Robust Face Recognition with Deep Multi-View Representation Learning

Jianshu Li [1,3]
jianshu@u.nus.edu

Jian Zhao[2]
zhaojian90@u.nus.edu

Fang Zhao[2]
elezhf@nus.edu.sg

Hao Liu [4,2]
hfut.haoliu@gmail.com

Jing Li[1]
etta090807@gmail.com

Jiashi Feng[2]
elefjia@nus.edu.sg

Terence Sim[1]
tsim@comp.nus.edu.sg

[1] School of Computing, National University of Singapore, Singapore
[2] Department of Electrical and Computer Engineering, National University of Singapore, Singapore
[3] SAP Research and Innovation Singapore, SAP Asia Pte Ltd
[4] School of Computer and Information, Hefei University of Technology, Hefei, Anhui China

ABSTRACT

This paper describes our proposed method targeting at the MSR Image Recognition Challenge MS-Celeb-1M. The challenge is to recognize *one million* celebrities from their face images captured in the real world. The challenge provides a large scale dataset crawled from the Web, which contains a large number of celebrities with many images for each subject. Given a new testing image, the challenge requires an identify for the image and the corresponding confidence score. To complete the challenge, we propose a two-stage approach consisting of data cleaning and multi-view deep representation learning. The data cleaning can effectively reduce the noise level of training data and thus improves the performance of deep learning based face recognition models. The multi-view representation learning enables the learned face representations to be more specific and discriminative. Thus the difficulties of recognizing faces out of a huge number of subjects are substantially relieved. Our proposed method achieves a coverage of 46.1% at 95% precision on the random set and a coverage of 33.0% at 95% precision on the hard set of this challenge.

Keywords

Face Recognition; Deep Learning; Multi-view Feature Representation; Model Ensemble

1. INTRODUCTION AND OVERVIEW

Recently many deep learning based face recognition techniques have been developed and achieved remarkable performance in various application scenarios, such as DeepFace [9], FaceNet [5] and DeepID [7]. However, most of the existing face recognition techniques do not consider how to disambiguate the identity of a person. Another issue significantly hindering the development of face recognition techniques is that publicly available face datasets are usually far smaller than being sufficient to build real face recognition systems. To address these issues, the MSR Image Recognition Challenge introduces a knowledge base where each face is linked to a unique entity key and provides a large scale face dataset (MsCeleb), which contains about $100,000$ celebrities with about 100 images for each subject.

To address the face identification task proposed in the challenge, we propose a two-stage method to learn robust and disambiguated human face representations for effectively recognizing human faces at large scale. The first stage is to clean the noisy data in the provided training set. Since the provided training dataset is crawled from the Internet without manually checking, there are a certain percentage of data which are actually noisy. For example, some of the faces are given wrong labels and some images even do not contain faces. To clean the noise, we first train a deep neural network on an existing large scale face dataset using a classification loss. We use the model pre-trained on the clean dataset to extract features for face images from MsCeleb. The activations from the penultimate layer are adopted as the feature representations of images from the MsCeleb dataset. Then for each subject, we apply an outlier detection method to remove the noisy images for the subject. Built on the cleaned dataset, the second stage of our proposed method is multi-view deep representation learning. To capture the intrinsic diversity of MsCeleb and enhance the generalization performance of face recognition models, we use multiple deep models with various architectures and loss functions to learn the representations of faces. These models provide distinct features to better characterize the data distribution jointly from different "views". Thus, representing a face from different views could effectively decrease the subject ambiguity and enhance recognition performance for an extremely large number of subjects. Then a separate classifier, whose input is the combined features from multiple deep models, is trained to perform multi-view feature fusion and classifica-

MM '16, October 15-19, 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2964284.2984061

tion. The overall structure is shown in Figure 1. The two-stage method is proved to be effective at learning robust representation from the noisy training data in the challenge.

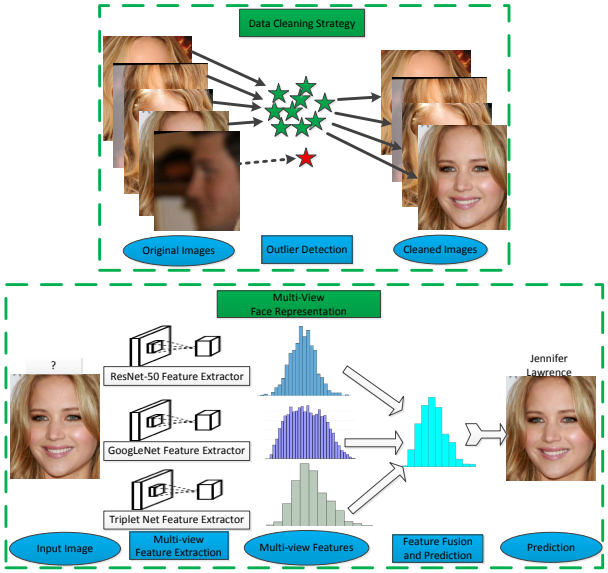

Figure 1: Overall structure of the proposed solution to the challenge: Data Cleaning Strategy and Multi-view Face Representation.

2. THE PROPOSED METHOD

2.1 Noisy Data Cleaning

Real world data are usually noisy, especially those crawled from the Internet. Automatic data cleaning is thus necessary for training machine learning models to better extract useful knowledge from the data. Bearing this in mind, in this face recognition challenge, we observe that the provided dataset is obtained by leveraging the public search engines. Therefore, a certain proportion of noisy data is inevitable and will hurt the performance of face recognition models trained on this dataset. It is clear that manual removal of the noise in the training data is almost impossible with affordable time cost.

Therefore, in our proposed method, automatic noisy cleaning is considered to be very important. We solve the problem of noisy data cleaning through outlier detection. Here, an outlier face image may be the one with an incorrect subject label or the one containing no faces. Firstly we transform the raw images to a high-level representative feature space, where outlier detection is more reliable and easier. In particular, we propose to adopt the activations from certain layers in a pre-trained Convolutional Neural Network (CNN) as the new representations. Then distances between the feature vectors of images for the same subject are calculated. Based on the distances, the feature vectors are clustered and outliers with abnormal distance with others can be detected.

One way to model the distance used in the outlier detection is to use Gaussian Mixture Model (GMM). Theoretically with GMM, we are able to get multiple cluster centers to model the intra-class variations, such as age, pose, illumination *etc.* Then the probability of each feature vector on the trained GMM model can be used as the distance metric to perform outlier detection. Some robustified variants of GMM [4] can also be used to eliminate the effect of outliers when learning the GMM model. Due to highly elevated time cost of the GMM training procedure, we simply use one cluster in the challenge as a practical solution, which is a special case of GMM with one Gaussian center. With only one cluster, we can easily find the cluster center of all feature vectors corresponding to one subject. The cluster center is treated as a reference point, and Euclidean distance of each feature vector of the subject to the center is calculated. Based on the distances, the outlier detection algorithm based on Thompson Tau method [1] is used to remove a fix proportion of the feature vectors as outliers. The above procedure is applied independently to the images of all the subjects in the training set.

2.2 Multi-View Deep Representation Learning

How to learn feature representations of the noisy data that are discriminative for a huge number of subjects (the number is 100,000 in this challenge) is a really challenging task, as the feature dimensions may be far smaller than the number of subjects and are ambiguous in representing the subjects. For the toy example in Figure 2, five classes lie in a two-dimensional space. It is hard to discriminate all the classes from a single direction of view. When combining information from multiple views, *e.g.*, from both x and y axes, it is easier to decide the decision boundaries of all the classes.

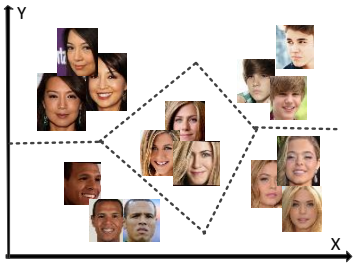

Figure 2: Illustration of benefits of multi-view classification.

We here leverage deep convolutional neural networks, which have been successfully applied to many domains, to learn deep face representations. In particular, we focus on learning the representations from different views in order to enhance the discriminative ability of the learned representations. In order to better characterize the face data distribution from different "views", we employ deep networks with various structures and loss functions to extract data representation from different perspectives. We now proceed to explain the details.

Firstly, two different network structures, Residual Net [2] and GoogLeNet [8] are used. Residual Net has the advantage of training substantially deep networks easily. We utilize the power of residual nets to build deep models and apply them to the large scale face recognition task. With the help of the residual structure, we train a deep model with 50 layers for the face recognition challenge. GoogLeNet is an efficient and successful deep model for computer vision related tasks. Through careful crafted design, GoogLeNet increases the depth and width of the network while keeping the computational budget constant with the Inception modules. Here we adopt it for this face recognition challenge.

Furthermore, two different loss functions, *i.e.*, cross entropy classification loss and triplet loss [5], are used. Cross

entropy classification loss directly distinguishes all the classes and focuses on learning the global feature which can give the best decision boundary to differentiate all the classes. The cross entropy loss is defined as

$$\mathcal{L}_{ce} = -\frac{1}{N}\sum_{i}^{N}\left[p_i \log \hat{p}_i + (1-p_i)\log(1-\hat{p}_i)\right],$$

where N is the number of all the images in the training set, p_i is the ground truth label for the i-th image and $\hat{p}_i$ is the corresponding predicted probability.

Triplet loss function aims to generate discriminative feature representation by properly increasing the similarity and difference of positive and negative pairs in each triplet. A triplet of sample images refers to a tuple of three sample images denoted as I, I^+ and I^-, which correspond to the anchor sample, the positive sample and the negative sample respectively. Here I and I^+ come from the same class (positive pair), while I and I^- are from different classes (negative pair). When extracting features from the image triplet, the truly matched images are supposed to be closer than the mismatched images. More specifically, we denote $\mathbf{H}$ as the extracted feature from image I. Then within a triplet of $\langle \mathbf{H}, \mathbf{H}^+, \mathbf{H}^- \rangle$, the feature of the positive sample $\mathbf{H}^+$ should be more similar to the reference sample $\mathbf{H}$ than the feature of the negative sample $\mathbf{H}^-$:

$$\left\|\mathbf{H}-\mathbf{H}^+\right\|_2^2 + \alpha < \left\|\mathbf{H}-\mathbf{H}^-\right\|_2^2.$$

Here α is an additionally introduced margin that is introduced to enhance the discriminative ability of learned features between positive and negative pairs. Therefore, for N triplets, the loss function to minimize is:

$$\mathcal{L}_{tri} = \frac{1}{N}\sum_{i}^{N}\left[\left\|\mathbf{H}_i-\mathbf{H}_i^+\right\|_2^2 - \left\|\mathbf{H}_i-\mathbf{H}_i^-\right\|_2^2 + \alpha\right]_+,$$

where $[\cdot]_+$ truncates the involved variable at zero.

To fuse the multi-view feature representations, we project the features from every model into a common D-dimensional feature space via function $f_i : \mathbb{R}^{d_i} \rightarrow \mathbb{R}^D$ with parameter w_i. A classifier f_c parametrized by w_c takes as input the fused feature in the common feature space and predicts the target label l. The loss function for the fusion process is written as

$$\mathcal{L}_{fusion} = \mathcal{L}_{ce}\left(f_c(\sum_i f_i(x_i)), l\right) + \sum_i \lambda_i R(w_i) + \lambda_c R(w_c),$$

where index i runs over all the base models, R is the regularizer on parameters and λs are the corresponding weights of the regularizers. By minimizing the loss function, the parameters for feature fusion w_i and classification w_c can be learnt jointly.

To summarize, we train the base deep models with three different network architectures for the face recognition challenge. The first one is a ResNet-50 net with the cross entropy loss, the second one is GoogLeNet with the cross entropy loss, and the third one is ResNet-18 with the triplet loss. After training the base models, we use them to extract feature representations of face images. For each image, we pass it through all the base networks and use the activations from penultimate layers of the base networks as high-level representations of the image. Finally we use a two-layer MultiLayer Perceptron (MLP) for feature fusion and classification.

3. IMPLEMENTATION DETAILS

3.1 Data Cleaning

In order to get the high-level representation of the face images facilitating data cleaning, we use CASIA-WebFace [10] to train a ResNet-50 model with classification loss. The activations from its penultimate layer are used as the feature representations of the images. CASIA-WebFace is a large scale clean face dataset containing $10,575$ subjects and $494,414$ images, thus it is good for learning the intended high-level face representation. We use Residual Net with 50 layers as the deep CNN model, with the final layer being a $10,575$ way classification layer with cross entropy loss. The ResNet-50 model is trained from scratch with a batch size of 8. The initial learning rate is set as 0.1, and is then reduced to 0.01 and 0.001 along the training process. We train the model for about 100 epochs for each learning rate step.

The trained model is used to extract data representation for further cleaning. For all the aligned face images in the MsCeleb dataset, we use the pre-trained ResNet-50 to extract their feature vectors. Then for all the feature vectors corresponding to one person, we calculate the median of all these vectors as the cluster center and Euclidean distance of each feature vector to the center. Based on the distances, we remove 10% of the feature vectors as outliers.

With this method, 10% of the training data are removed, resulting in a cleaner dataset. Some of the discarded data are visualized in Figure 3. We can see that the outlier removal algorithm can discover and remove some of the noise in the dataset. The discovered noise contains non-face objects, faces with opposite genders and faces with very low resolution. There are some failure cases, where the correct face belonging to the same celebrity in that class is removed by mistake. The effect of the data cleaning method is twofold. Firstly it reduces the complexity of the training data so that the feature learning becomes easier. Secondly it also removes some of the challenging faces in the dataset, which might also affect the performance negatively. The effect of the data cleaning method is demonstrated in the experiment section.

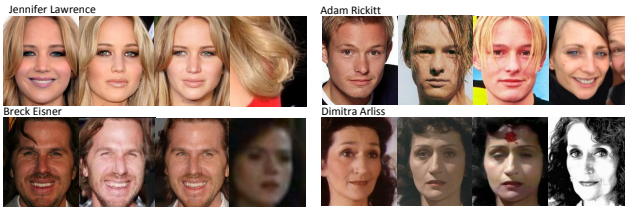

Figure 3: Correctly and wrongly detected outliers. For each celebrity, the first three images are references and the last image is the detected outlier. For Jennifer Lawrence, Adam Rickitt and Breck Eisner, the outliers are correctly identified. For Dimitra Arliss, the outlier is actually a challenging sample.

3.2 Base Model Feature Learning

3.2.1 ResNet-50

The first network for feature learning is the ResNet-50. We continue fine-tuning the network pre-trained on CASIA-WebFace with the cleaned data from MsCeleb dataset. The network structure is the same as the pre-trained model, except that the final $10,575$ way classification layer is replaced by a $76,674$ way classification layer. The initial learning

rate for the network is 0.001, except for the newly added last layer, which is 10 times larger. With a batch size of 8, the ResNet-50 model is trained at the initial learning rate until convergence. Then the training process is repeated at a learning rate of 0.0001.

3.2.2 GoogLeNet

The second network for feature learning is GoogLeNet. This model is trained on the Labelled Faces in the Wild (LFW) [3] dataset. We continue fine-tuning the network pre-trained on LFW with the cleaned data from MsCeleb dataset. The last layer of this network is modified the same way as ResNet-50, and a similar training schedule is used to fine-tune GoogLeNet.

3.2.3 TripletNet

The third network for feature learning is TripletNet, which is a ResNet-18 with triplet loss. The base Residual net with 18 layers is trained with CASIA-WebFace [10]. Then we use on-line triplet selection strategy to select triplets within each mini-batch. The on-line triplets selection approach selects triplets that are active and can contribute to improving the model within a mini-batch, so it is of higher efficiency and lower resource consumption. Specifically, instead of picking the hard positive, we adopt all positive pairs and randomly sampled negative samples added to each mini-batch. In practice, we find that using all positive pairs makes the model more stable and converge faster than selectively using hard positive pairs in a mini-batch. The TripletNet is trained with a batch size of 140 and a learning rate of 0.001 until convergence.

3.3 Feature Fusion

The dimension of the common feature space is set as $2,048$. The feature fusion and classification task is modeled by a two-layer MLP with $2,048$ hidden neurons. The MLP first reduces the dimension of the concatenated feature vector to $2,048$. Then the final layer performs a $76,674$ way classification based on the dimension reduced features. Dropout [6] is used between the two layers with a dropout rate of 0.5 to alleviate over-fitting.

4. EXPERIMENTS

The testing data are not provided, so we use the data from the dry run as the validation set. The validation set has two subsets, Dev1 and Dev2, each of which contains 500 images and ground truth labels. We test the performance of the baseline models as well as the ensemble model on the validation set.

4.1 Experiment with Data Cleaning

Our proposed data cleaning technique helps to clean the noisy data, which also filters out some challenging data. In this section, we conduct experiments to show the effectiveness of data cleaning. Two versions of ResNet-50 model are trained, one is trained on the original (and also noisy) data and the other on the data after applying the proposed data cleaning algorithm in Section 3.1. We train both models with an equal number of epochs and test them on the validation set. The coverages at 95% precision (C@95%P) and 99% precision (C@99%P) are shown in Table 1. From the table we can see that data cleaning can help increase the coverage at lower precision value, but will slightly hurt the performance at higher precision. This demonstrates the necessaries of achieving good trade off between the removal of noise and removal of hard positive samples. On one hand, as the noisy data are removed and the classification task of the data becomes easier. Thus the coverage is higher at lower precision. But on the other hand some of the challenging data, which help train a better face recognition model, are also removed and the valuable information is lost. So the performance at higher precision is compromised.

Table 1: Coverage of the models trained on original data and cleaned data respectively. A higher number indicates a better result. The results clearly demonstrate the effects of data cleaning (see more dicussions in the text).

Dev1	C@95%P	C@99%P
ResNet-50 on Original data	8.6%	7.0%
ResNet-50 on Cleaned data	13.0%	6.6%
Dev2	**C@95%P**	**C@99%P**
ResNet-50 on Original data	17.2%	4.2%
ResNet-50 on Cleaned data	26.4%	1.4%

4.2 Model Averaging v.s. Feature Fusion

Based on the base models, we have several ways of combining the models. One method is to simply averaging the scores from different models. Another method is to use the proposed feature fusion and classification modelling. We test both methods with our base models on the validation set. The base models used in this experiment are the same as the two ResNet-50 models used in Section 4.1. The results are shown in Table 2. Based on the experimental result, we can see our proposed feature fusion and classification method outperforms simply averaging the results.

Table 2: Effects of different model ensemble methods.

Dev1	C@95%P	C@99%P
Average	20.4%	17.4%
Ensemble	33.6%	22.6%
Dev2	**C@95%P**	**C@99%P**
Average	29.8%	0.8%
Ensemble	46.0%	0.8%

4.3 Performance of Each Base Model

For ResNet-50 and GoogLeNet, the final layers are classification layers and they will naturally produce a probability distribution over all the $76,674$ given subjects in the training set. We sort the predicted probability and pick the top 5 subjects with their corresponding probability as the identification output. For the TripletNet, direct prediction is not possible as it only learns the relative distance between face images of different subjects. So we use the trained network to extract the features of all the images in the training set as gallery features, and extract the features of all the images in the validation set as the probe features. Then for each validation image, we use its corresponding probe feature to search through the gallery and returns the 5 nearest feature vectors in the gallery as the identification result. The confidence scores of the returned nearest neighbours are the reciprocal of the distance between the probe feature and the gallery feature. The coverages at 95% and 99% precision, together with the top 5 and top 1 accuracies of the three base models on Dev1 and Dev2 are reported in Table 3. Dash symbol indicates that the certain precision in not achieved.

The TripletNet is not advantageous in terms of coverages at 95% and 99% precision, as the confidence score is the reciprocal of the distance, which might be very similar in the high dimensional feature space. We include this model for the final ensemble as the top 1 and top 5 accuracies are reasonably good, which indicates the learnt feature from this model is still valuable for our final multi-view model.

Table 3: Performance on Base Models.

Dev1	C@95%P	C@99%P	Top 5	Top 1
ResNet-50	13.0%	6.6%	35.8%	29.2%
GoogLeNet	9.0%	2.8%	37.2%	27.8%
TripletNet	—	—	32.4%	22.6%
Dev2	**C@95%P**	**C@99%P**	**Top 5**	**Top 1**
ResNet-50	26.4%	1.4%	44.8%	39.6%
GoogLeNet	16.2%	12.8%	47.6%	38.2%
TripletNet	0.6%	0.6%	53.8%	48.0%

4.4 Ensemble of Multi-View Models

The final model ensemble involves four models: ResNet-50, GoogLeNet and TripletNet trained on the filtered data, and ResNet-50 trained on the original data. The results are in Table 4 [1]. The corresponding ROC curves of the multi-view ensemble model on Dev1 and Dev2, showing the precision versus coverage, are plotted in Figure 4. Our final model achieves a coverage of 46.1% at 95% precision on the random set and a coverage of 33.0% at 95% precision on the hard set. Due to the use of various deep CNN models with different loss functions, the proposed multi-view model ensemble is effective even at a higher precision. Our model achieves a coverage of 33.9% at 99% precision on the random set and a coverage of 21.1% at 99% precision on the hard set, ranking the fourth and the second place among all the participating teams.

Table 4: Performance of Multi-view Ensemble.

Dev1	C@95%P	C@99%P	Top 5	Top 1
Average	28.4%	20.6%	45.6%	38.4%
Multi-view	40.8%	28.0%	51.0%	46.0%
Dev2	**C@95%P**	**C@99%P**	**Top 5**	**Top 1**
Average	42.6%	2.8%	54.2%	49.4%
Multi-view	50.6%	21.8%	56.2%	52.6%

5. CONCLUSION

In this paper, we introduced our multi-view learning solution for MSR Image Recognition Challenge MS-Celeb-1M. We presented a data cleaning technique to reduce the influence of noisy data, and a multi-view deep representation learning strategy to combine deep networks of diverse structures and loss functions. The experimental results on the provided large-scale face dataset demonstrate the effectiveness of our proposed two-stage approach for the face recognition challenge.

[1]Note that the model we use for this result is different from the model we use during the dry run.

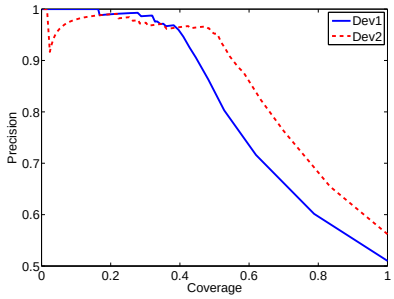

Figure 4: Precision-Coverage Curve of Multi-view Ensemble on the Validation Set.

6. REFERENCES

[1] J. M. Cimbala. Outliers. Technical report, Penn State University, September 2011.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[3] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[4] A. Lucas, R. Van Dijk, and T. Kloek. Outlier robust gmm estimation of leverage determinants in linear dynamic panel data models. *Available at SSRN 20611*, 1997.

[5] 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*, pages 815–823, 2015.

[6] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

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

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[9] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[10] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.