A Graph Based Unsupervised Feature Aggregation for Face Recognition

Yu Cheng[1,2], Yanfeng Li[1,3], Qiankun Liu[1], Yao Yuan[1], Venkata Sai Vijay Kumar Pedapudi[1], Xiaotian Fan[1,2], Chi Su[4], and Shengmei Shen[1]

[1]Pensees Pte Ltd
[2]National University of Singapore
[3]Xidian University
[4]Kingsoft Cloud
e0321276@u.nus.edu, liyanfeng@stu.xidian.edu.cn, {allen.liu, yuan.yao, sai.vijay, jane.shen}@pensees.ai, e0320795@u.nus.edu, suchi@kingsoft.com

Abstract

In most of the testing dataset, the images are collected from video clips or different environment conditions, which implies that the mutual information between pairs are significantly important. To address this problem and utilize this information, in this paper, we propose a graph-based unsupervised feature aggregation method for face recognition. Our method uses the inter-connection between pairs with a directed graph approach thus refine the pair-wise scores. First, based on the assumption that all features follow Gaussian distribution, we derive a iterative updating formula of features. Second, in discrete conditions, we build a directed graph where the affinity matrix is obtained from pair-wise similarities, and filtered by a pre-defined threshold along with K-nearest neighbor. Third, the affinity matrix is used to obtain a pseudo center matrix for the iterative update process. Besides evaluation on face recognition testing dataset, our proposed method can further be applied to semi-supervised learning to handle the unlabelled data for improving the performance of the deep models. We verified the effectiveness on 5 different datasets: IJB-C, CFP, YTF, TrillionPair and IQiYi Video dataset.

1. Introduction

In recent years, the performance of face recognition has been boosted by a large margin because of the success of deep learning. With the development of algorithms and increasing need for face recognition, larger and harder datasets are proposed in recent years. LFW [18] with small number of images and IDs was quickly saturated and larger dataset such as YTF [43], IJB-C [29] and MS-Celeb-1M [14] was then proposed in this research area. The face images are always captured from two types of sources: single photo or video sequence. However, most of the algorithms consider the recognition accuracy on individual images, where lacks of the relationship between testing images. In an offline evaluation testing environment, utilizing the mutual information is important.

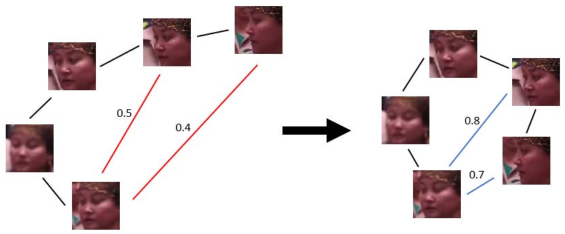

Figure 1. In a video sequence, one person's face pose may change a lot and cause wrong recognition for some image pairs (red line). However, the affinity information can be utilized to correct the recognition as the blue line.

As shown in Fig. 1, in a video captured in the wild environment, the change of face poses could be significant. In conventional one-to-one evaluation way, the comparison between large poses could result in low scores and fail in recognition. Improving the capability of model itself is usually insufficient. In this paper, we target to solve the problem using external process among the given testing samples.

Many recent papers about loss functions [11, 25, 38, 26, 41, 50] propose the idea of reducing intra-class distance, and similar to this, our proposed method aims to solve this problem by clustering. In face recognition testing protocols, the number of identities is usually unknown. Many conventional clustering method would fail due to lack of this information. Moreover, the conventional clustering approaches will take large amount of computational consumption such as k-means [27] and spectral clustering [16]. Therefore, we propose a clustering method using iterative update.

First, we assume that the feature distribution follows Gaussian distribution. For one given point in the domain, the expectation value in a neighborhood will point to the center of cluster. With multiple iterations, the testing features will converge to their corresponding centers. In real applications of face recognition, the samples are discrete, so we approximate it using directed graph. We firstly compute the affinity graph and replace the neighborhood constraint with K-nearest neighbor and certain threshold.

After clustering, we can further make use of unlabelled data to fine-tune our existing model. With our clustering results, the clusters are assigned with pseudo labels which will be added in training set. The performance could be boosted with these extra data. Our contributions can be summarised as follows:

- We introduce a post-processing module for refining the features using an iterative method.
- We propose a method for enhancing the offline evaluation in large-scale face recognition dataset.
- The proposed method can be utilized for semi-supervised learning and reach better performance using extra unlabelled data.

2. Related Works

Feature Aggregation The most intuitive way to aggregate all features in each face images set is to take the average or maximum of all features. [3, 6, 32] employ average/max pooling of the features to yield the aggregated representation. By fusing multiple models, [7] achieves better results than single NetVLAD [1]. [45] introduces attention mechanism in an aggregation module to learns compact representation that is invariant to the input frame order. Inspired by the image retrieval literature, [52] proposes a GhostVLAD layer to automatically learn to weight face descriptors. That is, down-weight the contribution of low-quality images and improve the importance of the high-quality ones. The GhostVLAD layer can be embedded into deep networks directly for an end-to-end training. Their methods surpass the state-of-the-art on the challenging IJB-A [20] and IJB-B [42] benchmarks by a large margin. [51] proposes a novel Multi-Prototype Network (MPNet) model to automatically learn multiple prototype face representations from raw video frames. Compared with existing set-based face recognition methods [4, 5, 8], MPNet can address the large intra-set variance issue with lower computational complexity. To both increase recognition accuracy and reduce the computational costs of template matching, [15] and [33] adopt the instance-level based aggregation methods which aggregate raw video frames directly instead of the features obtained by deep neural networks. Unlike all the previous methods, [13] proposes a component-wise feature aggregation network to adaptively aggregates all feature vectors into a single vector for video face recognition. The proposed component-wise feature aggregation network (C-FAN) can adaptively predicts quality values for each component of a learned feature vector, which produces a discriminative feature for images in a set.

Face Clustering often yields significant performance gain for large scale unlabeled data. Many clustering methods have been fully explored in the past few years. [9, 16] adopt spectral clustering group unlabeled faces. [31] proposes an approximate rank-order (ARO) metric to predict whether a node should be linked to its neighbors. Benefiting from the Approximate Nearest Neighbor (ANN) search algorithm, ARO is much more efficient for large-scale clustering tasks. Based on a linear SVM, [24] proposes the proximity-aware hierarchical clustering to exploit the pairwise similarity between samples. [35] designs a conditional pairwise clustering (ConPaC) method to estimate the adjacency matrix which can be used to select the number of clusters dynamically. [23, 53] propose a cluster-level affinity to deal with density-unbalanced data and tag face dataset respectively. [46, 40] improve learnable clustering method based on the graph convolution network (GCN). In particular, [46] uses the GCN to learn how to cluster rather than design new similarity metrics as most pervious works did. The proposed method achieves state-of-the-art performance in large-scale face clustering benchmarks. [40] regard clustering as a link prediction problem and utilize the GCN to predict linkages between pairs in a sub-graph. The proposed clustering method can also be easily extended to video face clustering tasks which demonstrates a good generalizability.

Graph convolution networks (GCNs) are proposed to tackle problems with non-Euclidean data. Compared with Convolutional Neural Networks (CNNs), GCNs have a strong capability of modeling graph-structured data. GCNs can be used in individual relations inference in social networks [21], recommendation engines [30, 48] and language processing [2, 28]. In addition, GCNs can also be applied to the area of computer vision tasks. [22, 44] use GCNs to predict semantic relations between object pairs in images. [39, 19, 47] apply GCNs to process unordered point cloud data for semantic segmentation. [36, 44] employ GCNs for skeleton-based action recognition. [46, 40] use GCNs for face clustering and both consequently boost the face recognition performance.

3. Method

Instead of tuning the network structures training parameters, we propose a method to improve the feature discriminativity by clustering. Inspired by recent works on loss

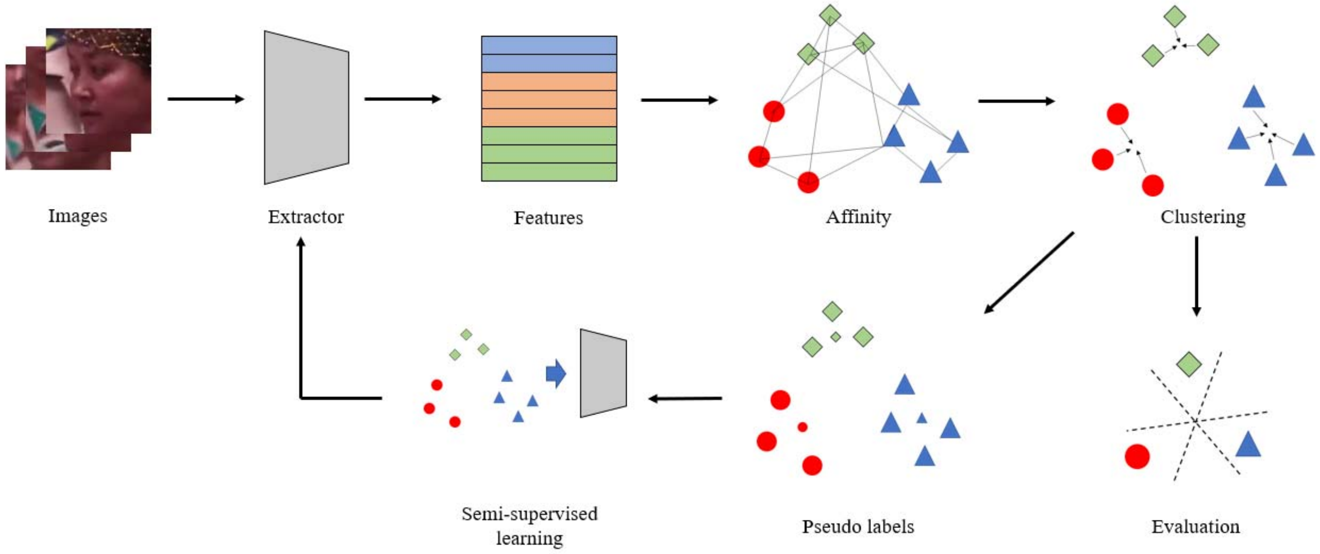

Figure 2. The pipeline of our structure. The images are firstly fed into feature extractors to obtain face features. The affinity matrix is then constructed and perform clustering. The results can be used for evaluation or further assigned with pseudo labels for fine-tuning the existing network.

functions which aims to compact the intra-class distance of features, we propose an unsupervised feature propagation method to compact the features after extraction phase. The pipeline is shown in Fig. 2. The images are firstly fed into extractor network and features are obtained. Next, the affinity matrix is derived to perform clustering. The clustered results can be used for evaluation or assigned with pseudo labels for fine-tuning the model.

3.1. Iterative Clustering

One of the classical unsupervised learning methods is clustering, such as K-means and spectral clustering. After clustering, the features can be represented by corresponding feature centers and with proper settings of hyperparameters, the features in each cluster belongs to the same identity. However, in face recognition tasks, the number of features are usually a big quantity, which will consume extreme large memory and CPU time using traditional clustering methods.

To this end, we propose an iterative clustering method as Eq. 1

$$F_i^t = F_i^{t-1} + \gamma(C_i - F_i^{t-1}) \tag{1}$$

, where F is the feature matrix and C is the cluster center matrix. A learning rate γ is applied for iterative updating.

Ideally, each row of center matrix C_i is the corresponding cluster center of feature F_i, and with the equation above, each feature will eventually converge to its center. However, in practice, the accurate center features cannot be figured out due to the considerable computational complexity; therefore, we use the pseudo center matrix C^* which will change from iteration to iteration, and result in Eq. 2.

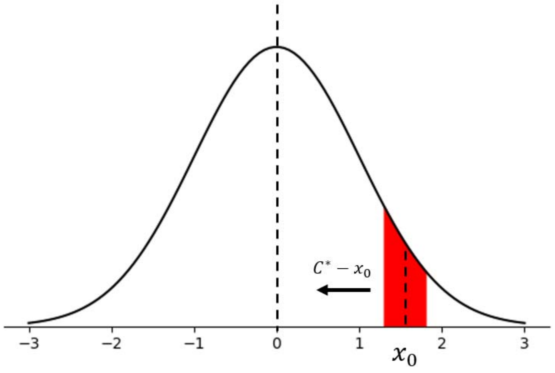

Figure 3. Considering a Gaussian distribution, the vector from point x_0 to the expectation value among the neighborhood of point x_0 is always pointed to the center of distribution.

$$F_i^t = F_i^{t-1} + \gamma(C_i^{*t-1} - F_i^{t-1}) \tag{2}$$

3.2. Pseudo Center Matrix Approximation

The pseudo center matrix enables the possibility to significantly reduce the computational complexity. We assume that for each class a, the features are normally distributed

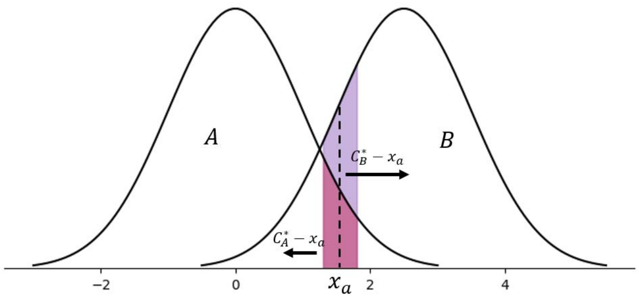

Figure 4. In the case with two distributions, wrong clustering could happen in the overlapping area. For a point x_a belongs to class a, when the probability density of class b is larger than that of class a, the convergence will move to wrong direction.

in the space with the mean value μ_a and standard deviation σ_a.

In the single cluster case, where all samples belong to the same class center, the equation should satisfy:

$$f_j - \int_{S_j} xN(x|\mu_a, \sigma_a)dx = f_j - C_j^* = \lambda(f_j - \mu_a) \quad (3)$$

, where S_j is the neighborhood area of the sample f_j and λ is a positive real number. Thus, our iterative clustering will finally converge to the cluster center.

However, in multi-class case, the convergence is not correct for every sample. Theoretically, for the sample x_a whose ground truth is a but when

$$N(x_a|\mu_a, \sigma_a) < N(x_a|\mu_b, \sigma_b) \quad (4)$$

, the sample will be assigned to wrong cluster, as illustrated in Fig. 4.

3.3. Directed Graph for Discrete Distribution

In the real-world applications, the features are discretely distributed and form an affinity graph. With a given feature matrix F of shape $n \times d$, the affinity matrix A is given by

$$A = FF^T \quad (5)$$

In discrete condition, we approximate the pseudo center matrix in Eq. 3 by the following equation

$$C_{i,j}^* = [A_{i,j} > t][j \in KNN_i] \quad (6)$$

, where the $[\cdot]$ stands for the Iverson Bracket. t is the threshold value that constraint the area in the similarity perspective. Moreover, a KNN is applied to constraint the neighborhood area and reduce the computational complexity.

Compared to conventional k-means clustering algorithm which will take $O(nkT)$ for n samples, K clusters and T iterations on average, our method yeilds the complexity of $O(nKT)$ where K nearest neighbors are selected in KNN process. For a large-scale dataset with a setting of $K = 15$ for KNN in clustering, our method is 100 times faster than conventional k-means approach with $k = 1500$. In addition, our proposed method does not require the pre-defined number of clusters.

3.4. Semi-supervised Learning with Clusters

Aside from boosting the evaluation performance using clustering, our approach also enables the possibility for semi-supervised learning. Our proposed method can be utilized for semi-supervised learning as well. For large-scale unlabelled data, we first apply our clustering and obtain an similarity matrix S. Then, the pairs with similarity score less than threshold t_s is pruned thus result in multiple clusters. The clustered samples are assigned with new labels and further added into training set for fine-tuning the network.

4. Experiment Results

4.1. Evaluations on IJB-C Co-variant Protocol

IJB-C dataset [29] is a large-scale public benchmark consisting of 31,334 images and 11,779 videos from 3,531 subjects, which are further split into 117,542 frames. The videos and photos are captured from the in-the-wild environment and containing complex pose resolution changes. The evaluation is carried out based on IJB-C 1:1 Co-variant Protocol (test 2). Moreover, for correctly reflecting the improvement brought by our algorithm, we remove the identites which overlap with MS-Celeb-1M dataset.

We train our model provided by i-Bug Lightweight Face Recognition Challenge [1] with the structure of EfficientNet [37]. The model is trained with ArcFace [11] loss function with $s = 64$ and $m = 0.5$. We use MTCNN [49] as the face detector and for non-detected images, we use the provided bounding boxes as the face box. Then, all the detected faces are aligned by 5 facial keypoints and resized to size 112×112.

The performance is shown in Fig. 5 6 and Tab. 1, where we can observe that the performance is largely dropped with smaller k and t. However, with higher threshold or larger k value in the KNN graph, the performance will be increased. Our best performance gives improvement from 0.59 to 0.67 at the $1e-7$ criteria.

Furthermore, to eliminate the negative effect brought by mis-detection or mis-alignment which is irrelevant to the recognition, we remove the non-detected samples in IJB-C

[1] https://ibug.doc.ic.ac.uk/resources/lightweight-face-recognition-challenge-workshop/

K	Threshold	Cleaned	$1e-7$	$1e-6$	$1e-5$	$1e-4$	$1e-3$
-	-	No	0.5899	0.6691	0.7429	0.8011	0.8491
5	0.75	No	0.6374	0.6955	0.7638	0.8320	0.8797
10	0.75	No	0.6588	0.7096	0.7693	0.8341	0.8802
15	0.75	No	0.6669	0.7142	0.7711	0.8348	0.8884
20	0.75	No	0.6711	0.7165	0.7720	0.8350	0.8804
5	0.55	No	0.1679	0.3127	0.5839	0.8003	0.8911
10	0.55	No	0.2221	0.3339	0.5073	0.7913	0.8925
15	0.55	No	0.2569	0.3668	0.4989	0.7938	0.8930
20	0.55	No	0.3140	0.3786	0.5018	0.7973	0.8933
-	-	Yes	0.5899	0.6690	0.7429	0.8011	0.8490
5	0.75	Yes	0.7247	0.7741	0.8061	0.8386	0.8668
10	0.75	Yes	0.7342	0.7776	0.8089	0.8395	0.8674
15	0.75	Yes	0.7369	0.7772	0.8091	0.8397	0.8674
20	0.75	Yes	0.7382	0.7802	0.8099	0.8397	0.8675
5	0.55	Yes	0.7127	0.7804	0.8342	0.8620	0.8837
10	0.55	Yes	0.6634	0.7738	0.8293	0.8648	0.8862
15	0.55	Yes	0.6666	0.7692	0.8271	0.8658	0.8873
20	0.55	Yes	0.6851	0.7785	0.8261	0.8660	0.8878

Table 1. The evaluation results on IJB-C dataset co-variant protocol (test 2). We removed the identities which is overlapped with MS-Celeb-1M dataset for better evaluation. We utilize the EfficientNet structure for feature extraction and tested different threshold and K size for clustering. Furthermore, we remove the images which is not detected to perform another evaluation on ”cleaned” dataset, which is more fair for recognition task.

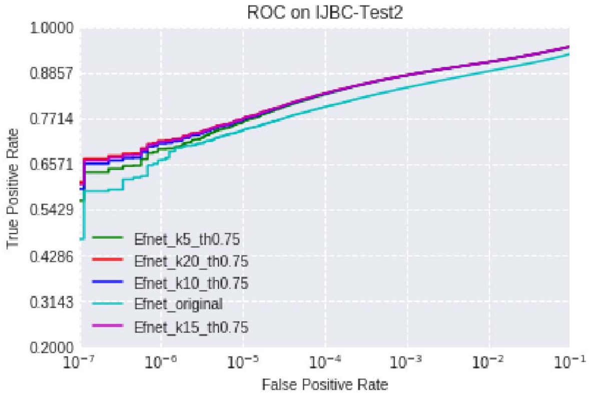

Figure 5. The evaluation on IJB-C dataset co-variant protocol with fixed threshold $t = 0.75$ and varies the K for KNN. A consistent improvement is observed compared to the baseline.

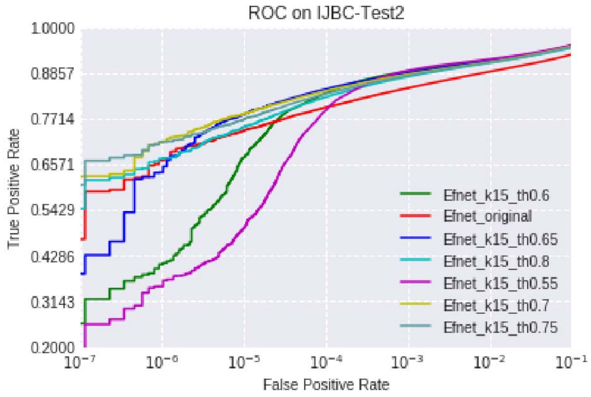

Figure 6. The evaluation result on IJB-C dataset co-variant protocol with fixed K for KNN. A huge drop in performance is observed with lower K but it is recovered with higher values.

dataset and perform another evaluation on this ”cleaned” version. In this circumstance, we can assume that the dataset is mostly clean and reliable, and the results is shown in Fig. 7 and Tab. 1. It is observed that with different hyper-parameter settings, our algorithm can give stable improvement and the result is insensitive to different hyper-parameters.

We show some representative failure cases which is shown in Fig. 8. The failure cases are usually caused by low image quality or mis-alignment. It shows the necessity of reducing noise before applying clustering, and the experiments in IJB-C dataset show the significant effectiveness of our clustering on cleaned dataset.

4.2. Evaluations on IJB-C Template Protocol

We further conduct the evaluation on IJB-C Template matching protocol (test 1) which evaluates on 15.6M template pairs. We extract all images features and aggregate them by the template ID provided. We take the naive averaging as the aggregation method for all templates. Then our

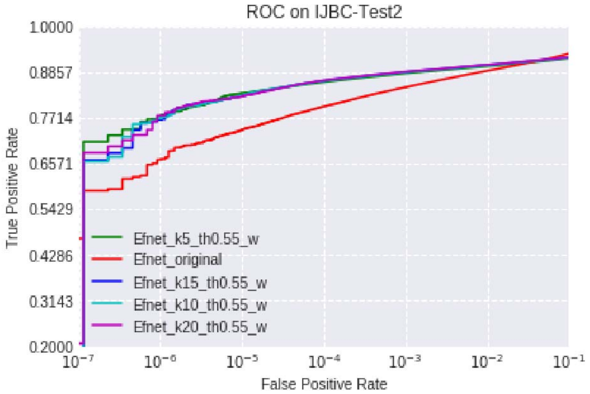

Figure 7. The evaluation result on IJB-C dataset with cleaned version, where the non-detected images are removed for better evaluation of recognition model. Even with lower threshold t, a constant improvement of performance can still be achieved.

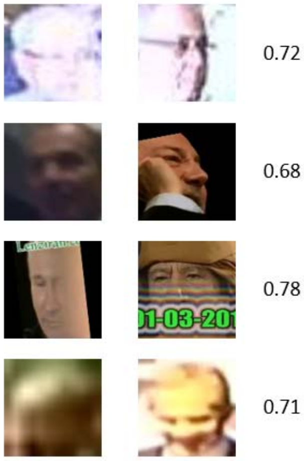

Figure 8. Some representative failure cases. The abnormal illumination, misalignment and extreme low resolution will increase the error of our clustering method.

cluster method was applied on the template features. The results are shown in Fig. 10 and Tab. 2. The results show that our method performs much better than the original score.

4.3. Evaluations on CFP Dataset

CFP [34] is a dataset including 7000 images and 500 identities. We follow the protocol which contains 7000 pairs of images and calculate the AUC curve. The MTCNN is utilized and all images are resized to 112×112.

Same as the experiments in Section 4.2, we evaluate our clustering result based on the same feature extractor network and different K and t values are tested. The results are shown in Fig. 11 and Tab. 3.

Since the CFP-FF protocol is already saturated, here we evaluate the performance on CFP-FP protocol. From Tab. 3, we can observe that the clustering method helps to improve the true positive rate (TPR) by almost 3 percent while false positive rate (FPR) equals 1e-3. However, when we increase the clustering threshold, the increase becomes not that obvious and K value have no effectiveness. One of the possible reason is that number of images and pairs is not sufficient, where our clustering becomes less effective.

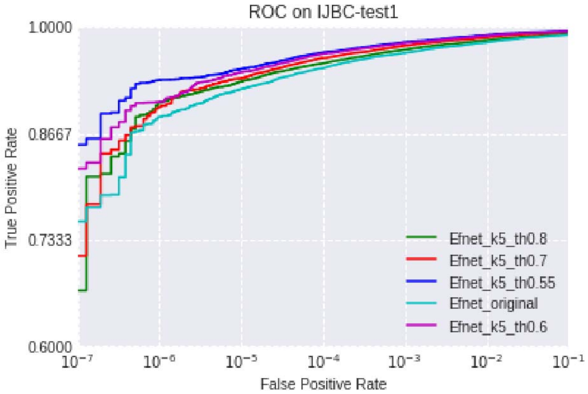

Figure 9. The evaluation results on IJB-C template matching protocol. With fixed K, variation of t gives consistent improvement.

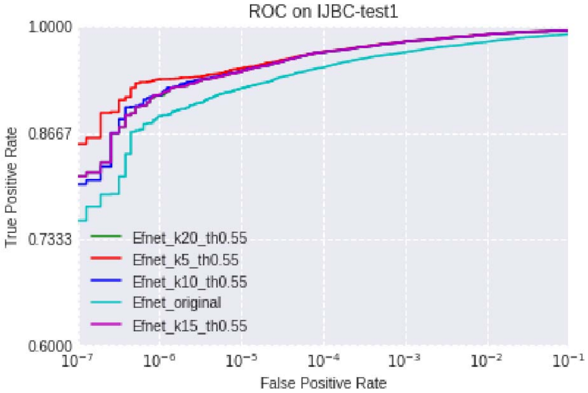

Figure 10. The evaluation results on IJB-C template matching protocol. With fixed t, variation of K gives improvement.

4.4. Evaluations on YTF Dataset

YouTube Faces DB [43] is a dataset containing 3,425 videos and 1,595 identities. The video clips are split into 621,126 frames. Provided with the 5,000 pairs, we perform our evaluation of our model as well as the effects of the cluster method.

The images are firstly processed in the same way as indicated in Section 4.2. For this dataset, each pair consists of multiple frames. We cluster all the video features after aggregating the frames features into one by sum up their features, and then perform the evaluation. The results are

K	Threshold	$1e-7$	$1e-6$	$1e-5$	$1e-4$	$1e-3$
-	-	0.7743	0.8877	0.9222	0.9485	0.9675
5	0.55	0.8604	0.9339	0.9477	0.9681	0.9812
10	0.55	0.8074	0.9141	0.9441	0.9676	0.9801
15	0.55	0.8169	0.9154	0.9433	0.9675	0.9810
20	0.55	0.8169	0.9139	0.9433	0.9675	0.9810
5	0.6	0.8302	0.9065	0.9434	0.9664	0.9801
5	0.7	0.7781	0.9011	0.9357	0.9613	0.9767
5	0.8	0.8127	0.9055	0.9317	0.9547	0.9717
10	0.6	0.7781	0.9018	0.9412	0.9655	0.9798
10	0.7	0.7229	0.8954	0.9362	0.9607	0.9766
10	0.8	0.7754	0.9054	0.9324	0.9546	0.9715

Table 2. The evaluation results on IJB-C dataset template maching protocol. We use the original dataset for evaluation and different threshold and K size is tested.

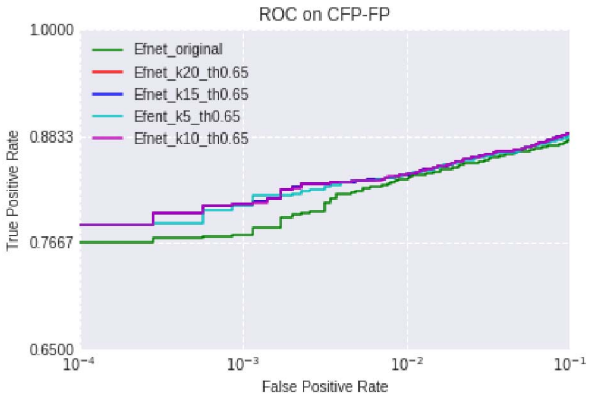

Figure 11. The evaluation results on CFP dataset. The threshold t is fixed at 0.65 and k varies from 5 to 20. The performance is improved in all cases.

k	t	$1e-4$	$1e-3$	$1e-2$	$1e-1$
-	-	0.7671	0.7820	0.8360	0.8791
5	0.65	0.7860	0.8177	0.8417	0.8846
10	0.65	0.7860	0.8109	0.8417	0.8869
15	0.65	0.7866	0.8111	0.8411	0.8863
20	0.65	0.7866	0.8111	0.8411	0.8863
5	0.60	0.7909	0.8106	0.8417	0.8877
10	0.60	0.7671	0.7820	0.8360	0.8791
15	0.60	0.8029	0.8146	0.8423	0.8889
20	0.60	0.8029	0.8146	0.8423	0.8889
5	0.70	0.7740	0.8014	0.8400	0.8846
10	0.70	0.7706	0.8057	0.8411	0.8851
15	0.70	0.7706	0.8057	0.8411	0.8851
20	0.70	0.7706	0.8057	0.8411	0.8851

Table 3. The evaluation results on CFP-FP dataset.

shown in Fig. 12 and Tab. 4.

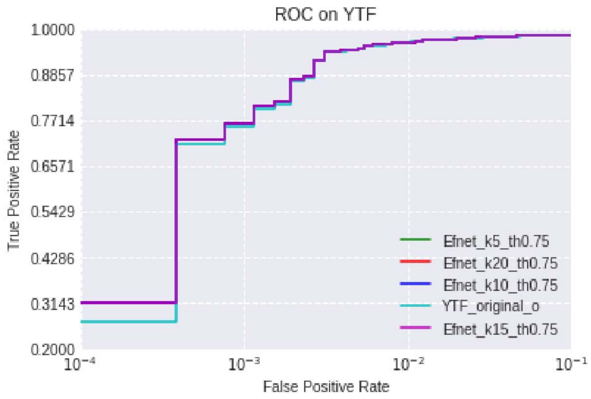

Figure 12. The ROC curve on YTF dataset. The performance is increased by the clustering.

The YTF result shows that our cluster method increase the performance significantly when FPR equals 1e-4, around 25 percent improvement while threshold equals 0.55. Similar to the CFP dataset, YFP is not sensitive to K values.

4.5. Evaluations on TrillionPair and IQiYi Dataset

Furthermore, we perform the evaluation on TrillionPair and IQiYi dataset. We follow the evaluation protocol of iBug Light-Weight Competition[12][2] where the faces are detected by RetinaFace [10] detector and resized to 112 × 112. The TrillionPair dataset contains 1.862 million images where there are 274K ground truth images from 5.7K identities and 1.58M distractors. The IQiYi dataset contains 6.3M face frames with 203,848 groups.

We trained one EfficientNet B0 [37] (noted as Light) and one ResNet-SE-100 [17] (noted as heavy). For both networks, we train them based on ArcFace loss function with $s = 64$ and $m = 0.5$. SGD optimizer is used with momen-

[2] http://39.104.128.76/overview

k	t	$1e-4$	$1e-3$	$1e-2$	$1e-1$
-	-	0.2670	0.8030	0.9683	0.9854
5	0.65	0.3469	0.7151	0.9679	0.9854
10	0.65	0.3482	0.7151	0.9679	0.9854
15	0.65	0.3482	0.7151	0.9679	0.9854
20	0.65	0.3482	0.7151	0.9679	0.9854
5	0.55	0.4981	0.7447	0.9708	0.9863
10	0.55	0.5056	0.7447	0.9708	0.9863
15	0.55	0.5056	0.7447	0.9708	0.9863
20	0.55	0.5056	0.7447	0.9708	0.9863
5	0.75	0.3161	0.8072	0.9688	0.9850
10	0.75	0.3161	0.8072	0.9688	0.9850
15	0.75	0.3161	0.8072	0.9688	0.9850
20	0.75	0.3161	0.8072	0.9688	0.9850

Table 4. The evaluation results on YTF dataset. The clustering obtains increase in performance for strict conditions.

Model	Protocol	Base	Clustered
Light	Trillion	0.8200	0.9341
Light	IQiYi	0.5702	0.7222
Heavy	Trillion	0.9200	0.9793
Heavy	IQiYi	0.6489	0.7259

Table 5. The evaluation is based on $1e-8$ for TrillionPair and $1e-4$ for IQiYi dataset.

tum 0.9 and learning rate 0.1 which is scaled by 0.1 every 4 iterations. For both datasets, we extract features from all the images and perform our clustering algorithm. In addition, for the IQiYi dataset, since it is a template-based protocol where images are split into different groups, we first conduct intra-group feature aggregation according to the features $L-norm$ where L is chosen to be 3.6 empirically.

The evaluation results are shown in Tab. 5. For hyper-parameter settings we choose $K=15$ and $t=0.55$ for both datasets since all the images are successfully detected and relatively reliable, refering to Section 4.2. Our clustering can significantly boost the performance of models where the light model is boosted by 20% and the heavy one by 7.5% on TrillionPair dataset. The same effect is also observed in IQiYi dataset where the light model is boosted by 15% and 8%, respectively.

4.6. Semi-supervised Learning

We also evaluated the semi-supervised learning with our proposed clustering method. We utilized the same training dataset as in previous sections but we remain only a proportion of identities labels for training. After training the initial model with labelled data, our clustering is applied to the remaining unlabelled data and fine-tune the network.

The evaluation is conducted on IJB-C dataset and the results are shown in Fig. 13. The lower-bounder is evaluated by the model with fully-supervised learning with limited labelled data, and upper-bound is the performance of model trained with the whole dataset. We observe that our clustering can effectively boost the performance of our model with unlabelled images. With only 50% of the dataset, the $TPR@FPR=1E-5$ is increased by 13% with our method and performance with other proportion of data is also boosted by a large margin.

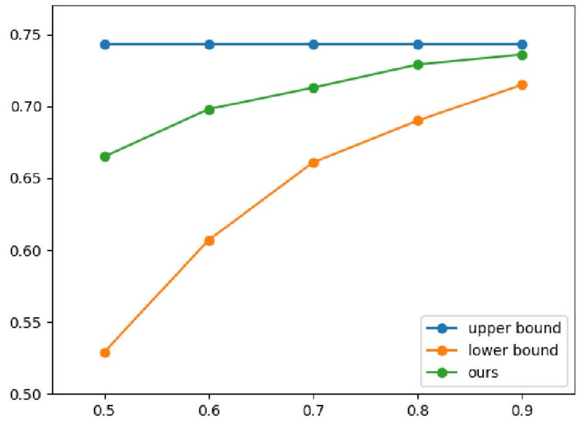

Figure 13. The evaluation of semi-supervised learning. The upper bound is the model trained with full data, and lower bound is the model with partial training data. The results are evaluated on IJB-C Co-variant protocol (test 2) uncleaned version as in Section 4.1. Results at $FPR=1e-5$ are shown in this figure.

5. Discussion and Future Work

From the evaluations on different in-the-wild datasets, we observe that our algorithm can give improvements based on extracted features. Instead of merely improving the network's performance, our method can boost the performance of recognition by post-processing. However, there still exist some limitations of our proposed method:

- The algorithm is somehow dependent on hyper-parameter setting, especially for noisy dataset, although it can be fixed by increasing the K and threshold value.

- The algorithm is not an online algorithm, which requires the image pool to be fixed.

- Extra computational consumption. The clustering process will cost extra 30 minutes for $1.5M$ images with feature dimension of 512 on single 2080Ti GPU according to our evaluation.

Although with the limitations above, our proposed method is still useful in various of offline applications. Furthermore, the soft-clustering method will be helpful to various types of tasks not limited to face recognition but can be extended to other applications such as person ReID and image search.

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. *IEEE Transactions on Pattern Analysis Machine Intelligence*, PP(99):1–1, 2017. 2

[2] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, and K. Sima'an. Graph convolutional encoders for syntax-aware neural machine translation. *CoRR*, abs/1704.04675, 2017. 2

[3] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. *CoRR*, abs/1710.08092, 2017. 2

[4] R. Chellappa, J. C. Chen, R. Ranjan, S. Sankaranarayanan, and C. D. Castillo. Towards the design of an end-to-end automated system for image and video-based recognition. 2017. 2

[5] J. C. Chen, V. M. Patel, and R. Chellappa. Unconstrained face verification using deep cnn features. 2015. 2

[6] J. C. Chen, R. Ranjan, A. Kumar, C. H. Chen, V. M. Patel, and R. Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. 2015. 2

[7] S. Chen, W. Xi, Y. Tang, X. Chen, and Y. G. Jiang. Aggregating frame-level features for large-scale video classification. In *CVPR Workshop*, 2017. 2

[8] A. R. Chowdhury, T. Y. Lin, S. Maji, and E. Learned-Miller. One-to-many face recognition with bilinear cnns. 2015. 2

[9] J. Cui, W. Fang, X. Rong, Y. Tian, and X. Tang. Easyalbum:an interactive photo annotation system based on face clustering and re-ranking. In *Conference on Human Factors in Computing Systems*, 2007. 2

[10] J. Deng, J. Guo, Z. Yuxiang, J. Yu, I. Kotsia, and S. Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. In *arxiv*, 2019. 7

[11] J. Deng, J. Guo, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *CVPR*, 2019. 1, 4

[12] J. Deng, J. Guo, D. Zhang, Y. Deng, X. Lu, S. Shi, and S. Zafeiriou. Lightweight face recognition challenge. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 7

[13] S. Gong, Y. Shi, and A. K. Jain. Video face recognition: Component-wise feature aggregation network (c-fan). 2019. 2

[14] Y. Guo, Z. Lei, Y. Hu, X. He, and J. Gao. *MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition*. 2016. 1

[15] T. Hassner, I. Masi, J. Kim, J. Choi, S. Harel, P. Natarajan, and G. Medioni. Pooling faces: Template based face recognition with pooled face images. 2016. 2

[16] J. Ho, M. H. Yang, J. Lim, and K. C. Lee. Clustering appearances of objects under varying illumination conditions. In *CVPR*, 2006. 2

[17] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 7

[18] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. 2007. 1

[19] Q. Huang, W. Wang, and U. Neumann. Recurrent slice networks for 3d segmentation on point clouds. 2018. 2

[20] B. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. 2015. 2

[21] T. Lei and H. Liu. Relational learning via latent social dimensions. In *Acm Sigkdd International Conference on Knowledge Discovery Data Mining*, 2009. 2

[22] Y. Li, W. Ouyang, B. Zhou, Y. Cui, J. Shi, and X. Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. 2018. 2

[23] W.-A. Lin, J.-C. Chen, C. D. Castillo, and R. Chellappa. Deep density clustering of unconstrained faces. 2018. 2

[24] W. A. Lin, J. C. Chen, and R. Chellappa. A proximity-aware hierarchical clustering of faces. 2017. 2

[25] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. *CVPR*, 2017. 1

[26] W. Liu, Y. Wen, Z. Yu, and Y. Meng. Large-margin softmax loss for convolutional neural networks. *ICML*, 2016. 1

[27] S. P. LLOYD. Least squares quantization in pcm. *IEEE Trans*, 28(2):129–137, 1982. 2

[28] D. Marcheggiani and I. Titov. Encoding sentences with graph convolutional networks for semantic role labeling. *CoRR*, abs/1703.04826, 2017. 2

[29] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, and P. Grother. Iarpa janus benchmark - c: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, Feb 2018. 1, 4

[30] F. Monti, M. M. Bronstein, and X. Bresson. Geometric matrix completion with recurrent multi-graph neural networks. 2017. 2

[31] C. Otto, D. Wang, and A. K. Jain. Clustering millions of faces by identity. *CoRR*, abs/1604.00989, 2016. 2

[32] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, 2015. 2

[33] Y. Rao, J. Lu, and J. Zhou. Learning discriminative aggregation network for video-based face recognition and person re-identification. *IJCV*, (2), 2018. 2

[34] S. Sengupta, J. C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016. 6

[35] Y. Shi, C. Otto, and A. K. Jain. Face clustering: Representation and pairwise constraints. *IEEE Transactions on Information Forensics Security*, PP(99):1–1, 2017. 2

[36] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. June 2019. 2

[37] M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 4, 7

[38] H. Wang, Y. Wang, Z. Zhou, X. Ji, Z. Li, D. Gong, J. Zhou, and W. Liu. Cosface: Large margin cosine loss for deep face recognition. *CVPR*, 2018. 1

[39] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan. Graph attention convolution for point cloud semantic segmentation. In *CVPR*, June 2019. 2

[40] Z. Wang, L. Zheng, Y. Li, and S. Wang. Linkage based face clustering via graph convolution network. 2019. 2

[41] Y. Wen, K. Zhang, Z. Li, and Q. Yu. A discriminative feature learning approach for deep face recognition. 2016. 1

[42] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, and P. Grother. Iarpa janus benchmark-b face dataset. 2017. 2

[43] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. 2011. 1, 6

[44] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. 2018. 2

[45] J. Yang, P. Ren, D. Chen, F. Wen, H. Li, and G. Hua. Neural aggregation network for video face recognition. *CoRR*, abs/1603.05474, 2016. 2

[46] L. Yang, X. Zhan, D. Chen, J. Yan, C. C. Loy, and D. Lin. Learning to cluster faces on an affinity graph. 2019. 2

[47] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang. 3d recurrent neural networks with context fusion for point cloud semantic segmentation. 2018. 2

[48] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec. Graph convolutional neural networks for web-scale recommender systems. *CoRR*, abs/1806.01973, 2018. 2

[49] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016. 4

[50] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. 2019. 1

[51] J. Zhao, J. Li, X. Tu, F. Zhao, and J. Feng. Multi-prototype networks for unconstrained set-based face recognition. 2019. 2

[52] Y. Zhong, R. Arandjelovic, and A. Zisserman. Ghostvlad for set-based face recognition. *CoRR*, abs/1810.09951, 2018. 2

[53] C. Zhu, W. Fang, and S. Jian. A rank-order distance based clustering algorithm for face tagging. 2011. 2