2020 25th International Conference on Pattern Recognition (ICPR)
Milan, Italy, Jan 10-15, 2021

Siamese Graph Convolution Network for Face Sketch Recognition

- An application using Graph structure for face photo-sketch recognition

Liang Fan
School of Computer Science and Informatics
Cardiff University,
Cardiff, CF243AA UK
FanL7@cardiff.ac.uk

Xianfang Sun
School of Computer Science and Informatics
Cardiff University,
Cardiff, CF243AA UK
sunx2@cardiff.ac.uk

Paul L. Rosin
School of Computer Science and Informatics
Cardiff University,
Cardiff, CF243AA UK
rosinpl@cardiff.ac.uk

***Abstract*—** In this paper, we present a novel Siamese graph convolution network (GCN) for face sketch recognition. To build a graph from an image, we utilize a deep learning method to detect the image edges, and then use a superpixel method to segment the edge image. Each segmented superpixel region is taken as a node, and each pair of adjacent regions forms an edge of the graph. Graphs from both a face sketch and a face photo are input into the Siamese GCN for recognition. A deep graph matching method is used to share messages between cross-modal graphs in this model. Experiments show that the GCN can obtain high performance on several face photo-sketch datasets, including seen and unseen face photo-sketch datasets. It is also shown that the model performance based on the graph structure representation of the data using the Siamese GCN is more stable than a Siamese CNN model**.**

Keywords— Siamese network ; graph convolution network (GCN) ; superpixel ; graph structure

I. INTRODUCTION

A suspect's image on an announcement or the news may be not a captured photo of the criminal, but a facial sketch which is drawn from the description of victims or witnesses. These sketches as a unique clue are used to retrieve the corresponded photos from the police's face photo dataset, or to search for the suspect by publishing the sketch. The portrait sketch which is drawn from a frontal human photo can express the quality of human skin, the structural features arising due to different ages and genders, etc. However, for forensic sketch artists, the suspect's sketch is generated by a simple description from victims or witnesses. Since victims or witnesses cannot remember all face attribute details, such as the shape of the nose or ears, or eye colour, the painter has to draw the sketch through a combination of their imagination and experience. This means that standard face recognition systems cannot be used to search a face sketch from a face photo dataset. The earliest algorithm [1] is designed to directly generate a pseudo photo or sketch according to the corresponding image using the Karhunen–Loeve Transform for face photo-sketch recognition. Subsequently, a nonlinear dimension reduction method [2] is proposed to generate more real looking sketch photos for increasing recognition accuracy. A feature descriptor called local feature-based discriminant analysis (LFDA) [3] to extract features from the face photo and face sketch. Then the extracted features are used to compare the distance between the target sketch and a set of photos. The multi-view discriminant analysis method maps face photos and sketches into a common space to increase the generalization of the classifier.

Instead of traditional methods, deep learning is used to reduce the effect of sketch distortion and noise. Galea and Farrugia proposed a DCNN model [33] for face verification based on a triplet network to extract feature representations from face sketches and face photos. Due to the weight-shared triplet structure, this model allows flexibility in the variation of the facial features.

The existing face photo-sketch recognition methods based on deep learning avoid spatial topology structure from face images, such as skin, race, or hair colour. A new loss function [4] which combine a transformation and mapping function is used to fuse the facial attributes and the geometrical properties of sketch from a deep two-channel coupled CNN framework. The new loss function consists of three parts, one is to minimize the intra-class distances of photos or sketch-attribute pairs, the middle is for intra-class separability, last one is to keep a maximize distance behind the similarity inter-class samples. Generative Adversarial Network (GAN) [5] is a novel strategy to improve recognition accuracy based on the synthesis-based method. It designs a new loss function that combines with the advantages of CycleGAN and the conditional GANs method to ensure the pseudo image's quality for recognition. These methods all get better results on the face photo-sketch dataset. However, in order to use the spatial relationship of face attributes to show a realistic face image, some basic facial features will be discarded to avoid distortion which is produced by rich facial local changes and special illumination. This causes the feature extractor to fail to extract valid features which are similarity features with face photo for recognition. Moreover, the variations in the thickness and lightness of the lines increases the sketch's noise. It leads to the similarity for the extracted features

978-1-7281-8808-9/20/$31.00 ©2020 IEEE

2020 25th International Conference on Pattern Recognition (ICPR) | 978-1-7281-8808-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPR48806.2021.9412917

between different persons being higher than between the same person.

In this paper, we propose to extract graphs from images to reduce the effect of uncontrollable factors, such as illumination, expression. Then, a Siamese graph convolution network (GCN) is designed to learn an embedding space. Finally, we use the contrastive loss function to optimize the node and edge information and compare the similarity between each pair of graphs. The contributions of our paper are as follows:

(1) We utilize a CNN and two types of superpixel methods to generate a graph structure from face photos and face sketches. Firstly, a full convolution network is used to extract image edges from photos and sketches. The superpixel methods are then used to cluster similar pixels into small regions. After the features are extracted from each cluster region, a graph is built that contains face contour information.

(2) A Siamese GCN is designed to transfer a graph structure into an embedding space with the intrinsic structural properties of graphs. In addition, this Siamese GCN can capture the topology of the graph and the relationship among nodes with shared weights and keep similar graph structure and node information for recognition.

(3) We combine a deep graph matching method with GCN and the MoNet network to extract more similar cross-modal graph features than that extracted by the original weight-shared Siamese network. The proposed method uses the contrastive loss function to measure the graph distance based on the Euclidean distance. It can reduce the difference between two graphs of the same class from different modalities.

II. Related Work

A. Siamese Networks

Deep convolutional networks extract features using filters and generated model samples from the same probability distribution. The advantage of convolution networks for recognition is that they utilize sparsely connected methods to transfer information into the next layer, so that it can avoid the extracted features being affected by other regions, and ensure that each image has unique features. However, because of variations in illumination and viewpoint, it leads to the extracted features being different, which is problematic for recognition. Meanwhile, convolution networks are used on datasets of different modalities, such as face photo and face sketch images. Due to the differences in imaging principles between different types of images, the data distribution of different modalities varies greatly. To extract corresponding features from data of different modalities, a Siamese network is usually used, which consists of two identical convolution networks and extracts similarity features from two channels of neural networks with shared parameters to learn the distribution of similarity features from the same object.

The first Siamese network [6] was proposed to learn a similarity metric from pairs of input face images. The main idea of this model is that pairs of input face images are mapped into a target space which can minimize the distance between intra-class data and maximize the distance between inter-class data. The advantage is that the Siamese network, label represents same sample or different sample and can achieve high performance for small datasets using deep learning. An improved Siamese convolution network model [7] was built for face verification. This model fuses the convolution and subsampling operations to reduce the complexity and the number of parameters. It is difficult to learn a suitable feature using deep learning method for small datasets. The Siamese network model [8] adopts a score system to rank similarities of nonlinear features between input data using shared-weight CNNs. After training the model, the powerful discrimination function of this model is applied not only for new data, but also for new categories in unknown class data. However, the margin in the contrastive loss function must be a constant value for every inter-class samples, which restricts the ability of further improving recognition accuracy.

B. Graph Convolution Network

The target of the GCN model is to extract spatial features from a graph which is built using the relationship between nodes and edges. The first GCN was proposed by [11]. The advantage of this GCN is that it extracts features from various graph structure data, especially from weakly connected graphs. However, this design cannot achieve shared weight strategy for different position on a graph. Different from the first GCN [11], a set of parameters is added into the convolution kernel in the GCN proposed [12] to reduce the complexity of parameters. GCNs can be used to extract information on first-order neighbours in the graph [13]. Meanwhile, for each node, the principle of graph convolution filters is similar to the filters on CNN model. The s-GCN [14] utilizes graph Siamese network to reveal the relationship between a specific disease and brain structure using graph representation. A novel SGCN [15] utilize graph convolution network to learning the similarity between each image which is represented by the region adjacency graph (RAG). Then, PCA-GM model [16] adopts graph structure based on embedding to seek the relationships between nodes for matching.

A graphical representation based on markov networks [17] is proposed for face photo-sketch recognition. They use Markov networks to select a set of nearest image patch from overlapping photo image patches and overlapping sketch patches based on a coupled representation similarity metric. The advantage is that the Markov network extracts the spatial features for recognition. The deep sparse graph neural network (DSGNN) [18] extracts an undirected graph G from the face photo image for recognition. The graph nodes are the divided blocks for each face image. Undirected edges are generated using the Euclidean distance to calculate the correlation between pairs of image patches. After learning features using deep sparse graph neural networks, the recognition accuracy reaches 99.5% on the LFW dataset. However, this method is sensitive to the effects of

occlusion and illumination when extracting features using the CNN model to generate the graph structure data. The GCN model [19] is utilize to predict a new node from existed graph model. They utilize a KNN to build a graph structure after extracting face features using CNN model. Then the similarity nodes are clustered by GCN using the weighted average between adjacent nodes and neighbour nodes. This method supposes that if two face images have the same ID, there is a connectivity after inference from graph. A hierarchical multigraph network [20] is built to improve the graph classification accuracy on image datasets. In the first step, the graph for an image is built by superpixels of the images. This method builds a three-layers graph convolution network on graph data to extract node information for increasing recognition accuracy of low-resolution image dataset.

III. METHODOLOGY

This paper utilizes graph structure data as input to reduce the modality gap between photos and sketches for face recognition based on a Siamese network. The architecture of our model is shown in Fig. 1.

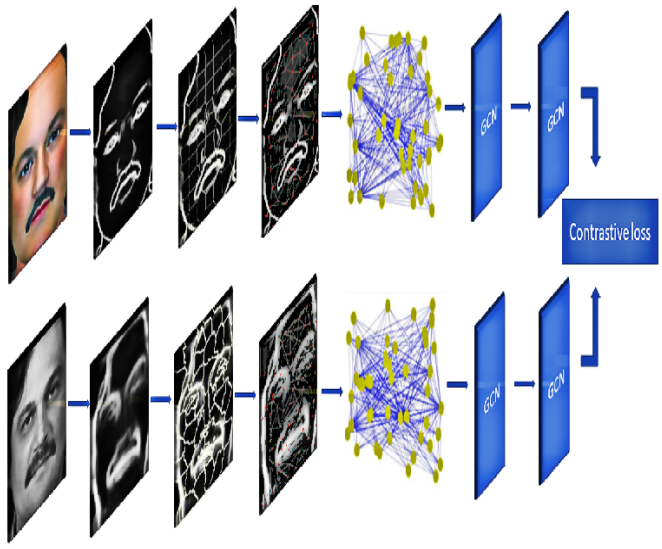

Fig. 1. Architecture of the Siamese graph network model. Each branch of our model consists of two graph layers for extracting graph features.

The input graph structure data is generated from images using superpixel methods [21] [22]. Firstly, the holistically-nested edge detection (HED) method [23] is used to generate an edge image for image information simplification. Then the superpixel method is used on the edge image to segment the image into regions. A graph is generated by taking the centre of each region as a node and the distance between each pair of the nodes as the edge feature. After that, a set of input data which is composed of two graphs, including one from the sketch image and one from the photo image, is input into our Siamese network model. Each channel in the Siamese network model consists of two graph convolution layers to extract features from the graph on an embedding space. Finally, Euclidean distance is used in the contrastive loss function to measure the distance between each reconstructed graph for recognition. The aim of our model is to measure the similarity between the graphs for increasing recognition accuracy.

A. Graph Structure Data for Images

Photos are generated by optical imaging principles. A face photo uses the relationship between pixels to describe all features of the real human face on a two-dimensional space. In contrast, a face sketch uses the geometric deformation and the line density to represent the illumination and the characteristics of the real face. Since the representations are different between photos and sketches, convolutional networks can only capture the local structure of the sketch, but cannot fully extract colour and texture information. It fails to represent the same model features as the photo, because the formation of sketch leads to the extracted features from sketch lack 'sense of feature'. Therefore, we propose to build a graph structure based on the image features and structure information, as is shown in Fig. 2.

In the first step, we utilized an edge detection method to extract the contour of the face image and reduce the background noise. It not only keeps the image structure properties, but also reduces the weakly relevant information.

Traditional edge detection methods, such as Sobel, Prewitt, Canny, HOG, utilize local region changes, including colour changing and illuminations, to search image edges. However, sketches use lines of different widths to represent texture features, which leads to that some facial details cannot be represented. The low-level features extracted by traditional methods do not reflect the real sketch edges. Moreover, it is difficult to extract colour, illumination, and gradients from sketch images to detect edges because texture features in sketch images have weak edge distribution patterns.

CNN-based methods [24] utilize kernels of large receptive fields to extract global feature and details from images and pooling layers to increase the recognition accuracy. Large receptive fields and pooling layers in the low convolution layers remove more details than the high convolution layers. Hence, the low convolution layers focus on extracting the edge features, and the high convolution layers focus on global semantic features. Thus, we can use deep learning method on face photo and sketch images to obtain the image contours from a high convolution layer which includes more semantic information than a low convolution layer.

Meanwhile, the HED network [23] combines multi-scale features with a multi-level feature to map several multiple side output layers on the main convolutional network. It obtains a set of edges from different scales. The drawback of the HED network is that this model adopts many downsampling layers and does not fully fuse multi-scale features, producing rough and fuzzy lines as edge detection results.

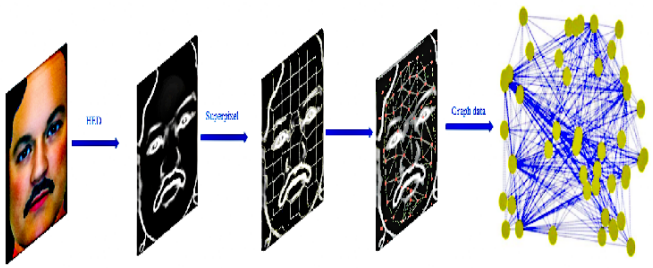

Fig. 2. The pipeline of creating graph structure data from an image. The first step is to extract image edges using the holistically-nested edge detection method. Next, a superpixel segmentation of the edge image is generated. Then, a region adjacency graph is built based on the superpixel segmentation.

Then we use the superpixel method to segment image as for build graphs, such as Quickshift [21] and SLIC [22] method, to cluster adjacent pixels with similar features into the same region. The superpixel methods aggregate similar image pixels to several sub-region blocks with regular shapes and consistent local structure, according to the similarity criteria of image visual features and spatial distance. This method tends to global representation using the image's local features and structure information to reduce data redundancy. According to the correlation between each pixel, the pixel colors, and the similarity of brightness in the image edge, we use superpixel methods.

Each superpixel region is used as a node in an undirected graph structure. The relationships between adjacent regions, which is obtained using feature's information, is mapped as graph edge. The image is mapped as a weighted undirected graph $G(V,E)$. In graph G, $v=\{1,2,\cdots,N\}$ is the regions using superpixels for image. E is the similarity between adjacent regions. The weights of corresponding edges $W(v_i,v_j)$ are the difference between the region features.

B. Graph Convolution Network

We adopts two strategies: GCN and MoNet [25] on our graph data for recognition. For the input undirected graph data $G(V,E)$ with N nodes V and edge E, we utilize Graph Convolutional Layer on graph-structure data to extract features. The core of GCN for convolution is to use the Fourier transform on a graph as:

$$f * g_\theta = U * diag(U^T g_\theta) U^T f \tag{1}$$

where g_θ is the convolutional kernel, $U^T g_\theta$ and $U^T f$ represent the Fourier transformation of g_θ and f respectively, induced from the Laplace matrix of the graph. $U=(U_1,U_2,\ldots,U_n)$ are the orthonormal eigenvectors of the Laplacian matrix. The core of GCN is that the eigenfunction of the Laplacian is transferred to the eigenvector of the Laplacian matrix calculated from graph G. The node feature of each layer in the GCN network is composed by the convolution of signals.

Then an activation function is used to perform a nonlinear transformation to obtain a matrix that aggregates features of adjacent vertices for generating a new node representation. According to the principles of convolutional networks, GCN utilizes some overlapping convolution layers to achieve multi-order neighbourhood information for update. In our model, we adopt GCN to extract the first-order neighbourhood information from graphs using graph topology directly. For the l^{th} layer of GCN model, the extracted message $f_{message}^{(l+1)}$ from this layer is represented as:

$$f_{message}^{(l+1)} = w_0^l f_i^l + \sum_{E_{i\in N(V_I)}} w_1^l f_i^l \tag{2}$$

where $N(V_i)$ is a set of nodes which connect with V_i in graph $G(V,E)$, w_0^l and w_1^l are weights of nodes.

The original Siamese network uses the contrastive loss function to calculate the similarity between two graphs in an embedding space. Instead of mapping graph into a vector space, we use the graph matching networks [26] to update nodes of our graph network model. It receives and clusters the information between neighbouring nodes for each selected node, and fuses the local graph structure information. This method not only aggregates messages on the edges of each graph, but also changed the way of update for nodes in each propagation layer using a cross-graph matching vector. This cross-graph matching vector can measure the matching degree between nodes in one graph with several nodes in another graph.

Another strategy is to build graph layers based on the MoNet method [25]. MoNet introduces node pseudo-coordinates to determine the relative position between a node and its neighbours in D-dimensions. The convolution calculation of a node x is defined in MoNet as:

$$(f * g)(x) = \sum_{j=1}^{J} g_j D_j(x) f \tag{3}$$

where f is the signal on the graph, g is the convolutional kernel with dimension J, g_i is the j^{th} element of g, and $D_j(x)f$ is a weighted sum of the signal on x's neighbouring nodes, where the weight is dependent on the pseudo-coordinates of each neighbour.

IV. Experiment Setting

We utilize the MTCNN model [27] to detect the location of face images and crop all face images to size 128*128. After extracting image edges, the contours of the images are represented by grayscale images. After that, we build and test four models, including two Siamese networks based on GCN using SLIC and Quickshift, and two Siamese networks based on MoNet using SLIC and Quickshift, respectively.

In detail, we use SLIC and Quickshift to build graphs for all face photos and face sketches respectively. For SLIC, we extract N<100 superpixels; each superpixel region can be represented as a node, an edge value is computed as the spatial

distance between the superpixel regions. For Quickshift, the kernel size is 2. Then we build GCN and MoNet as layers in our Siamese network. For MoNet, we use (4) to compute the pseudo-coordinate between two nodes.

$$u(x,y) = \left(\frac{1}{\sqrt{\deg(x)}}, \frac{1}{\sqrt{\deg(y)}}\right) \quad (4)$$

In our model, two graphs from a sketch and a photo respectively, are input into the Siamese network. Then loss function in our model is contrastive loss which compares the similarity between pairs of input graph data.

$$Loss = \frac{1}{2N}\sum_{(n=1)}^{N} y_n d_n^2 + (1-y_n)max(L - d_n^2, 0)^2 \quad (5)$$

where y_n is the label for each input pair. $y_n = 0$ represents the pair in the same class, while $y_n = 1$ represents the pair in different classes. L is a margin to measure the distance between same class and different class data. In order to increase the distance between different class data, we use the squared Euclidean distance to measure the difference of two samples. We train this Siamese graph network using the Adam optimizer. The learning rate is set as 1e-6.

V. RUSULTS

Because composite face sketches are widely applied in forensics for face recognition, we use three composite face photo-sketch datasets which have different characteristics (UoM-SGFSA, UoM-SGFSB, and e-PRIP dataset) to test our models. The UoM-SGFS dataset [28] contains two types of face colour sketches, UoM-SGFSA and UoM-SGFSB. The similarity between photos and sketches in the UoM-SGFSB dataset is higher than that in UoM-SGFSA, where SetA is created using Corel Paintshop Pro X7 with EFIT-V software to reduce the similarity with the corresponding photo than SetB. The viewed face sketches in e-PRIP dataset [29] are generated by the corresponding photograph which is from the AR dataset using FACES software. The quality of the generated sketch from the FACES software is closer to photo quality. The categories of face attributes are limited in the software. The generated viewed sketch can be recognized from the shape and proportional difference between face attributes.

We compare the performance of our models with some state-of-the-art models ([33], [30], [29]). Tables I ~ III show the performance comparison on the three different composite face sketch datasets. Compared with all the other methods, the Siamese model with GCN and Quickshift gets the best Top-1 accuracy on UoM-SGFSA and e-PRIP datasets, where its recognition accuracy achieves 74.16% and 55.28%, respectively. However, the recognition accuracy of our models is lower than the result in [30] on UoM-SGFSB dataset.

TABLE I. EXPERIMENTAL RESULTS ON UOM-SGFSA DATASET

Method	Top-1 accuracy	Top-10 accuracy
[30]	64.80%	92.13%
[31]		68.3%
[32]		96.7%
DCNN [33]	31.60%	66.13%
Siamese GCN (Quickshift)	74.16%	76.66%
Siamese MoNet (Quickshift)	64.17%	74.17%
Siamese GCN (SLIC)	68.33%	72.25%
Siamese MoNet (SLIC)	66.65%	73.33%

TABLE II. EXPERIMENTAL RESULTS ON UOM-SGFSB DATASET

Method	Top-1 accuracy	Top-10 accuracy
[30]	72.53%	94.80%
[32]		96.13%
DCNN [33]	52.17%	82.67%
Siamese GCN (Quickshift)	65%	80.83%
Siamese MoNet (Quickshift)	62.5%	80%
Siamese GCN (SLIC)	60.83%	77.5%
Siamese MoNet (SLIC)	59.1%	79.17%

From the tables we can see that the performance using the Quickshift method is better than that using the SLIC method. The SLIC algorithm uses K-means clustering to obtain superpixel regions under an average distribution of cluster centres. Because of ignoring the image edge information, it leads to inaccuracy in the segmentation results of superpixel blocks. Compared with Quickshift algorithm which makes superpixels segmentation based on the image's Color density, the advantage of SLIC method does not only segment color image, but also compatible to implement on a grayscale image. Different regions are classified into the same superpixel block, producing under-segmented superpixel blocks. From the tables we can also see that the performance of GCN is better than that of MoNet. The GCN trains all the nodes in the graph to obtain a new graph representation in the embedding space. A new graph presentation is extracted using the generated graph from the last graph convolution layer and the optimized node embedding. However, the representation of each node is affected by all related nodes. It may lead to the effect that the graph convolution network is worse than the convolution network.

TABLE III. EXPERIMENTAL RESULTS ON E-PRIP DATASET

Method	Top-1 accuracy	Top-10 accuracy
[29]	52%	60.20%
DCNN [33]	54.90%	80.80%
AADCNN with attributes [34]		76.4%
AADCNN without attributes [34]		69.1%
[35]		91.73%
Siamese GCN (Quickshift)	55.28%	73.9%
Siamese MoNet (Quickshift)	50.4%	67.48%
Siamese GCN (SLIC)	47.15%	63.4%
Siamese MoNet (SLIC)	48.78%	61.78%

We also test our model performance on a hand-drawn face photo-sketch dataset. The sketches in the CUFS [36] and CUFSF [37] datasets are all drawn by artists according to the corresponding front photos. Hand-drawn sketches utilize the facial contour and the ratio between the locations of facial features to depict the intrinsic features. The drawn images typically have some distortions and inaccuracies on a character's position, illumination and pose, because of natural limitations of the artists. We use graph structure to obtain the relationship between pixels on the image edge to avoid the effect of image's distortion. However, HED method cannot extract suitable image edges to build graph data. The performance achieves 87.71% and 82.25% for the CUFS dataset and CUFSF dataset, respectively. Although the results all exceed 80%, the dataset is too small for recognition. The results on hand-drawn face photo-sketch datasets do not show the real performance of our model.

TABLE IV. EXPERIMENTAL RESULTS ON CUFSF DATASET

Method	Top-1 accuracy
[38]	80.80%
DCNN [33]	82.80%
[29]	52%
Siamese GCN (Quickshift)	82.25%
Siamese MoNet (Quickshift)	80.75%
Siamese GCN (SLIC)	77.5%
Siamese MoNet (SLIC)	75.5%

TABLE V. EXPERIMENTAL RESULTS ON CUFS DATASET

Method	Top-1 accuracy
Siamese GCN (Quickshift)	87.71%
Siamese MoNet (Quickshift)	85.9 %
Siamese GCN (SLIC)	82.4 %
Siamese MoNet (SLIC)	78.9%

VI. CONCLUSION

In this paper, we present a Siamese network based on graph structure data for face photo-sketch recognition. This model constructs two graph convolution layers for each channel to learn a set of graphs on an embedding space. In order to reduce the modality gap between face photos and sketches, we utilize a superpixel method on the contour images obtained from the HED model to extract a similar graph structure data from the sketch and the correspond photo. Experiments show that the similarity is higher between graph data of face photos and sketches using the Quickshift method than using SLIC. We test our methods on composite face photo-sketch datasets and hand-drawn face photo-sketch datasets. For composite face photo-sketch datasets, the Top-1 recognition accuracy for the UoM-SGFSA dataset is better than the state-of-the-art methods and reaches 74.16%. For hand-drawn face photo-sketch datasets, the performance is better than the results on composite face photo-sketch datasets

REFERENCES

[1] X. Tang and X. Wang, "Face sketch recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 50–57, 2004.

[2] Q. Liu, X. Tang, H. Jin, H. Lu, and S. Ma, "A nonlinear approach for face sketch synthesis and recognition," in *Computer Vision and Pattern Recognition (CVPR), 2005. IEEE Computer Society Conference on*, 2005, vol. 1, pp. 1005–1010, Accessed: Apr. 04, 2017. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/1467376/.

[3] K. P. Siddharth and D. R. Kisku, "Heterogeneous Face Identification by Fusion of Local Descriptors," in *Advance Computing Conference (IACC), 2017 IEEE 7th International*, 2017, pp. 886–894.

[4] S. Nagpal, M. Singh, R. Singh, M. Vatsa, A. Noore, and A. Majumdar, "Face sketch matching via coupled deep transform learning," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5419–5428.

[5] S. Yu, H. Han, S. Shan, A. Dantcheva, and X. Chen, "Improving Face Sketch Recognition via Adversarial Sketch-Photo Transformation," in *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, 2019, pp. 1–8.

[6] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Computer Vision and Pattern Recognition (CVPR), 2005. IEEE Computer Society Conference on*, 2005, vol. 1, pp. 539–546.

[7] M. Khalil-Hani and L. S. Sung, "A convolutional neural network approach for face verification," in *2014 International Conference on High Performance Computing & Simulation (HPCS)*, 2014, pp. 707–714.

[8] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, 2015, vol. 2.

[9] Q. Yu, Y. Yang, F. Liu, Y.-Z. Song, T. Xiang, and T. M. Hospedales, "Sketch-a-net: A deep neural network that beats humans," *International Journal of Computer Vision*, vol. 122, no. 3, pp. 411–425, 2017.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2015, pp. 815–823.

[11] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *ArXiv Prepr. ArXiv13126203*, 2013.

[12] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in

Advances in neural information processing systems, 2016, pp. 3844–3852.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ArXiv Prepr. ArXiv160902907*, 2016.

[14] S. I. Ktena *et al.*, "Metric learning with spectral graph convolutions on brain connectivity networks," *NeuroImage*, vol. 169, pp. 431–442, 2018.

[15] U. Chaudhuri, B. Banerjee, and A. Bhattacharya, "Siamese graph convolutional network for content based remote sensing image retrieval," *Computer vision and image understanding*, vol. 184, pp. 22–30, 2019.

[16] R. Wang, J. Yan, and X. Yang, "Learning combinatorial embedding networks for deep graph matching," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3056–3065.

[17] C. Peng, X. Gao, N. Wang, and J. Li, "Graphical representation for heterogeneous face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 301–312, 2017.

[18] R. Wu, S. Kamata, and T. Breckon, "Face recognition via deep sparse graph neural networks," 2017.

[19] Z. Wang, L. Zheng, Y. Li, and S. Wang, "Linkage based face clustering via graph convolution network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1117–1125.

[20] B. Knyazev, X. Lin, M. R. Amer, and G. W. Taylor, "Image Classification with Hierarchical Multigraph Networks," *ArXiv Prepr. ArXiv190709000*, 2019.

[21] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *European conference on computer vision*, 2008, pp. 705–718.

[22] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[23] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[24] R. He, X. Wu, Z. Sun, and T. Tan, "Wasserstein CNN: Learning Invariant Features for NIR-VIS Face Recognition," *ArXiv Prepr. ArXiv170802412*, 2017.

[25] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5115–5124.

[26] Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli, "Graph matching networks for learning the similarity of graph structured objects," *ArXiv Prepr. ArXiv190412787*, 2019.

[27] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[28] C. Galea and R. A. Farrugia, "A large-scale software-generated face composite sketch database," in *Biometrics Special Interest Group (BIOSIG), 2016 International Conference of the*, 2016, pp. 1–5.

[29] P. Mittal, M. Vatsa, and R. Singh, "Composite sketch recognition via deep network-a transfer learning approach," in *Biometrics (ICB), 2015 International Conference on*, 2015, pp. 251–256, Accessed: Apr. 04, 2017. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7139092/.

[30] C. Peng, N. Wang, J. Li, and X. Gao, "DLFace: Deep local descriptor for cross-modality face recognition," *Pattern Recognition*, vol. 90, pp. 161–171, 2019.

[31] D. Liu, J. Li, N. Wang, C. Peng, and X. Gao, "Composite components-based face sketch recognition," *Neurocomputing*, vol. 302, pp. 46–54, 2018.

[32] X. Xue, J. Xu, and X. Mao, "Composite Sketch Recognition Using Multi-scale Hog Features and Semantic Attributes," in *2019 International Conference on Cyberworlds (CW)*, 2019, pp. 121–127.

[33] C. Galea and R. A. Farrugia, "Matching software-generated sketches to face photographs with a very deep CNN, morphed faces, and transfer learning," *IEEE Trans. Inf. Forensics and Security.*, vol. 13, no. 6, pp. 1421–1431, 2017.

[34] S. M. Iranmanesh, H. Kazemi, S. Soleymani, A. Dabouei, and N. M. Nasrabadi, "Deep sketch-photo face recognition assisted by facial attributes," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2018, pp. 1–10.

[35] D. Liu, N. Wang, C. Peng, J. Li, and X. Gao, "Deep Attribute Guided Representation for Heterogeneous Face Recognition," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 835–841.

[36] X. Wang and X. Tang, "Face photo-sketch synthesis and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 11, pp. 1955–1967, 2009.

[37] W. Zhang, X. Wang, and X. Tang, "Coupled information-theoretic encoding for face photo-sketch recognition," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 513–520, Accessed: Apr. 04, 2017. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/5995324/.

[38] W. Wan, Y. Gao, and H. J. Lee, "Transfer deep feature learning for face sketch recognition," *Neural Computing and Application*, pp. 1–10, 2019.