Sparse Graph Based Deep Learning Networks for Face Recognition*

Renjie WU†(a), Student Member and Sei-ichiro KAMATA†(b), Member

SUMMARY 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 intra-personal variations. Recently, graph deep learning approaches have emerged for representing structured graph data. A graph is a powerful tool for representing complex information of the face image. In this paper, we survey the recent research related to the graph structure of Convolutional Neural Networks and try to devise a definition of graph structure included in Compressed Sensing and Deep Learning. This paper devoted to the story explain of two properties of our graph - sparse and depth. Sparse can be advantageous since features are more likely to be linearly separable and they are more robust. The depth means that this is a multi-resolution multi-channel learning process. We think that sparse graph based deep neural network can more effectively make similar objects to attract each other, the relative, different objects mutually exclusive, similar to a better sparse multi-resolution clustering. Based on this concept, we propose a sparse graph representation based on the face correlation information that is embedded via the sparse reconstruction and deep learning within an irregular domain. The resulting classification is remarkably robust. The proposed method achieves high recognition rates of 99.61% (94.67%) on the benchmark LFW (YTF) facial evaluation database.

key words: face recognition, atom decomposition, sparse graph reconstruction, deep learning network

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]–[6] 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 problem in face recognition includes 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. The prime solution is sparse coding, once the main research direction.

In the past few decades, some face recognition approaches focus on the sparse coding. Olshausen and Field[7] 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 face recognition algorithms are enabled to finding the succinct representations of the stimuli based on the sparse coding. And, the sparse coding can learns basis functions that capture some high-level features from an unlabeled training dataset. In 2007, Shan[8] proposed a hierarchical model called Recursive ICA, which captures non-linear statistical structures of the visual inputs, that cannot be captured by a single layer of ICA. Shan[9] 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[11] proposed a supervised learning method of deep face representation. His approach greatly reduced the intra-personal variations in the face representation. Subsequently, DeepID[12] and DeepFace[13], [14] were proposed to learn a discriminative deep face representation in large-scale face identification. In DeepID2+[15], 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[17] was 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[18], GoogLeNet and VGG[19] 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. Here, we use the DeepID2+ frame-
Face images are divided into $I$ blocks. In this case, the input image is first resized to $120 \times 120$ pixels, next divided into $15 \times 15$ blocks. Size of each block is $8 \times 8$ pixels, where $I = 225$.

This view shows the sparse graph structure. Obviously, different poses of the same person have similar feature structure, and even different people may have some similar facial sub-regions. The sparse reconstruction is possible to make up for missed information of face image. Each vertex is connected to a $(k = 3)$-nearest neighbor related vertex in this example.

The depiction of sparse graph in deep learning network. An example shows the graph convolution (similar to graph clustering) in three layers. 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.

work to learn the features for face recognition because we just need to verify the ability of deep sparse graph. Although deep learning networks have been very successful, but there are still limited to an underlying Euclidean structure. However, when dealing with complex signals such as image networks, knowledge networks, or something like social networks, recently there has been a growing interest in trying to apply learning on non-Euclidean geomet-
Fig. 4  Sparse graph based deep learning networks. A proposed network is based on a traditional Siamese network. For the Siamese network, parameter updating is mirrored across both subnetworks. In this illustration, we just show the left subnetwork. Proposed network has 5 convolutional layers (sparse graph conv 1–5), and the filters are utilized by Cayley filter [28] with vector of control points. Supervisory signals are connected to full-convolutional layer (FC 1–4, after pooling layer (Pool 1–4)), while the lower convolutional layers can trained with back-propagated from higher layers. Joint Bayesian model [29] is used to learn the supervisory signals from right subnetwork. The final full-connection (FC5) feature extraction is used for face recognition.

to express human face data.

As discussed above, we need to verify the performance gain by using a sparse graph. First, a notion of sparsity appears in various contexts and is a typical example of a hard to define notion. A definition of sparse graph in which the number of edges is much less than the possible number of edges [24]. Of course, Paul E. Black’s definition can not strictly differentiate sparse or not. Here, Lee [25] gave a more informative definition of sparse graph. In order to optimize the deep neural network, we should introduce a graph structural information from local to global. Specifically, 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. For example, each face image is divided into 7 blocks as shown in 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 subnetwork 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. In fact, we also tested other face graph model, such as Elastic Bunch Graph (EGB) and face landmarking graph. Experimental results show that the first design has the highest recognition rate. Therefore, in this article, we still use the first design.

In deep layers of sparse graph deep learning network, similar to a multi-resolution pyramid of sparse graph, graph of deep layer is the abstraction of the graph of shallow layer. From another point of view, these sparse graphs are multi-scale clustering results of the shallow layer graphs. The Deep Learning process increases the classification by topological resolution. The essence of Deep Learning process finds the eigenvalues of graph Laplace, and then use k-means clustering for eigenvector of graphs, and into the next layer. Figure 3 illustrates a multi-resolution clustering pyramid of a sparse graph. Obviously, we usually do not consider the connection between vertices in traditional convolutional neural networks. For multi-scale clustering of graphs, we can refer to the literature [23]. Here, each feature block is first sparsely reconstructed by the sparse coding based on the graph. Subsequently, the coding is guided by sparse graph constraints for localization expression of face information. The resulting reconstructed face sparse graph is an input to the deep neural network. As shown in Fig. 2, the similar feature blocks are connected by checking the common weights corresponding to face correlation. In this paper, our network is based on a traditional Siamese network as shown in Fig. 4.

From the experiments, the proposed sparse graph based deep learning network is significantly better than the previous DeepID or DeepFace for face recognition. Compared to other state-of-the-art methods [13]–[15], [17], [26], [27], our method is more accurate than most of the other methods, and is comparable to FaceNet [13].

Taken together, our contributions are as follows:
1. We construct a sparse graph using face correlation information. Some results show that the proposed sparse graph is better than other traditional face graphs in our network.

2. In the preprocessing, we propose a graph-based sparse reconstruction to improve robustness of face recognition.

3. We show a face sparse graph based deep learning network. Our filters of linear complexity and pooling layer design are simple and fast.

The rest of the paper is organized as follows. Section 2 shows the brief summaries of our modeling and the related work. Section 3 describes the sparse graph and concrete prototype. Subsequently, Sect. 4 shows the sparse reconstruction of the graph. The detail on our network is described in the Sect. 5. Simulation and the results are shown in Sect. 6. Finally, conclusions and future work are discussed in Sect. 7.

2. Related Work

This section shows the related works. Recently, many studies have focused on learning-based face feature extraction and deep learning network.

The deep learning network learns by stacking many hidden layers. DeepFace [13] contains a nine-layer deep CNN model with two convolutional layers and achieves accuracies of 97.35% and 91.4% through alignment of images based on a 3D model and using an ensemble of CNNs on the LFW and YTF datasets, respectively. DeepID [12] was proposed for face recognition and verification with a nine-layer network and four convolutional layers. This deep learning network first learns weights through face identification, extracts features using the last hidden layer outputs and later generalizes them to face verification. DeepID is trained on the Celebrity Faces dataset (CelebFaces) [12] and achieved an accuracy of 97.45% on the LFW dataset. FaceNet [17] is a deep learning network based on GoogLeNet [16]. The network was proposed in [17] and trained on a face dataset with 100 to 200 million images of around 8 million identities. This network uses triplets’ ethnic and provide a method to directly learn an embedding into an Euclidean space for face verification. FaceNet has been evaluated on the LFW and YTF datasets and achieved accuracies of 99.63% and 95.12%, respectively. We can use this model to better learn relational features for face similarity. However, if use above methods, the learned features can not contain personal variations of identity information. Some identity information may have been lost in the hidden layer. We need to train high-level feature while implement the information completely.

Recently, deep learning methods to automatically extract structural characteristics from graphs have been proposed [21], [30], [31]. Examples of deep learning applied to non-grid, non-Euclidean space includes graph wavelets from applying deep auto-encoders to graphs and using the properties of automatically extracted features [32], analysis of molecular fingerprints of proteins saved as graphs [21], and a CNN-based model for handling tree structures in the context of programming language processing [33]. Particularly relevant to our approach is the Geometric Deep Learning model [34], which is a deep learning approach that can extract the features of the 3D geometric data.

In this paper, we come up with a new face recognition algorithm via sparse graph deep learning network. We set up the common feature hypothesis to focus on the sparse coding of the universal images and then extract the basis function in common. The basis functions are then taken into the sparse coding so that the sparse face graph could be reconstructed. Here we use the Euclidean distance to calculate the correlation distance between two different blocks. The graph-guide sparse reconstruction can obtain a more complete structural information. The learning network can optimize to the higher features. Our network is more robust to pose, illumination variations and occlusion.

3. Sparse Graph Construction

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

3.1 Definition of Sparse Graph

In the following, we work with finite simple undirected graphs. In mathematics, if a graph has a few edges then we call it a sparse graph. Here, some standard graph theory terminology are used, and we find it useful to introduce the following:

Let $G_r = (V_r, E_r)$ be a graph of $r$-th layer, the size of vertex set $V_r$ is $|V_r|$, and the size of edge set $E_r$ is $|E_r|$. If
Before the input layer ($r = 0$), let us denote a face image dataset $\{Y^1, Y^2, \ldots, Y^L\}$ with $L$ images. As mentioned before, we divide each face image into $I$ blocks as shown in Fig. 1. Then $Y^i = \{y^{i1}, y^{i2}, \ldots, y^{iI}\}$ is a set of $i$-th face image, and $y^{ij} \in \mathbb{R}^M$ is an $M$-dimensional vector of $i$-th block.

The graph construction produces $G_0 = (V_0, E_0)$ consisting of a vertex set $V_0 = V_0^1 \cup V_0^2 \cup \ldots \cup V_0^L$, where each vertex subset $V_0^i$ is associated with the dataset $Y^i$. The vertex set $V_0^i$ can be expressed as $V_0^i = \{v_0^{i1}, v_0^{i2}, \ldots, v_0^{iI}\}$. For $i$-th block of $i$-th face image, the vertex $v_0^{ij}$ corresponds to $y^{ij}$. For brevity, take $E_0$ to be a set of undirected edges as $E_0 = \{e_{uv} | u, v \in V_0\}$, where $u$ and $v$ are used to assign any two different feature blocks. $G_0$ is a complete graph.

Obviously, $G_0$ is redundant. We must limit the edge set to make the graph more sparse, that $|E_0|$ is much smaller than $|E_0'|$. For $G_0$, the sparseness can be represented by a binary-valued matrix $P \in \mathbb{B}^{n \times n}$, where $p_{uv}$ is 1 if $u$ and $v$ is connected and $p_{uv} = 0$ otherwise. Then, any element $w_{uv}$ of the (weighted) adjacency matrix of $G_0$ is defined as $w_{uv} = p_{uv} \times w'_{uv}$, where $w'_{uv}$ is an element of $n \times n$ full-(weighted) adjacency matrix.

We have two steps to make $G_0$ (initial sparse graph). Firstly, Let $d$ be a neighborhood threshold. The $d$-neighborhood $N^d_r(u)$ of a vertex $u \in V_0$ is the subset of vertices of $G_0$ at distance at most $d$ from $u$ in $G_0: N^d_r(u) = \{v \in V_0 | dist_{G_0}(u, v) \leq d\}$. And then, mutual $k$-NN [35] graph is included to optimize the set of edges. mutual $k$-NN graphs often produce hubs, or vertices with an extremely high degree (i.e., the number of edges incident to a vertex). These hub vertices indeed deteriorate the accuracy of classification. Then, the mutual $k$-NN graph is a weighted undirected graph connecting each vertex to its mutual $k$-NN in the original sample space. The (weighted) graph is defined as
\(\mathcal{G}_0 = (V_0, E_0), \mathcal{E}_0 = \{e_{uv} | u \in N^k_{\mathcal{G}_0}(v) \lor v \in N^k_{\mathcal{G}_0}(u), u, v \in V\}, N^k_{\mathcal{G}_0}(u) = \{v \in V_0 | v \text{ is } k \text{ nearest vertices from } u\}.

Then, we have an optimization problem as following:

\[
\begin{align*}
\max_{p_{uv} \in \mathcal{E}_0} & \quad \sum_{u,v} p_{uv}' w_{uv}' \\
\text{s.t.} & \quad \sum_{v} p_{uv}' = k, p_{uv}' = 0, \forall u, v \in \{1, 2, \ldots, n\}, \quad (1) \\
& \quad p_{uv} = \min(p_{uv}', p_{vu}').
\end{align*}
\]

Based on Lee [25]'s definition, from a result of Nash-Williams, the graphs of degeneracy at most \(k\) are exactly the \((k + 1)/2, 1)\)-sparse graphs. Thus mutual \(k\)-NN graphs are exactly the \((k + 1)/2, 1)\)-sparse graphs. In \(\mathcal{G}_0\), the graph contains global information across all face images. The similar vertices (blocks) are connected by edges. After sparse reconstruction, we can learn sparse graph \(\mathcal{G}_0\), that is presented in Sect. 4. In the deep learning network, one batch of sparse subgraphs \(\mathcal{G}_l = (V_l', E_l')\) are input of \((r + 1)\)-th sparse graph convolutional layer, that is presented in Sect. 5. The graph variation as shown in Fig. 3.

3.3 Weight Detail

For brevity, we suppose two different vertices \(u\) and \(v\) correspond to two vectors \(y_u\) and \(y_v\). If \(w_{uv} > 0\), we suppose there is an edge \(e_{uv}\) between two vertices. In other words the corresponding feature vectors belong to the same category. The edge set should be redefined by mutual \(k\)-NN. Next, we have to calculate the \(w_{uv}\).

Here, the mutual \(k\)-NN is 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 \(V_0\), if \(u\) (or \(v\)) is among the mutual \(k\)-NN of \(v\) (or \(u\)), the weight \(w_{uv}\) is introduced by a Gaussian kernel function as follows:

\[
w_{uv} = \begin{dcases} 
e^{-\frac{(v-u)^2}{2\sigma^2}}, & \text{if } u \in N_k(v) \text{ and } v \in N_k(u), \\
0, & \text{otherwise}. \end{dcases}
\]

where \(\sigma\) is the bandwidth parameter of the kernel. After this setting, the graph is undirected and the adjacent relations between two vectors is symmetric.

3.4 Other State-of-the-Art Face Graphs

Here, we also test the performance of two frequently face graphs. The first is EGB (Elastic Bunch Graph) from Elastic Bunch Graph Matching [36]. Elastic Bunch Graph Matching is an algorithm in computer vision for recognizing objects or object classes in an image based on a graph representation extracted from other images. The visual features used are based on Gabor wavelets, which have been found similar to simple cells in the brain primary visual cortex. However, EBGM can only be applied to objects with a common structure, such as faces in frontal pose, sharing a common set of landmarks like the tip of the nose or the corner of an eye.

The second is a face landmarking graph. Face landmarking, defined as the detection and localization of certain characteristic points on the face. Many algorithms have been proposed [37]–[40]. We define a face landmark as a prominent feature. Commonly used landmarks are the eye corners, the nose tip, the nostril corners, the mouth corners, the end points of the eyebrow arcs, ear lobes, nasiona, chin, etc. Landmarks such as eye corners or nose tip are known to be less affected by facial expressions, and are referred to as fiducial points. Some examples as shown in Fig. 7.

Finally, we tested and compared these two traditional face maps as show in Table 1. From Table 1, we can look that the performance of these two face graph types is not as good as mutual \(k\)-NN face graph. We estimate that the reason is the local texture loss in the training process.

4. Sparse Reconstruction Graph \(\mathcal{G}_0\)

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 \(Y = (y_1^{l_1}, y_2^{l_2}, \ldots, y_N^{l_I}) \in \mathbb{R}^{M \times (L \times I)}\), where each column represents a data vertex \(y_i^{l_j} \in \mathbb{R}^M\) on the graph. Let \(D = (d_1, d_2, \ldots, d_N) \in \mathbb{R}^{M \times N}\) be a common dictionary. Here \(d_n\) is the basic function with \(N \gg M\) (obviously, solution of \(Y = D S\) 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 \(S = (s_1, s_2, \ldots, s_N) \in \mathbb{R}^{N \times (L \times I)}\). The optimization problem can be written as follows:

\[
\min_{S, \lambda} \frac{1}{2} \|Y - DS\|^2_F + \lambda \|S\|_1, \quad (3)
\]

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

The sparse coefficient $S$ is regularized by the $l_1$-norm, sparsity is estimated by blocks in $D$. The structure of sparse graph is missing in Eq. (3). 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 $Y$, which is reflected in the correlation between the rows in $S$ by embedding function $\frac{1}{2} \sum_{e_{uv} \in E_{\text{struct}}} w_{uv} ||s_u - s_v||^2_w = ||SAS^T||_{l_2}$, where $\Lambda \in \mathbb{R}^{(L \times L) \times (L \times L)}$ is an unnormalized graph Laplacian matrix and $||.||_{l_2}$ is trace norm. Obviously, $\Lambda = U - W$ is an symmetric positive-semidefinite matrix, where $U = \text{diag}(\sum_{w \in \mathbb{W}} w_{um})$ is the degree matrix; $W = (w_{um})$ is the symmetric weight matrix.

In order to integrate the graph structure into our sparse coding, the loss function Eq. (3) can be rewritten as follows:

$$\min_{s,y,\Lambda} \frac{1}{2} ||Y - DS||^2_F + \lambda ||S||_{l_2,1} + \gamma ||SAS^T||_{l_2}.$$

where $\gamma$ is the regularization parameter and $||.||_{l_2,1}$ 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 $s_0^{l,i}$ corresponds to $\hat{y}^{l,i} = Ds_0^{l,i}$. Table 1 shows the performance improvement of the sparse reconstruction.

5. Sparse Graph Based Deep Learning Network from $G_1$ to $G_R$

In this section, we describe the proposed sparse graph based deep learning network 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 [19], GoogLeNet [16], ResNet [18]) 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 [34]. Here, we use the graph to describe the spatial correlation between the vertices and perform convolution by the multiplication in the spectral graph domain.

### 5.1 Network Structure

As we discussed, a sparse graph $G'_l = (V'_l, E'_l)$ means sparse reconstructed mutual k-NN graph of $l$-th face image that consists of vertices $V'_l$ and the edges $E'_l$, $w_{uv}$ is the weight of $e_{uv}$ between two vertices $u$ and $v$, and each vertex $v^{l,i}$ corresponds to vector $\hat{y}^{l,i}$. For $l$-th face image, $\Lambda = \hat{U} - \hat{W}$ is an unnormalized Laplacian matrix; $\hat{U} = \text{diag}(\sum_{w \in \mathbb{W}} w_{um})$ is the degree matrix; $\hat{W} = (w_{um})$ is the symmetric weight matrix. Since $\Lambda$ is a symmetric positive-semidefinite matrix that admits an eigenvalue decomposition $\hat{\Lambda} = \Phi \Lambda \Phi^T$, where the orthonormal eigenvectors $\Phi = (\phi^{l,1}, \phi^{l,2}, \ldots, \phi^{l,Q})$, and $\Lambda = \text{diag}(\Lambda^{l,1}, \Lambda^{l,2}, \ldots, \Lambda^{l,Q})$ is the diagonal matrix of the corresponding non-negative eigenvalues. The graph Fourier transform of $\hat{y}$ is then defined as $\Phi^T \hat{y}$, and its inverse as $\Phi (\Phi^T \hat{y})$. In the Fourier domain, a convolution operator on graph $*_{\rho}$ is defined as $f *_{\rho} \hat{y} = \Phi (f(\Phi^T \hat{y})) \odot (\Phi^T \hat{y}) = \Phi (f \odot (\Phi^T \hat{y}))$, where $\odot$ is the element-wise Hadamard product. Now, we can describe a convolution construction from layer $r$ to layer $(r + 1)$, without pooling layer ($R = 5$ in this paper):

$$y_r^{l,q} = \rho (\sum_p^Q \Phi^{r,p}_{l,q} F^{r,p}_{l,q} \odot \Phi^T \hat{y}_r),$$

where, $p = 1, 2, \ldots, P$ is an index of $r$-layer vector $y_r^{l,p}$, $q = 1, 2, \ldots, Q$ is an index of $(r+1)$-layer vector $y_{r+1}^{l,q}$, $F^{r,p}_{l,q}$ is a diagonal matrix of spectral multipliers representing a learned filter in the frequency domain, and $\rho$ is a nonlinearity applied to the vertex-wise function values.

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

$$\min_F \Sigma_{l=1}^L \ell(f(Y_l), O_l) + \kappa ||F||^2_{l,r,F},$$

where $f(Y_l)$ is an output of the network, $O_l$ is the ground-truth label defined as $O_l = (o_{1l}, o_{2l}, \ldots, o_{cl}) \in \mathbb{B}^l$ and $o_{cl} = 1$ 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:

$$\min_F \Sigma_{l=1}^L \ell(f(Y_l), O_l) + \eta ||W_l||_1 + \nu ||W_l||_1 + \kappa ||S_r^{l=1}||F||^2_{l,F}.$$
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. Our learning network 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. Our network is based on a traditional Siamese network as shown in Fig. 4, and a convolution construction as shown in Fig. 8.

5.2 Pooling Layer Design

In this time, the vertices of the input graph are not arranged in any meaningful way. That would result in a memory inefficient, slow, and hardly parallelizable implementation. We can arrange the vertices such that a graph pooling operation becomes as a CNN pooling. We have two rules:

1. After pooling, each node either have two parent node (e.g. vertex 1 of pooling layer 1/2 in Fig. 9, its parent node is vertex 1 and 2 of pooling layer 0); one singleton and a null node as parent (e.g. vertex 6 of pooling layer 1/2, its parent node is vertex 10 of pooling layer 0)

2. the vertex of largest degree is selected from two nearest vertices (e.g. vertex 1 of pooling layer 1/2 is inherited from vertex 2 of pooling layer 0)

Then, arbitrarily ordering the nodes at the coarsest level, then propagating this ordering to the finest levels, i.e. node n has nodes 2n and 2n+1 as parent, produces a regular ordering in the finest level. Figure 9 shows an example of the whole process.

6. Experiments

6.1 Implementation Details

In this section, we perform face recognition on benchmark face databases, including LFW and YTF to demonstrate the performance of our learning network. Generally, we follow the same strategy used in DeepID2+ [15] method - Siamese network and Joint Bayesian model [29]. Here, the CeleFaces+ dataset [12] and the WDRf dataset [29] 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.

For regularization-parameters, we use a 5-fold cross
validation on the training dataset to tune the parameters $\lambda$ and $\gamma$. At first, a face image is represented on a 15 $\times$ 15 2D-blocks, and each block size is 8 $\times$ 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+ [15], validating a set of 2000 people to determine learning rates and training iterations, momentum of 0.9.

As shown in Fig. 4, we use the following notation to describe our network architectures. $SGC(m)$ denote a sparse graph convolutional layer with $m$ feature maps, $FC(m)$ denotes a fully connected layer with $m$ hidden units, $P(m)$ denotes a graph pooling layer of size and stride $m$, activation function is $ReLU max(y, 0)$. After each graph pooling layer, fully connected layer $FC(512)$ is an input of Joint Bayesian model. The final layer is a softmax regression after final FC layer, and mini-batches are of size $S = 100$. Then, the architecture of our network is $SGDLN : SGC(32) - P(4) - SGC(64) - P(4) - SGC(64) - P(4) - SGC(64) - P(4) - SGC(64) - FC(512)$.

We compare the proposed learning network with state-of-the-art methods on face recognition. Including High-dimensional LBP [41], DDML (LBP) [42], DeepFace [13], FaceNet [17], Parkhi’s approach [10], DeepID2+ [15], DeepID3 [27] and CNN-3DMM estimation [43]. Experiments show that proposed learning network has better recognition performance.

### 6.2 LFW Database

Labeled Faces in the Wild (LFW) is a face photograph database for the problem of unconstrained face recognition. The data set contains more than 13,000 images of 1,680 peoples. The facial expressions open or closed eyes, glasses or no glasses, some tilting and rotation, illumination and obstruction also vary. In this case, the 6,000 face pairs of LFW are tested for face recognition.

Proposed learning network achieves higher accuracy 99.61% for face recognition. The accuracy is comparable with previous state-of-the-art methods on LFW are shown in Table 3. To compare with DeepID2+ nets and DeepID3, our network improves approximately 0.14% and 0.08% 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. There are two reasons: Firstly, in DeepID2+, 25-patches of 1 face image are used to train a network. In our case, we just use 1 sparse graph from 1 face image. Secondly, the graph convolutional neural network trains faster than traditional convolutional neural networks.

### 6.3 YTF Database

YouTube Faces (YTF) database also designed for studying the problem of unconstrained face recognition. The data set contains 3,425 videos of 1,595 different people. An average of 2.15 videos are available for each subject. The shortest clip duration is 48 frames, the longest clip is 6,070 frames, and the average length of a video clip is 181.3 frames. In designing YTF data set and benchmarks, we follow the example of the LFW database. Specifically, we randomly extracted 6,000 face pairs from all video frames along with labels indicating the identities of each person. And then, these 6,000 face pairs are tested for face recognition.

For this experiment, all the captured face can be directly obtained by traditional face detection method. We compare to some start-of-the-art methods. From Table 4, we can see that the improvement of our learning network over the DeepID2+ methods is about 1.47% accuracy. The improvement of our learning network over the Deepface methods is about 3.27% accuracy. The proposed learning network could achieve higher recognition rates than the other methods with face features on YTF database. This validates that the sparse graph based deep learning network is more better than these tradition CNN-like learning methods.

### 7. Conclusions

We proposed a sparse graph based deep learning network for face recognition. The experiments validated the high performance of proposed network on the LFW and YTF reference database. The face representations of graph sparse reconstruction are more sparse and robust to background noise and occlusion. The sparse graph based deep learning network is more highly selective to person identities and can reduce the number of parameters without losing the accuracy. This work shows that the deep learning network of a sparse graph is feasible within face recognition, and it is believed that there remains substantial room for extension of this concept. Here, we will try to optimize the directed graph framework based sparse graph deep learning network in future work. This is obviously feasible, also one of the open questions, but the network will be more complex and difficult to optimize in consideration of training costs.
References

[1] X. Wang and X. Tang, “Random sampling for subspace face recognition,” International Journal of Computer Vision, vol.70, no.1, pp.91–104, 2006.

[2] T. Aihonen, A. Hadid, and M. Pietikainen, “Face description with local binary patterns: Application to face recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol.28, no.12, pp.2037–2041, 2006.

[3] N. Kumar, A.C. Berg, F.N. Belhumeur, and S.K. Nayar, “Attribute and simile classifiers for face verification,” Proc. ICCV, pp.365–372, 2009.

[4] Q. Yin, X. Tang, and J. Sun, “An associate-predict model for face recognition,” Proc. CVPR, pp.697–504, 2013.

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

[6] Z. Zhu, P. Luo, X. Wang, and X. Tang, “Deep learning identity-preserving face space,” Proc. ICCV, pp.113–120, 2013.

[7] B.A. Olshausen and D.J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by v1?,” Vision research, vol.37, no.23, pp.3311–3325, 1997.

[8] H. Shan, L. Zhang, and G.W. Cottrell, “Recursive ica,” Advances in neural information processing systems, pp.1273–1280, 2007.

[9] H. Shan and G.W. Cottrell, “Looking around the backyard helps to recognize faces and digits,” Proc. CVPR, pp.1–8, 2008.

[10] O.M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” Proc. BMVC, pp.1–12, 2015.

[11] Y. Sun, X. Wang, and X. Tang, “Hybrid deep learning for face verification,” Proc. ICCV, pp.1489–1496, 2013.

[12] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation from predicting 10,000 classes,” Proc. CVPR, pp.1891–1894, 2014.

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

[14] Y. Taigman, M. Yang, M.A. Ranzato, and L. Wolf, “Web-scale training for face identification,” Proc. CVPR, pp.2746–2754, 2015.

[15] Y. Sun, X. Wang, and X. Tang, “Deeply learned face representations are sparse, selective, and robust,” Proc. CVPR, pp.2892–2900, 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,” Proc. CVPR, pp.1–9, 2015.

[17] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” Proc. CVPR, pp.815–823, 2015.

[18] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proc. CVPR, pp.770–778, 2016.

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

[20] D. Lazer, A.S. Pentland, L. Adamic, S. Aral, A.L. Barabasi, D. Brewer, and T. Jebara, “Life in the network: the coming age of computational social science,” Science, vol.323, no.5915, p.721, 2009.

[21] D.K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R.P. Adams, “Convolutional networks on graphs for learning molecular fingerprints,” Advances in neural information processing systems, pp.2224–2232, 2015.

[22] E.H. Davidson, J.P. Rast, P. Oliveri, A. Ransick, C. Calestani, C.H. Yuh, and O. Otim, “A genomic regulatory network for development,” Science, vol.295, no.5560, pp.1669–1678, 2002.

[23] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral networks and locally connected networks on graphs,” arXiv preprint arXiv:1312.6203, 2013.

[24] V. Pietreer and P.E. Black, “Dictionary of algorithms and data structures,” NIST, 2005.

[25] A. Lee and I. Streinu, “Pebble game algorithms and sparse graphs,” Discrete Mathematics, vol.208, no.8, pp.1425–1437, 2008.

[26] K. Simonyan, O.M. Parkhi, A. Vedaldi, and A. Zisserman, “Fisher vector faces in the wild,” Proc. BMVC, pp.1–12, 2013.

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

[28] 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.

[29] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, “Bayesian face re-visited: A joint formulation,” Proc. ECCV, pp.566–579, 2012.

[30] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, “Gated graph sequence neural networks,” arXiv preprint arXiv:1511.05493, 2015.

[31] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” Neural Information Processing Systems, pp.3844–3852, 2016.

[32] R. Rustamov and L.J. Guibas, “Wavelets on graphs via deep learning,” Advances in neural information processing systems, pp.998–1006, 2013.

[33] L. Mou, G. Li, L. Zhang, T. Wang, and Z. Jin, “Convolutional neural networks over tree structures for programming language processing,” Proc. AAAI, pp.1287–1293, 2016.

[34] M.M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, “Geometric deep learning: going beyond euclidean data,” arXiv preprint arXiv:1610.09097, 2016.

[35] M.R. Brito, E.L. Chávez, A.J. Quiroz, and J.E. Yuchik, “Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection,” Statistics and Probability Letters, vol.35, no.1, pp.33–42, 1997.

[36] L. Wiskott, N. Krüger, N. Kügler, and C.V. D. Malbsburg, “Face recognition by elastic bunch graph matching,” IEEE Trans. Pattern Anal. Mach. Intell., vol.19, no.7, pp.775–779, 1997.

[37] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” Proc. CVPR, pp.2879–2886, 2012.

[38] T. Baltrusaitis, P. Robinson, and L.-P. Morency, “Constrained local neural fields for robust facial landmark detection in the wild,” Proc. ICCV, pp.354–361, 2013.

[39] Z. Zhang, P. Luo, C.C. Loy, and X. Tang, “Facial landmark detection by deep multi-task learning,” Proc. ECCV, pp.94–108, 2014.

[40] Z.-H. Feng, P. Huber, J. Kittler, W. Christmas, and X.-J. Wu, “Random cascaded-regression copse for robust facial landmark detection,” IEEE Signal Process. Lett., vol.22, no.1, pp.76–80, 2015.

[41] D. Shen, X. Cao, P. Wen, and J. Sun, “Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification,” Proc. CVPR, pp.3025–3032, 2013.

[42] J. Hu, J. Lu, and Y.-P. Tan, “Discriminative deep metric learning for face verification in the wild,” Proc. CVPR, pp.1875–1882, 2014.

[43] A.T. Tran, T. Hassner, I. Masi, and G. Medioni, “Regression robust and discriminative 3d morphable models with a very deep neural network,” Proc. CVPR, pp.1493–1502, 2017.
Sei-ichiro Kamata received the M.S. degree in computer science from Kyushu University, Fukuoka, Japan, in 1985, and the Doctor of Computer Science, Kyushu Institute of Technology, Kitakyushu, Japan, in 1995. From 1985 to 1988, he was with NEC, Ltd., Kawasaki, Japan. In 1988, he joined the faculty at Kyushu Institute of Technology. From 1996 to 2001, he has been an Associate Professor in the Department of Intelligent System, Graduate School of Information Science and Electrical Engineering, Kyushu University. Since 2003, he has been a professor in the Graduate School of Information, Production and Systems, Waseda University. In 1990 and 1994, he was a Visiting Researcher at the University of Maine, Orono. His research interests include image processing, pattern recognition, image compression, and space-filling curve application. He is a member of IEEE and ITE in Japan.