Eliminating Intra-class Variance in Classification Neural Networks with Adversarial Losses

Siva Chaitanya Mynepalli, Satyaki Chakraborty, Rishi Madhok

Robotics Institute, Carnegie Mellon University

Abstract. The established procedure to train classification neural networks has been to employ a softmax layer to classify an embedding into given number of classes. The softmax function approximates boundaries between classes as angular bisectors in N-dimensional space. Therefore, there is huge variance in the embeddings of a given class. It is necessary to eliminate this intra-class variation to confidently identify out-of-distribution samples while employing a classification network in the real world. Additionally, eliminating intra-class variance might lead to performance gains in face recognition algorithms, as has been observed previously. In this project, we would explore novel ways of reducing intra-class variance. Particularly, we propose to employ an adversarial network to penalize intra-class variance thereby eliminating variation in the original classification network.

Keywords: Deep learning, Adversarial losses, Face Recognition

1 Introduction

Why should we care about intra-class variance? Deep classification networks perform exceeding well on closed datasets. However, they are not designed to function well in open sets. In open sets, it is possible to observe object classes that have never been seen during training. For example, a classification network trained on imagenet would never have seen what a goldfish looks like.

Classification networks confidently predict unseen objects to one of the N known training classes. However, we want our network to identify them as out-of-training distribution classes.

The current best approach to do this fits one-vs-all classifiers on the embeddings of the penultimate layer. A test image is predicted as out-of-distribution if

it is rejected by all the one-vs-all classifiers. Obviously, the one-vs-allthese classifiers would perform better if the embeddings belonging to the same class are compact and discriminative., i.e they have less intra class variance.

Much research has been conducted on both face identification and face verification, with greater focus on the latter. Research on face identification has mostly focused on using closed-set protocols, which assume that all probe images used in evaluation contain identities of subjects that are enrolled in the gallery. Real systems, however, where only a fraction of probe sample identities are enrolled in the gallery, cannot make this closed-set assumption. Instead, they must assume an open set of probe samples and be able to reject/ignore those that correspond to unknown identities.

Another very relevant application is face recognition. Here, the subject identities of test images are not usually present in training data. The classifier networks trained on face data sets are used as feature extractors. Features extracted from a 'probe' or test image is then compared against a the features extracted from a 'reference' image to verify if a given pair of images belong to the same identity or not. Again, it is imperative that features belonging to the same identity are as should be close to each other.

That is, a feature of an identity should be most similar to features of the same identity irrespective of latent factors like pose, illumination, and expression.

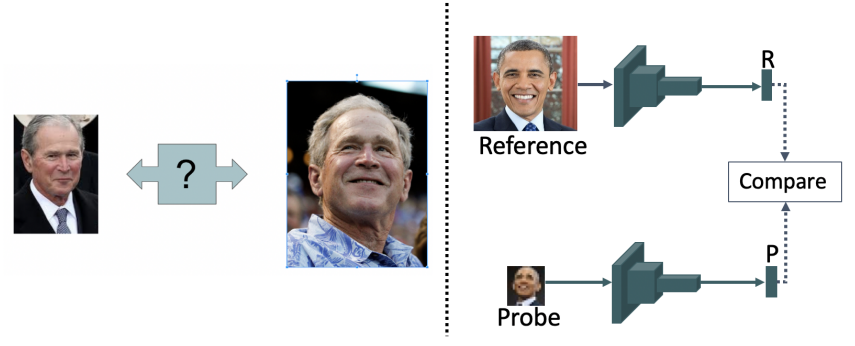


Fig. 1: The figure above shows a general pipeline for a face verification system

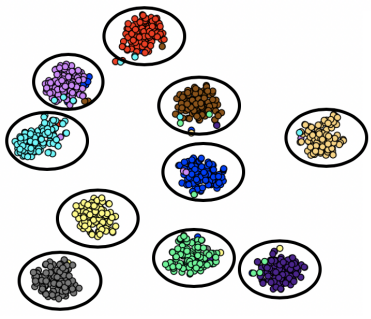


Fig. 2: The figure above shows distinct class embeddings

2 Related work

The necessity for open-set face recognition has been widely acknowledged for over a decade [9]. However, only a few works such as, e.g., [15] [3] [5] [7] [6] [11] have addressed the problem by predominantly focusing on obtaining an ad hoc rejection threshold on similarity score under an openset evaluation protocol [9]. For example, Best-Rowden et al. [6] showed that a simple thresholding of a commercial of the shelf (COTS) algorithm works perfectly for verification, but does not provide decent open-set identification performance. The development of classifiers that explicitly model probability of inclusion [2] of probe samples with respect to a region of known support of the gallery has received far less attention in the face recognition community. For security-oriented applications where the enrollment process must be quick, the cost of false alarms is high, and the cost of missed alarms is even higher, the notion of using an ad hoc rejection threshold on similarity is problematic because the concept of unknown may change as more samples are enrolled and data bandwidth is variable, so a one size fits all threshold may not work well. A classifier that can efficiently be retrained with each enrolled gallery template to autonomously assess the probability that probe data comes from regions of known support on behalf of the gallery while considering variable data bandwidths is a far more appealing alternative.

Thus, the motivation for applying classifiers that are open-set-by-design to face recognition problems is manifested. Several such classifiers have been developed in the computer vision community [2] [12] [13] [1], but their application has been limited to toy problems on modifications of canonical computer vision datasets like MNIST [14], or to generic object recognition problems like the Im-

ageNet challenge [10]. However, object recognition problems inherently differ from biometric applications insofar as they are far more coarse-grained, the notion of enrollment does not exist, and deep learning solutions can be obtained by training an end-to-end network on the training set and using that end-to-end network as a classifier.

Face identification systems that use deep features [8] [16] [17] [4], by contrast, use truncated forward passes over pre-trained networks to extract features at enrollment or query time. The networks are trained in an end-to-end manner on labeled face identities, which generally differ from the identities enrolled into gallery templates. Templates are constructed during enrollment, e.g., by collecting extracted feature vectors from several images of each given subject. At query time, probe templates consisting of one or more extracted feature vectors of one subject, are matched against gallery templates. The identification procedure commonly takes the form of finding the gallery template with the sample of maximum similarity to the corresponding probe. Cosine is a common measure of similarity between feature vectors extracted from a face network [16] [4]. Particularly, when templates vary in number of images, feature vectors are sometimes aggregated for a given identity prior to matching, e.g., by taking the mean feature vector [8] [4].

3 Method

Most classification networks used for classification use categorical class entropy is used as a supervisory signal. This is sufficient to perform well in normal classification settings like object or action classification, as all possible testing examples are present in the training set. However, for open set problems like face recognition, it is not possible to have all testing examples in the training set. Therefore features used for face recognition need to be discriminative and not just separable [18]. Being discriminative ensures that predicting a label with nearest neighbors still outputs a reasonable solution for any target example. Several methods have been proposed to achieve such discriminative features. Two such methods are contrastive loss and triplet loss. In both these approaches the number of possible training examples rises exponentially. One needs to find suitable sampling schemes to sample training example pairs or triplets to ensure fast convergence. An unideal scheme might lead to less than optimal performance.

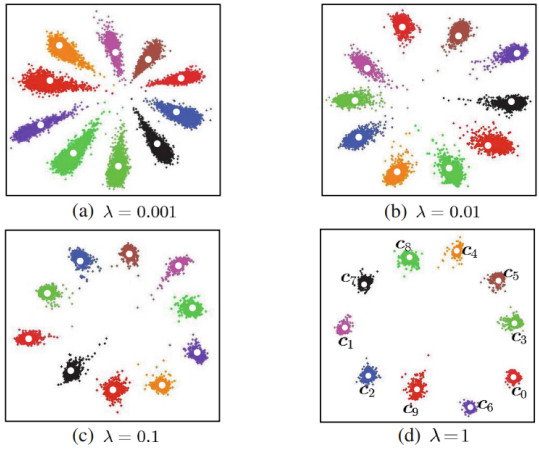


Fig. 3: The figure shows the distribution of deep features learned under the joint supervision of categorical cross-entropy and center loss. The different colors represent examples drawn from different classes. λ denotes the weight of the weight of the center loss in the overall loss function given in equation shown below on page 6. The various c_i's represent the centers of the various classes.

3.1 Center loss

An efficient way to ensure that the deep features are discriminative and generalizable to open set settings is to use center loss. In this method, the authors proposed to learn a center (or mean) for each class and penalize the euclidean distance between this mean and a training example from the same class.

$$L_{\mathrm{c}} = \frac{1}{2}\sum_{i=1}^{m}\|x_i - c_{y_i}\|_2^2 \tag{1}$$

where, c_{y_i} is the y_ith class centers which need to be updated after every iteration. This is inefficient. As a workaround, the authors proposed to update the class centers with the mini-batch of sampled training examples weighted by a constant α. The update operation is shown in the following equation.

$$\frac{\partial \mathcal{L}_C}{\partial \boldsymbol{x}_i} = \boldsymbol{x}_i - \boldsymbol{c}_{y_i}$$

$$\Delta \boldsymbol{c}_j = \frac{\sum_{i=1}^{m} \delta(y_i = j) \cdot (\boldsymbol{c}_j - \boldsymbol{x}_i)}{1 + \sum_{i=1}^{m} \delta(y_i = j)}$$

$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C$$

$$= -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T \boldsymbol{x}_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T \boldsymbol{x}_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$$

The center loss is efficiently contracts the distance between deep features of the same class. To maximize inter-class separation, center loss is combined with the normal softmax + categorical cross entropy loss, as shown in the above equation. By training on both these losses jointly, the discriminability of deep features is greatly enhanced as shown in Figure 6.

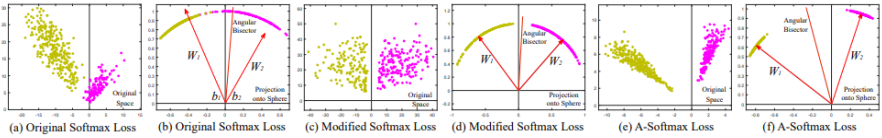
3.2 A study in softmax

Another method to reduce intra-class variance is to modify the softmax layer to restrict the space occupied by embeddings of the same class. We describe the effect of various modifications to softmax in this section.

Softmax: A normal softmax function which classifies a given embedding into one of N classes. Its operation can be described by the following equation,

$$p_i = \frac{exp(W_i^T x + b_i)}{\sum_{j=1}^{N} exp(W_j^T x + b_j)} \tag{2}$$

Consider a softmax designed for binary classification. An input is classifier into class 1 if $p_1 > p_2$ and vice-versa. From Equation 4, we observe that $W_1^T x + b_1$ and $W_2^T x + b_2$ determine this decision. We can rewrite these expressions as $\|W_1\|\|x\|cos(\theta_1) + b_1$ and $\|W_2\|\|x\|cos(\theta_2) + b_2$. The decision boundary between these expressions is dependent on the magnitude of $\|W_i\|$. Therefore, it might cause overlap between embeddings of neighboring classes as seen in Figure 7. **Modified softmax:** By normalizing the weights and zeroing the biases, we find that the decision boundary is converted to the angular bisector of W_1 and W_2. This ensures that there is no overlap between embeddings of adjacent classes



(a) Original Softmax Loss (b) Original Softmax Loss (c) Modified Softmax Loss (d) Modified Softmax Loss (e) A-Softmax Loss (f) A-Softmax Loss

Fig. 4: The figure shows the effect of various modifications of softmax function. For this experiment, the authors designed a CNN which employs 2D embeddings to classify images on a subset of CASIA face dataset. Yellow dots represent one class and purple dots represent the other. It can be observed that the features learned by normal softmax function, although separable, overlap with embeddings of different class. The modified softmax loss function which has normalized weights in the softmax layer generates embeddings which are discriminable. The angular softmax function massively increases the angular margin of the learned embeddings.

making them discriminable. The operation of such a modified softmax function are given by the following equation.

$$p_i = \frac{exp(\|x\|cos(\theta_i))}{\sum_{j=1}^{N} exp(\|x\|cos(\theta_j))} \tag{3}$$

Angular softmax: Although the emebddings generated by training a clssification network do not overlap with neighboring classes, it is imperative that we introduce an angular margin between embeddings of neighboring classes to maximize discriminability. Instead of including a new loss function to achieve this, the authors proppose to modify the softmax function further to incorporate angular margins. Consider an embedding x of an example from class 1, and θ_i be the angle between x and W_i. Modified softmax requires that $cos(\theta_i) > cos(\theta_j) \forall j$. Modifying this condition to $cos(m\theta_i) > cos(\theta_j) \forall j$ with $m >= 2$ results in a stricter decision boundary because a lower bound value of $cos(\theta_i)$ should be the greatest. From an angular perspective, correctly classifying x into class 1 requires $\theta_1 < \theta_2/m$. This is a stricter condition. The operation of such a softmax function is given by,

$$p_i = \frac{exp(\|x\|cos(m\theta_i))}{\sum_{j=1,j\neq i}^{N} exp(\|x\|cos(\theta_j)) + exp(\|x\|cos(m\theta_i))} \tag{4}$$

Such modification is more helpful than center loss because the general practice to find nearest neighbors is to find the nearest neighbors on the basis of angu-

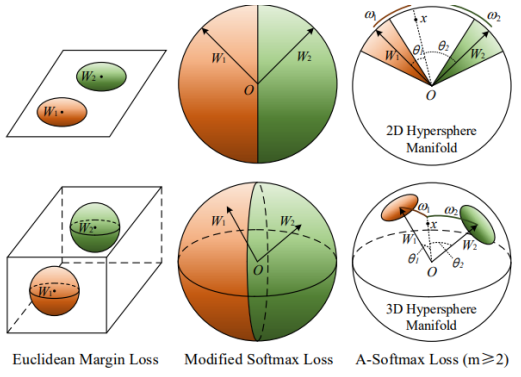


Fig. 5: An illustration of embeddings generated by center loss vs angular softmax.The compactness of embeddings generated by angular softmax is more relevant because of the general practice of identifying nearest neighbors on teh basis of angular distance.

lar difference. The difference between embeddings generated by center loss and angular softmax is shown in Figure 5.

Our method: Contrary to these methods, we devised a training scheme for classification networks which employs adversarial losses to eliminate intra-class variance. The classifier network (C) generates an embedding which is input to an adversarial network (A). The adversarial network tries to distinguish between embeddings of images of the same identity. If it is successful, we penalize the original classifier network by a loss proportional to difference in predicted probabilities, which is the adversarial loss. Similar to center loss, we train such classification networks jointly with both softmax and adversarial losses to maximize inter-class separability while minimizing intra-class variance.

How do we implement this? The original classifier network takes an image as input and generates an embedding. The embedding is passed through a softmax layer to be predicted into one of N classes, and categorical cross-entropy is used as one component of our loss function. The embedding generated by the classifier is then input to an adversarial network which classifies this embedding into one of 2N classes. The first N classes are formed by N anchor images, one from each training class. We want the network to overfit to these anchor images. The next N classes are composed of images other than the anchor image for each of N

classes. The probabilities output by the adversarial network on an input are given by,

$$p_{adv} = A(C(X)) \tag{5}$$

where A is the adversarial network, C is the original classification network, and X is the input example. $C(X)$ is the embedding generated by the classification network. Now, the probability of class i of $2N$ classes in the adversarial network is given by $p_{adv}(i)$.

The classifier trained on the normal softmax loss, as given by 6. It is also penalized if the adversarial network is able to accurately identify if a given image is not the anchor image. The penalty is proportional to the difference in predicted probabilities of the non-anchor class and the anchor class, as given by Equation 7. The joint loss is a weighted combination of both these losses, as shown in Equation 8.

$$L_{softmax} = -\sum_{i=1}^{N} log \frac{exp(W_i^T A(X) + b_i)}{\sum_{j=1}^{N} exp(W_j^T A(X) + b_j)} \tag{6}$$

$$L_{adv} = p_{adv}(2i) - p_{adv}(i) \tag{7}$$

$$L = L_{softmax} + \lambda(L_{adv}) \tag{8}$$

Notice that the adversarial loss would be minimum when the predicted probability for the anchor class ($2i$) is 1, and the probability of the non-anchor class is 0. That is, we want the embedding of a non-anchor image to be so close to the embedding of an anchor image that they are not distinguishable by an adversarial network which is overfit to the anchor images. As all embeddings are now close to the anchor embedding, transitivity implies they are close to each other.

4 Implementation details

One can observe that this architecture is very similar to Generative Adversarial Networks. As all GANs, training this network is hard. It is harder because there are 2*N classes for the adversary rather than the traditional real/fake classes. The major issue is that each update of the classified causes a drift in the embedding of the anchors. If the adversary network does not immediately adapt to this change,

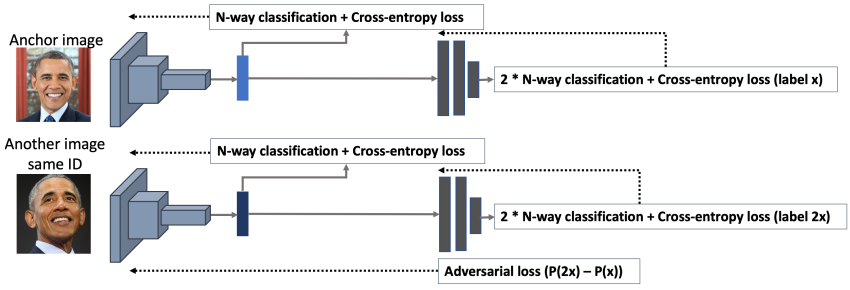


Fig. 6: The figure illustrates our training scheme. One image of each identity in the training dataset is randomly chosen an 'anchor image'. All other images are 'non-anchor images'. The classification network is trained with categorical-crossentropy loss to distinguish between images of different classes. The adversarial network is trained to take the embeddings generated by the classification network and classify them into one of $2N$ classes. The ith class in first N classes is composed of an anchor image. The $i + N$th class is composed of all other images of the same identity as the anchor image in ith class. The classification network is also trained to minimize the adversarial loss given by the difference of probabilities output by the adversarial network for the $i + N$th class and ith class $\forall i$ respectively.

it will never be able to accurately distinguish between anchor and non-anchor images. Also, as the classifier is penalized for any difference between their embeddings, it breaks down and outputs similar embeddings for images across all classes.

We circumvent this issue with an innovative sampling scheme. In order to make sure that the discriminator adjusts to the most recent update for the classifier weights, it is important that the same batch of anchor images are used for classification in the discriminator. Thus, for two consecutive parameter update steps of the embedding generator and the embedding discriminator it is important to fix the anchor images but we can randomly sample the non anchor images. For instance, in our case, given a batch size of 512, we randomly sample with replacement 256 identities out of total N identities. First we sample the anchor image of each of the sampled 256 identities. Then we sample one non-anchor image from them. The batch to train the classifier is composed of the sampled anchor and non-anchor images from the 256 identities. And immediately after the classifier

is updated with this batch, the adversary is updated with a batch composed of the same set of anchor images and a different set of non-anchor images. This ensures that the adversary is in-step with the classifier. Also, batch sampling procedure is more efficient as we reuse half of the images from the classifier batch to train the adversary.

5 Results

In this section we show some qualitative results demonstrating the impact of our training scheme. The three columns show 2D embeddings output by a CNN trying to classify between 2, 3, and 8 classes respectively. The top row illustrates the 2D embeddings output by a vanilla classifier trained with softmax and categorical cross-entropy. Similarly, the bottom row depicts embeddings generated by our model for the same set of images. It is important to note that all these embeddings are directly taken from the classifier without any form of dimensionality reduction (like PCA or t-SNE), which is why we make the embedding space of the classifier a 2D vector space.

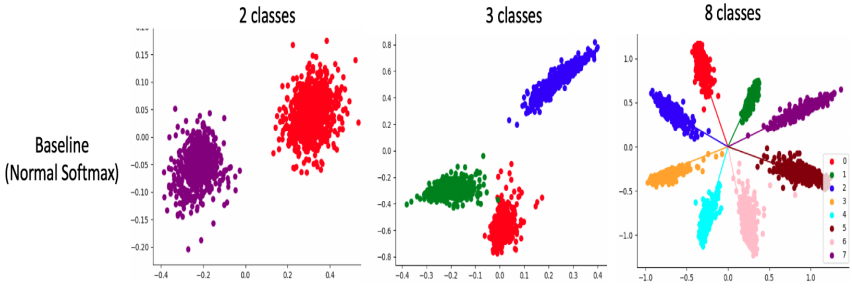


Fig. 7: An illustration of embeddings generated by vanilla classifier with cross entropy loss.

In figure 7 we see the 2D embeddings from the vanilla classifier. Embeddings belonging to the same identity are shown in the same colour. The variance along the second principal component of the clusters denotes the intra class variance. However, the variance along the first principal component is not as significant since we normalize the embedding space before passing it into the softmax layer.

The normalisation is done because we often use the inverse of the cosine distance as a similarity metric for comparing embeddings.

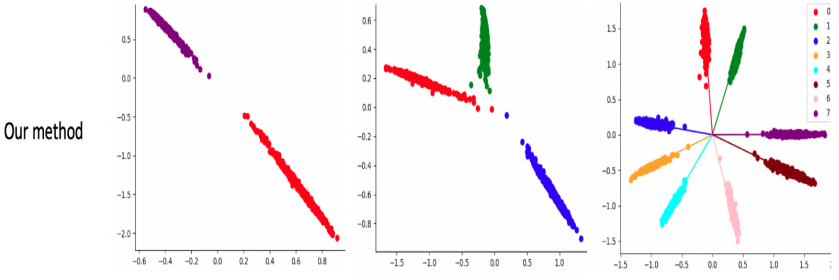


Fig. 8: An illustration of embeddings generated by our novel loss function. We can observe that our classification network learns to output embeddings with reduced intra-class variance.

In figure 8 one would clearly observe that the intra-class variance has effectively been eliminated. This can be concluded from the fact that the variance along the second principal component of the individual clusters have been reduced significantly. Thus, our model represents a class with a single direction. Note that since these embeddings are normalized before being input to the softmax layer, they get projected on to the surface of a sphere, where the embeddings belonging to the same cluster (denoted by the same colour) clump together more densely resulting in increased compactness. This shows that our model generates discriminative embeddings with little intra-class variance.

6 Conclusion

In this report, we show a simple yet effective method of reducing intra class variance for embeddings belonging to the same identity. As mentioned earlier, our goal is to be able to generalise well for out of distribution classes which is often a challenge for face recognition tasks. We show that adding a novel adverserial loss and an effective batch sampling strategy can simultaneously minimize intra-class distance and maximize inter-class distance. The results we obtain are in sync with our theoretical establishments.

References

1. Bendale, A., Boult., T.E.: Towards open set deep networks., pp. 15–29. Springer Berlin Heidelberg (2010)
2. E. M. Rudd, L. P. Jain, W.J.S., Boult., T.E.: The extreme value machine. CoRR abs/1409.0473 (2014)
3. H. K. Ekenel, L.S.T., Stiefelhagen., R.: Openset face recognition-based visitor interface system. Neural Comput. 24(8), 2151–2184 (Aug 2012)
4. J.-C. Chen, V.M.P., Chellappa., R.: Unconstrained face verification using deep cnn features. In Winter Conference on Applications of Computer Vision (WACV) abs/1411.2539 (2014)
5. J. Stallkamp, H.K.E., Stiefelhagen, R.: Video-based face recognition on real-world data. In: Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A. (eds.) Advances in Neural Information Processing Systems 23, pp. 1243–1251. Curran Associates, Inc. (2010)
6. L. Best-Rowden, H. Han, C.O.B.F.K., Jain, A.K.: Unconstrained face recognition: Identifying a person of interest from a media collection. In: Xing, E.P., Jebara, T. (eds.) Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 32, pp. 595–603. PMLR, Bejing, China (22–24 Jun 2014)
7. Li, F., Wechsler, H.: Open set face recognition using transduction. In: Advances in Neural Information Processing Systems 27, pp. 1808–1816 (2014)
8. O. M. Parkhi, A.V., Zisserman., A.: Deep face recognition. CoRR abs/1411.2539 (2014)
9. P. J. Phillips, P.G., Micheals., R.: Handbook of face recognition, chapter evaluation methods in face recognition. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2. pp. 3104–3112. NIPS'14, MIT Press (2014)
10. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei., L.: Imagenet large scale visual recognition challenge. In: Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A. (eds.) Advances in Neural Information Processing Systems 23, pp. 1243–1251. Curran Associates, Inc. (2010)
11. S. Liao, Z. Lei, D.Y., Li., S.Z.: A benchmark study of large-scale unconstrained face recognition. In: Advances in Neural Information Processing Systems 27, pp. 2204–2212 (2014)
12. W. J. Scheirer, A. de Rezende Rocha, A.S., Boult., T.E.: Image description using visual dependency representations. In: Toward open set recognition. pp. 1292–1302. ACL (2013)
13. W. J. Scheirer, L.P.J., Boult., T.E.: Probability models for open set recognition. In: Proceedings of the 13th Conference of the European Chapter of the Association for

Computational Linguistics. pp. 747–756. EACL '12, Association for Computational Linguistics (2012)
14. Y. LeCun, C.C., Burges., C.J.C.: The mnist database of handwritten digits. In: Lafferty, J.D., Williams, C.K.I., Shawe-Taylor, J., Zemel, R.S., Culotta, A. (eds.) Advances in Neural Information Processing Systems 23, pp. 1243–1251. Curran Associates, Inc. (2010)
15. Y. Sun, D. Liang, X.W., Tang., X.: Deepid3: Face recognition with very deep neural networks. CoRR abs/1411.2539 (2014)
16. Y. Sun, Y. Chen, X.W., Tang., X.: Deep learning face representation by joint identification-verification. CoRR abs/1411.2539 (2014)
17. Y. Taigman, M. Yang, M.R., Wolf., L.: Deepface: Closing the gap to human-level performance in face verification. In Computer Vision and Pattern Recognition (CVPR) abs/1411.2539 (2014)
18. Yandong Wen, Kaipeng Zhang, Z.L., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: Xing, E.P., Jebara, T. (eds.) Proceedings of the 31st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 32, pp. 595–603. PMLR, Bejing, China (22–24 Jun 2014)