Improved Performance of Face Recognition using CNN with Constrained Triplet Loss Layer

Henry Wing Fung Yeung, Jiaxi Li, Yuk Ying Chung
School of Information Technologies, The University of Sydney, NSW 2006, Australia, Sydney
Email: henrywfyeung@gmail.com, jili2506@uni.sydney.edu.au, vera.chung@sydney.edu.au

***Abstract*—Recognizing human faces is one of the most popular problems in the field of pattern recognition. Many approaches and methods have been tested and applied on the topic, especially neural networks. This paper proposed a new loss layer that can be replaced at the bottom of a neural network architecture in terms of face recognition, called constrained triplet loss layer (CTLL). In order to make more confident predictions and classifications, this loss layer helps the deep learning model to specify further distinguishable clusters between different people (classes) by placing extra constraints on images of the same person (intra-person) while putting margins on images of a different person (inter-person). This proposed constrained triplet loss layer improved the recognition accuracy on faces by 2%.**

***Index Terms*—Convolutional Neural Network, Face Recognition, Inter/Intra-personal Constrains, Performance Optimization**

I. Introduction

The most common procedure of face recognition involves face detection, face alignment, feature extraction and at the end, analysis of the extracted features. Hence, the concept of feature extraction becomes extremely important in order to make clear and confident classifications or predictions [1]. Within the training process, the model used to generate representations is one of the most crucial factors. Many models are used and tested as the face recognizer throughout the years, such as SVM [2] and random forest [3].

The ability to interpret the unique meanings or features hidden within each input image provides better understandings on the problem and data. Recently, together with the advancement in solving the ImageNet classification problem, deep learning has been greatly evolved and deployed into many fields, in particular face recognition. In this domain, the deep neural network model is acting as a feature extractor which produces representations given input images.

In order to produce distinct representations of faces in deep learning, convolutional neural networks (CNN) with various architectures are used for achieving high accuracy. One of the representative models for face recognition is FaceNet which used a convolutional deep neural network with inception architectures [4]. A deep convolutional neural network consists of a few layers, such as a convolutional layer, pooling layer, fully connected layer etc. At the bottom of CNN, there is a loss layer that can be used to adjust learning parameters. It calculates the error and gradients which are based on the input images and then back-propagate the values to previous layers and their neurons for adjustment. Therefore, a properly constructed loss layer can be crucial to train an effective deep neural network model.

Upon generating representations or embeddings of input images, all the images are randomly placed through space which makes the images of same person to be poorly distributed. Triplet loss function ensures the representations of the same person lie within the same "cluster" [5]. That is, images in the projection space will have better distributions by separating different people. Nonetheless, even though simple "clusters" are formed for different people, there are some images within the same cluster can be still poorly distributed due to the variety of poses and age difference of the same person, which can result in wrong classification or very low confidence (probability). In this paper, the constrained triplet loss has proposed to distinguishing this kind of face images. Our results have shown to have an improvement on accuracies by 2% in LFW dataset.

This paper is organized into six sections. The distribution is as follow: in Section II, the problem statement of CNN face recognizer is introduced, followed by the proposed methodology with constrained loss layer in Section III. In Section IV, experiment setups and results are provided. A discussion is carried out in Section V. At the end, future work and conclusion are given in Section VI.

II. Problem Definition

A triplet loss function can be directed used on the output representations or embeddings [4] which can be described in Figure 1.

Unlike the loss function that used in [6], which enables every face of one class to be projected onto the embedding space as a single point. Triplet loss, however, is enforces an extra margin between each positive (same person) and negative (different person) face pair from one class to others. That is, the additional margins ensure the faces of the same person stay within one cluster and therefore, it can be used to discriminate identities. This loss is also mentioned in the context of nearest-neighbor technique [7].

The embedding can be represented by the function $\Theta(x)$, which indicates the representation of x in a d-dimensional space R^d with Euclidean measures. A triplet involves exactly three images, which are denoted as x^a, x^p, and x^n. x^a is called the anchor image which is an input image. The other images of the same person are represented as x^p, whereas the images of different person are described as x^n. As described

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

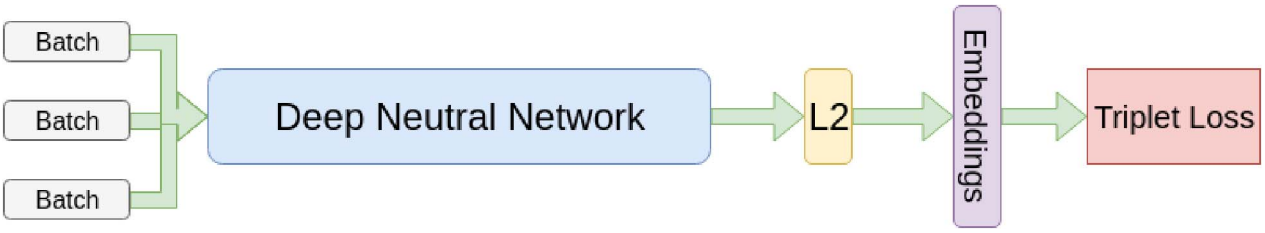

Fig. 1. Triplet Model Structure. The neural network takes a number of batches as input and followed by the deep architecture (CNN). The L2 normalization layer will produce the face representations (embeddings). At the end, the triplet loss function is applied on these embeddings

above, triplet loss function is aimed to push away images of the different person to form clusters in the d-dimensional space.

To have a clearer picture of this process, triplet loss function maximizes the distance between different person. This process can be illustrated in Figure 2.

In terms of mathematics, we are trying to enforce a margin α between $||x^a - x^p||^2$ and $||x^a - x^n||^2$ as shown in Equation 1, where (x^a, x^p, x_n) is a triplet set from all possible triplets T with cardinality N.

$$||x^a - x^n||^2 - ||x^a - x^p||^2 > \alpha, \quad (x^a, x^p, x_n) \in T \tag{1}$$

Therefore, the loss function L can then be inducted as:

$$L = \frac{1}{N}\sum_{i}^{N}\left(||\Theta(x^a) - \Theta(x^p)||^2 - ||\Theta(x^a) - \Theta(x^n)||^2 + \alpha\right) \tag{2}$$

Note that, $\Theta()$ is the function that generates the embeddings of images.

In Equation 2, the loss function requires the value of $||\Theta(x^a) - \Theta(x^n)||^2$ be greater than $||\Theta(x^a) - \Theta(x^p)||^2$ by at least the defined margin α. Although this triplet loss function only defines what the distance between x^a and x^n, and it does not specify how close the images of the same person x^a and x^p should stay within the space. That is, the original triplet loss describes and defines the inter-personal variation, but ignores the intra-personal variation. As a consequence, ignoring intra-personal variation could result in a poor distribution of images of the same person. It is caused by the large average distance between the same person in the cluster due to pose or age variation, leading to inconsistency in recognition outcome.

III. Method

Based on the observation above, a function that specifies intra-class variation is introduced to the loss layer. Besides the part declaring extra distance between x^a and x^n, an additional function is added to the loss function in order to group images of the same person into a well-distributed "cluster". A new parameter γ is introduced which is used to determine desired distance between images of the same person. In this stage, images of different person x^n are no longer the concerns since they are covered by enforcing extra margin in Equation 1. That is, the distance of x^a and x^p should be less than a certain value γ. This relation has to be satisfied as Equation 3.

$$||\Theta(x^a - x^p)||^2 < \gamma \tag{3}$$

With this proposed formula adding into the function, the new triplet loss can be defined as Equation 4.

$$L = \frac{1}{N}\sum_{i}^{N}\Big[\left(||\Theta(x^a) - \Theta(x^p)||^2 - ||\Theta(x^a) - \Theta(x^n)||^2 + \alpha\right) + \left(\kappa||\Theta(x^a) - \Theta(x^p)||^2 - \gamma\right)\Big] \tag{4}$$

Where $\left(||\Theta(x^a) - \Theta(x^p)||^2 - ||\Theta(x^a) - \Theta(x^n)||^2 + \alpha\right)$ covers an inter-personal constraint and $\left(||\Theta(x^a) - \Theta(x^p)||^2 - \gamma\right)$ implies an intra-personal constraint. Note that, κ used in this equation is intended to balance the weight between inter-person constraint and intra-person constraint as weighting parameter.

This resulting triplet loss function with intra-person constraint can lead to a smaller sized clusters for each person in the space as shown in Figure 3. Similar work has been done in people identification [5]

As a result, smaller clusters of the different person in the space implies a more confident prediction or classification. That is, whenever this model is making prediction or classifying decisions, it produces an answer with comprehensive considerations.

IV. Experiment and Results

A. Configurations

DNN model requires a powerful graphics card for the training process to finish within a reasonable time. For the experiment below, NVIDIA GTX 980 Ti GPU is used for this purpose.

B. Dataset

The data set used for the purpose of evaluating this model is the well known Labelled Face in the Wild (LFW) [8]. This data set contains a total of 13233 images and 5749 different people. For each image inside this data, it is represented in the size of 250×250.

LFW is designed to study on how to solve problems of recognizing unconstrained faces. Each face in the data set has been labelled with the name of that person, and the only

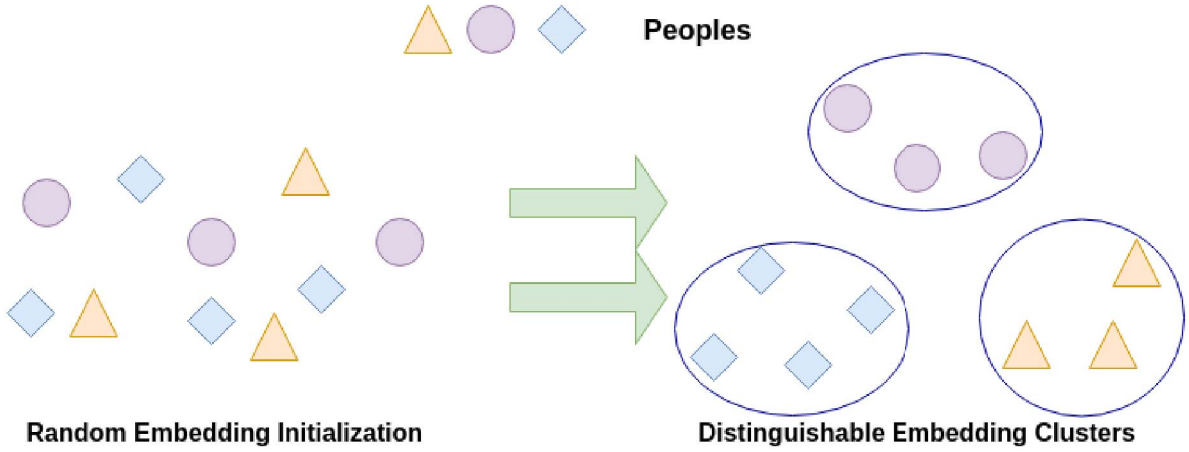

Fig. 2. Illustration of the Triplet Loss Function. The Graph on the left indicates the embeddings of people in a space, which are random located. In the right-hand side, the graph describes the allocation of different embeddings after training process.

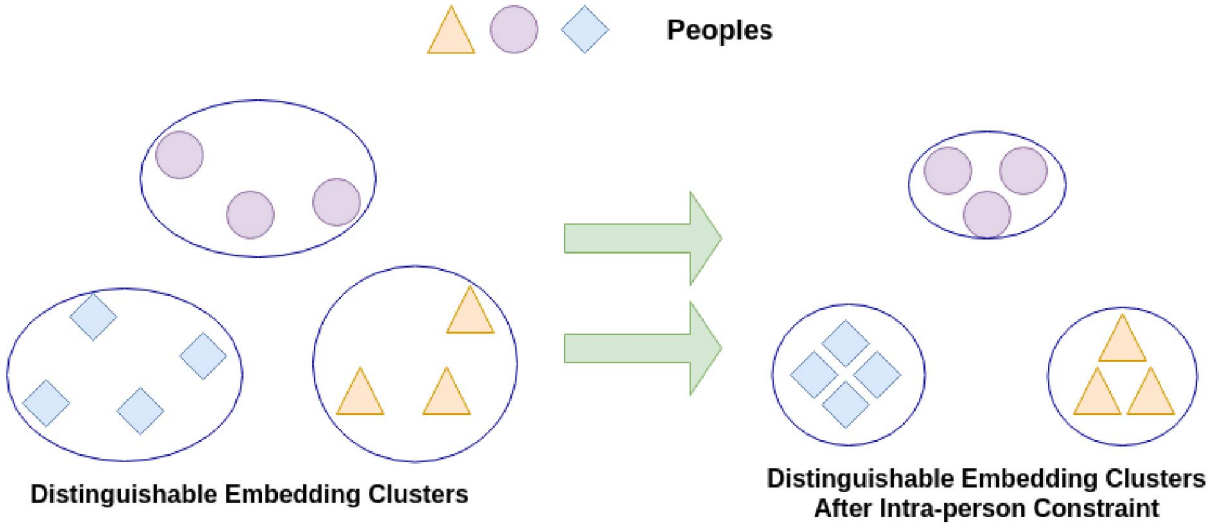

Fig. 3. Illustration On Clusters After Applying Intra-person Constraint. After the constraint is added to loss function, the clusters of each person will grow smaller since the distance of images for same person are pulled closer to each other.

constraint on this data set is the faces which has been tested with Viola-Jones face detector in [8].

Although LFW contains images of 5749 people in dataset, there is a large number of people that only contains one single picture. Therefore, there are only around 1600 people in the data set will have 2 or more pictures. According to the Section III, each triplet is formulated of 3 images, and 2 of them are from the same person. Hence, people with 2 or more images are selected for experiments. In this paper, a set of 1600 people is selected from a total of 9511 images. For training a deep neural network, only the human face will be fed into the model after applying 2D or 3D alignment techniques as shown in Figure 4.

For a better differentiation between the origin triplet loss functions and intra-person constrained triplet loss, the data set is divided equally into two sets for training and testing:

1) Training Dataset: The training data set is selecting the person who has more than 10 pictures of himself. In this case, sufficiently large number of images of the same person are necessary for the constrained loss function to minimize the cluster in the space describe in Figure 3. In our experiment, we have selected around 4500 images and 158 different people within this training set.

2) Testing Dataset: The testing data set consists of the rest of the images from LFW who has more than 2 but less than 10 images. We have used approximately 4800 images with 1600 different people in the testing set.

C. Mini-Batch

DNN is the technology which requires a large amount of training data in order to achieve distinct performance. However, these data is too heavy to fit into memory unless there is a ”super computer”. Therefore, training data needs

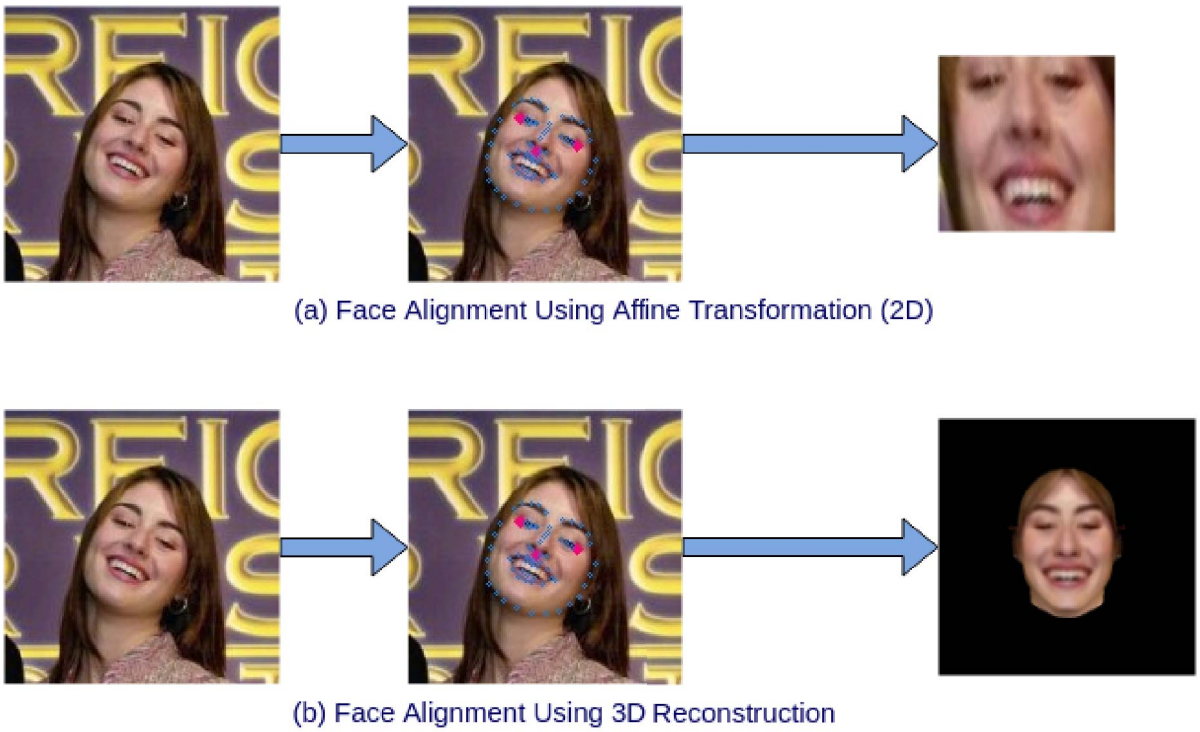

Fig. 4. A sample picture selected from LFW [9]. After applied alignment technique, the only input for neural network will be the pure human face without any backgrounds.

to be distributed into many mini batches, and single batch becomes the basic unit to train this DNN model. In this setup, a total of 150 batches will be used in each iteration or epoch.

Besides the memory problem, how to prevent overfitting is another problem that we need to consider. There are a few ways to prevent overfitting in a neural network, such as setting up dropouts or stop the training procedure earlier. In this experiment, a simpler method is carried out as following:

i) For each mini batch, 15 people are selected randomly, and for each person, we select at most 10 random pictures of themselves. Therefore, at most 150 pictures are selected in each batch. Note that, between the batches, selections have a chance to be overlapped.

ii) For each iteration or epoch, small-sized mini-batch is preferred, since it has good effects in speeding up convergence process when using Stochastic Gradient Descent (SGD) [10].

D. Triplet Selection

Florian et al. described that how to select triplets among training data set has a great impact on models [4]. In this experiment, online triplet selection is used to generate triplets for every mini-batch. Within a mini-batch, since we need to distinguish inter-person variance, all the positive pairs (x^a, x^p) are fed to the loss function, while only the negative pairs with large difference are picked from all negative pairs (x^a, x^n) [4].

However, Florian et al. mentioned that selecting "hardest negative pair" may result in an unexpected local minimum in early training stage [4]. Therefore, "semi-hard" negative pairs are only selected for this purpose. "Semi-hard negative pair" is the pair that lies within the predefined margin α, but it is still "hard" since the distance of $(anchor, negative)$ pair is larger than $(anchor, positive)$ pair as Equation 5.

$$||f(x^a - x^p)||^2 < ||f(x^a - x^n)||^2 \tag{5}$$

E. Training Model

Due to the limitation of the hardware, training on a large deep neural network model is extremely time-consuming. Therefore, a smaller-sized model is used in this experiment on testing impacts on constrained triplets. Instead of using a big deep neural network, a representative small version of FaceNet, NN4 model [11], is used for experiments in this paper.

The detailed architecture of the deep model is given in Table I.

Note that, in the projection column, the number of reduced dimensionality is specified as "p". For example, "128p" indicates the dimension is reduced to 128 dimensions after either max pooling or L_2 pooling.

Upon training the target deep model, both original triplet loss and constrained triplet loss are presented separately in the training process. In this case, the triplet loss function is used to calculate the distance between images and backpropagate the gradients to the deep model rather than using another criterion.

Recall the constrained triplet loss function (Equation 4) from Section III,

TABLE I
A SMALLER DEEP NEURAL NETWORK MODEL VERSION (NN4) OF FACENET. $\#3 \times 3R$ AND $\#5 \times 5R$ DENOTE THE NUMBER OF 1×1 FILTER USED BEFORE THE 3×3 AND 5×5 CONVOLUTIONAL LAYERS. POOLING PROJECTION SPECIFIES THE PROJECTION OR REDUCTION LAYER AFTER APPLIED MAX POOLING (M) OR L_2 POOLING (L_2). "NP" IS DENOTED AS THE N-DIMENSION AFTER POOLING.

Layer	Input	Output	#1×1	#3×3R	#3×3	#5×5R	#5×5	Pool Proj.
conv1	$96 \times 96 \times 3$	$48 \times 48 \times 64$						
maxp-¿norm	$48 \times 48 \times 64$	$24 \times 24 \times 64$						m,$3 \times 3, 2$
inception(2)	$24 \times 24 \times 64$	$24 \times 24 \times 192$		64	192			
norm-¿maxp	$24 \times 24 \times 192$	$12 \times 12 \times 192$						m, $3 \times 3, 2$
inception(3a)	$12 \times 12 \times 192$	$12 \times 12 \times 256$	64	96	128	16	32	m,32p
inception(3b)	$12 \times 12 \times 256$	$12 \times 12 \times 320$	64	96	128	32	64	L_2,64p
inception(3c)	$12 \times 12 \times 320$	$6 \times 6 \times 640$		128	256,2	32	64,2	m,$3 \times 3, 2$
inception(4a)	$6 \times 6 \times 640$	$6 \times 6 \times 640$	256	96	192	32	64	L_2,128p
inception(4e)	$6 \times 6 \times 640$	$3 \times 3 \times 1024$		160	256,2	64	128,2	m, $3 \times 3, 2$
inception(5a)	$3 \times 3 \times 1024$	$3 \times 3 \times 736$	256	96	384			L_2,96p
inception(5b)	$3 \times 3 \times 736$	$3 \times 3 \times 736$	256	96	384			m,96p
avg pool	$3 \times 3 \times 736$	$1 \times 1 \times 736$						
fc(linear)	$1 \times 1 \times 736$	$1 \times 1 \times 128$						
l_2 norm	$1 \times 1 \times 128$	$1 \times 1 \times 128$						

$$L = \frac{1}{N}\sum_{i}^{N}\Big[\big(||\Theta(x^a) - \Theta(x^p)||^2 - ||\Theta(x^a) - \Theta(x^n)||^2 + \alpha\big) + \kappa\big(||\Theta(x^a) - \Theta(x^p)||^2 - \gamma\big)\Big]$$

The enforced margin between different person α is chosen to be 1, the distance constraints of the same person γ is set to 0.01, and the weight factor κ has the value of 0.01.

F. Pair Testing

For the purpose for keeping track on every epoch, online testing is enabled. At the end of each epoch, an evaluation is carried out to evaluate how good the current model is.

The testing method is simple but meaningful. Evaluation is based on pairs of images which are selected from the testing data set mentioned in Section IV-B. For each pair in the testing, both pictures are fed into the current model. However, the loss layer is removed in this process, since there is no need to backpropagate the adjusted weights to the model. That means only a forward propagation is operated to get image representations or embeddings in this stage.

Once both representations are acquired, the similarity is being calculated using Euclidean Distance as shown in Equation 6

$$dist(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{\big(a_1 - b_1\big)^2 + \big(a_2 - b_2\big)^2 + ...} \qquad (6)$$

There are in total 6000 image pairs for evaluation, and the pairs are randomly selected from the testing data set. However, in order to conduct a fair evaluation, 60% of image pairs are selected from the same person, while 40% are produced from different person.

Our experiment has been conducted with 10-fold cross validation to get the meaningful results. All the accuracy results shown in Table III and IV are produced by calculating the average accuracy from 10 folds. Data are equally distributed in each fold. For each iteration, the experiment is trained on 9 folds and it computes accuracies on the remaining 1 testing fold. When distances between pairs of images are being calculated, a threshold is required to determine whether the 2 images belong to the same person or not. The best threshold for this job is being discovered and selected on training folds, then the same threshold is used for evaluation on the remaining testing folds.

G. Results

The experiment is focused on studying the improvement on placing constrained layer into a deep neural network model. As specified above, the evaluation process is to predict or verify whether a pair of images is of the same person or not.

Both NN4 and Constrained NN4 described in Table II are the models used for experiments mentioned in Section IV-E.

From 10-fold cross-validations, the best thresholds for both NN4 and its constrained model are selected to be 0.91 and

TABLE II
THE MODELS DESCRIBED IN THIS EXPERIMENT ON EVALUATION OF 6000 IMAGE PAIRS GENERATED AS SECTION IV-F. NOTE THAT, THE CONSTRAINED MODEL HAD BEEN IMPROVED BY 2% IN RECOGNITION ACCURACY.

Model	Accuracy
NN4	0.8173
NN4 with Constrains	0.8372

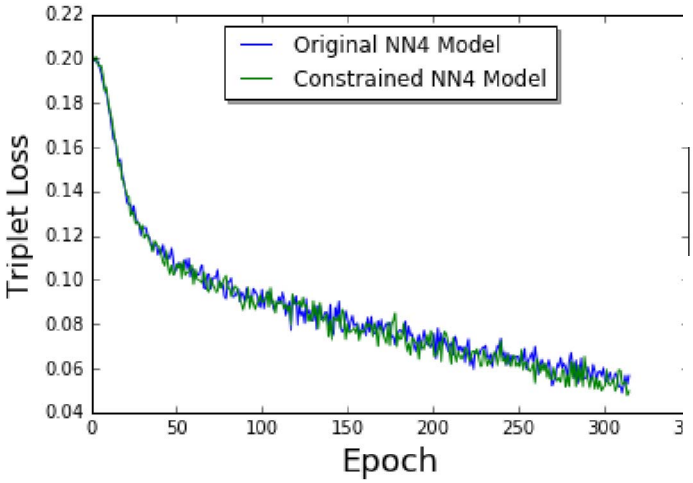

Fig. 5. Comparison of Triplet Loss and Constrained Triplet Loss on NN4 Model.

1.00 respectively. The details of threshold selection in every fold are specified in Table III and Table IV.

Note that, the threshold is selected in the cross-validation on training data, and then this particular threshold is used as a fixed parameter during the evaluation process.

Figure 5 demonstrates the convergence of triplet loss and its constrained version. Due to the time consumption, the training process is stopped at around $320th$ epoch (or iteration). Even though the constrained triplet loss function is adding extra errors into the original function (Equation 4), both loss functions have a similar converging rate.

In Figure 6 and Figure 7, both graphs represent the Receiver Operating Characteristic (ROC) curve for original NN4 model and constrained NN4 model (described in Section IV-E) respectively. A curve of the best model should have TPR value of 1 all the time. However, both model in this experiment cannot reach this optimal since the data setup.

Every curve of the model in Figure 6 and 7 is obtained from 10-fold cross validation by averaging the threshold from each fold. Results of Eigenface and OpenBR is acquired from the original LFW data set [8]. Besides, human cropped ROC result is provided by Kumar et al. [12].

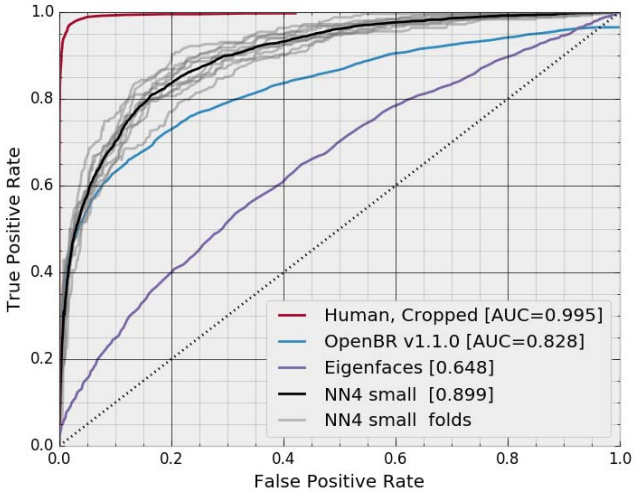

Fig. 6. Representation of Receiver Operating Characteristic curve of Original NN4 Architecture

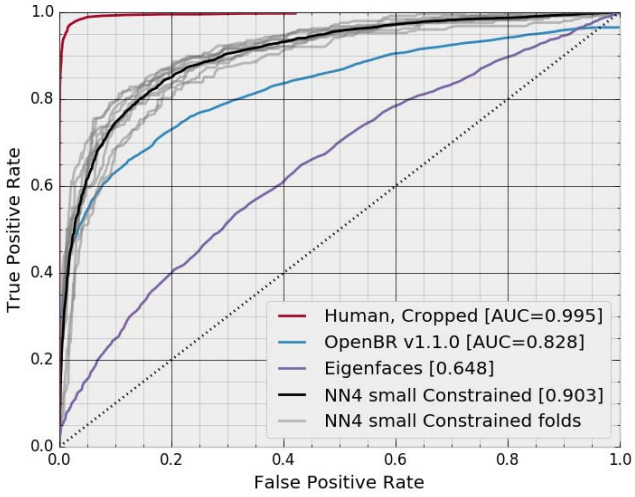

Fig. 7. Representation of Receiver Operating Characteristic curve of Constrained NN4 Architecture

V. DISCUSSION

From the results in Section IV-G, it is obvious that the model with additional intra-person constraints on triplet loss dominants the original model with just inter-person constraints by approximately 2% on accuracy. However, different experiment setups could result in distinct outcomes with positive or even negative impacts, especially different training dataset.

In the studies of deep learning and neural network, training data is always the center of development, especially the methodologies used for training.

Recall the intra-person constrained triplet loss function in Section III, it can be described in two different parts as Equation 7.

TABLE III
THRESHOLDS FOR NN4. DESCRIBES BEST THRESHOLDS SELECTED FOR EACH FOLD IN EVALUATION.

Fold	Threshold	Accuracy
0	0.92	0.81
1	0.92	**0.83**
2	0.91	0.82
3	0.89	0.82
4	0.91	0.78
5	0.91	0.81
6	0.91	0.81
7	0.89	0.82
8	0.91	0.81
9	0.91	**0.83**

TABLE IV
THRESHOLDS FOR CONSTRAINED NN4. DESCRIBES BEST THRESHOLDS SELECTED FOR EACH FOLD IN EVALUATION.

Fold	Threshold	Accuracy
0	0.88	0.81
1	1.00	**0.84**
2	1.00	**0.84**
3	1.00	0.83
4	0.88	0.81
5	0.88	0.83
6	0.88	0.82
7	0.90	0.83
8	0.88	0.83
9	0.88	0.83

$$L = \frac{1}{N}\sum_{i}^{N}\Bigg[\underbrace{\left(||\Theta(x^a) - \Theta(x^p)||^2 - ||\Theta(x^a) - \Theta(x^n)||^2 + \alpha\right)}_{\text{Inter-Person Enforcement}} + \underbrace{\left(\kappa||\Theta(x^a) - \Theta(x^p)||^2 - \gamma\right)}_{\text{Intra-Person Constrain}}\Bigg] \tag{7}$$

When there is a large number of images of the same person, a large amount of $||\Theta(x^a) - \Theta(x^p)||^2$ is also added into loss function since it will be frequently called. However, if the image pairs of the same person are seldom within the training data set, the second part in Equation 7, intra-person constraints, becomes extremely small and makes this constraint on the model insignificant. That is, if there are only very few image pairs of same person existing in training data set, this particularly constrained loss function would behave almost identical to the model with only inter-person enforcement as described in Section III and below.

$$L = \frac{1}{N}\sum_{i}^{N}\Bigg[\underbrace{\left(||\Theta(x^a) - \Theta(x^p)||^2 - ||\Theta(x^a) - \Theta(x^n)||^2 + \alpha\right)}_{\text{Inter-Person Enforcement}}\Bigg]$$

Another simple comparison is setup to validate this constraint theory. Recall the training and testing setups described in Section IV-B, it has 158 people with 4600+ images for training and 1600 people with 4800+ images for testing. Within the comparison, the training and testing data are getting swapped for this purpose. In our experiment, this particular comparison model is trained on the data of 1600 different people with around 4800 images and then test on the data of 158 different people with approximately 4500 images.

TABLE V
THE SWAPPED TRAINING AND TESTING DATA SET EXPERIMENTING ON EVALUATE 6000 IMAGE PAIRS WITH THE METHODOLOGY DESCRIBED IN SECTION IV-F.

Model	Accuracy
NN4	0.8363
NN4 with Constrains	0.8340

Even though the settings are identical, the best thresholds that are used by the following test method are being selected with the value 1.09 and 1.16 for the original model and constrained model respectively.

Note that, the convergence rates for both models are similar to the experiment results in the previous section since the batch size and triplet selection method are identical.

The comparison of results is described in Table V. Both original and constrained model achieved accuracy around 0.83. However, the difference between constrained and non-constrained network is insignificant comparing to the experiment in Section IV, where the improvement is 2%.

The non-constrained model will behave similar to the model with constraints when the impact of intra-person constraint is insignificant. When swapping the training/testing data set, the resulting training data set consists of pictures of around

1600 people, but with only 3 or 4 images for each person, which indicates a low coverage for each class. The small numbered intra-person images make the constrained loss function performs in a "weak constraint" condition, which makes constrained model to become less "constrained" or no "constrained".

VI. Conclusion

In this paper, an intra-person constrained triplet loss function is proposed to improve the performance of CNN model on recognizing faces. This constrained triplet not only enforces an extra margin on representations of different people but also keeps the representations of the same person closer to make a more confident response. Moreover, the result on intra-person constrained criterion in deep model demonstrates that the improvements in performance is significant and can be extended to research with similar problem formulation.

VII. Acknowledgement

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

[1] M. Nikravesh, I. Guyon, S. Gunn, and L. Zadeh, "Feature extraction: Foundations and applications," 2006.

[2] K. I. Kim, K. Jung, and H. J. Kim, "Face recognition using kernel principal component analysis," *IEEE signal processing letters*, vol. 9, no. 2, pp. 40–42, 2002.

[3] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, "Real-time facial feature detection using conditional regression forests," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2578–2585.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[5] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems*, 2014, pp. 1988–1996.

[7] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in neural information processing systems*, 2005, pp. 1473–1480.

[8] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," in *Advances in Face Detection and Facial Image Analysis*. Springer, 2016, pp. 189–248.

[9] 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, Tech. Rep., 2007.

[10] D. R. Wilson and T. R. Martinez, "The general inefficiency of batch training for gradient descent learning," *Neural Networks*, vol. 16, no. 10, pp. 1429–1451, 2003.

[11] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

[12] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 365–372.