\mathbf{Y}_l), O_l) + \eta \|\mathbf{W}_k\|_* + \nu \|\mathbf{W}_k\|_1 + \kappa \Sigma^{K-1}_{k=1} \|\mathbf{F}_k\|^2_F. \tag{6}$$

We discuss and optimization of the DeepID2+ core architectures, including the supervisory signals to full connection layers. The supervisory signals help to learn better mid-level features and optimize the deep neural network. The DSGNN is trained by the sparse graph-guide reconstructed face images that are conductive to smoothing and robust face blocks. Add the network takes batch blocks as graph input. These batch blocks are selected by different positions, channels and poses such that networks could learn sufficient information.

5 Experiments

In this section, we verify face recognition accuracy using benchmark face databases, including LFW to demonstrate the performance. Here, the CeleFaces+ dataset [19] and the WDRef dataset [3] are merged to train the network. The merged dataset includes 290,000 faces of 12,000 persons. 2000 peoples are trained by Joint Bayesian model [3]. Finally, the 6,000 face pairs of LFW are tested for face recognition. For regularization-parameters, we use a 5-fold cross validation on the training dataset to tune the parameters λ and γ. At first, a face

image is represented on a 15×15 2D-blocks, and each block size is 8×8 pixels. We construct a mutual 6-NN graph of the 2D-block with $15^2 = 225$ nodes (for max pooling of size 4, we need to add some null-nodes to backfill missing nodes, for example, $225 + 31 \, mod \, 4 = 0$). Our hyper-parameters are borrowed from the TensorFlow tutorial with DeepID2+ [20], validating a set of 2000 peoples to determine learning rates and training iterations, momentum of 0.9. Our proposed DSGNN achieves higher accuracy 99.58% for face recognition. The accuracy is comparable with previous state-of-the-art methods on LFW are shown in Table. 1. To compare with DeepID2+ nets and DeepID3, DSGNN improve approximately 0.11% and 0.05% average accuracies over DeepID2+ and DeepID3, respectively. The proposed network achieves nearly 70% speedup compared to DeepID2+ (221ms of 25 patches) with implementation on an NVIDIA 780 GPU.

Table 1: Face recognition rates (complete face test) on LFW database.

Method	rates
High-dim LBP	0.9517
DeepFace	0.9735
FaceNet [11]	0.9963
Parkhi's approach [10]	0.9865
DeepID2+ [20]	0.9947
DeepID3 [15]	0.9953
DSGNN	0.9958

6 Conclusions

We proposed a deep sparse graph neural network (DSGNN) for face recognition. The experiments validated the high performance of proposed network on the LFW reference database, exceeding prior work in the field. The face representations of graph sparse reconstruction are more sparse and robust to background noise and occlusion. The DSGNN is more highly selective to person identities and can reduce the number of parameters without losing the accuracy. This work shows that the neural network of a deep sparse graph is feasible within face recognition, and it is believed that there remains substantial room for extension of this concept. Here, due to the limitations of the DeepID framework itself, the accuracy of our network does not exceed that of FaceNet, we will try to optimize the GoogleNet and ResNet framework based on the deep sparse graph in future work. This is obviously feasible, as features of each block can be extracted, but the network will be more complex and difficult to optimise in consideration of training costs.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number 15K00248.

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, 28(12):2037–2041, 2006.

[2] M.R. Brito, E.L. Chavez, A.J. Quiroz, and J.E. Yukich. Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics and Probability Letters*, 35(1):33–42, 1997.

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

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

[5] L. Zhang H. Shan and G.W. Cottrell. Recursive ica. In *Advances in neural information processing systems*, pages 1273–1280, 2007.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016.

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

[8] R. Levie, F. Monti, X. Bresson, and M.M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *arXiv preprint arXiv:1705.07664*, 2017.

[9] B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[10] O.M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proc. BMVC*, pages 1–12, 2015.

[11] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. CVPR*, pages 815–823, 2015.

[12] H. Shan and G.W. Cottrell. Looking around the backyard helps to recognize faces and digits. In *Proc. CVPR*, pages 1–8, 2008.

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

[14] K. Simonyan, O.M. Parkhi, A. Vedaldi, and A. Zisserman. Fisher vector faces in the wild. In *Proc. BMVC*, pages 1–12, 2013.

[15] Y. Sun, D. Liang, X. Wang, and X. Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, pages 1–9, 2015.

[17] X. Wang and X. Tang. Random sampling for subspace face recognition. *International Journal of Computer Vision*, 70(1):91–104, 2006.

[18] X. Wang Y. Sun and X. Tang. Hybrid deep learning for face verification. In *Proc. ICCV*, pages 1489–1496, 2013.

[19] X. Wang Y. Sun and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proc. CVPR*, pages 1891–1898, 2014.

[20] X. Wang Y. Sun and X. Tang. Deeply learned face representations are sparse, selective, and robust. In *Proc. CVPR*, pages 2892–2900, 2015.

[21] M.A. Ranzato Y. Taigman, M. Yang and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proc. CVPR*, pages 1701–1708, 2014.

[22] M.A. Ranzato Y. Taigman, M. Yang and L. Wolf. Web-scale training for face identification. In *Proc. CVPR*, pages 2746–2754, 2015.

[23] Q. Yin, X. Tang, and J. Sun. An associate-predict model for face recognition. In *Proc. CVPR*, pages 497–504, 2013.

[24] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *Proc. ICCV*, pages 113–120, 2013.