Low-shot Face Recognition with Hybrid Classifiers

Yue Wu†, Hongfu Liu†, Yun Fu†‡
†Department of Electrical & Computer Engineering, Northeastern University, Boston, MA, USA
‡College of Computer & Information Science, Northeastern University, Boston, MA, USA
{yuewu,yunfu}@ece.neu.edu, liu.hongf@husky.neu.edu

Abstract

In this paper, we present our solution to the MS-Celeb-1M Low-shot Face Recognition Challenge. This challenge aims to recognize 21,000 celebrities, in which 20,000 celebrities (Base Set) come with 50-100 images per person for training. But only one training image is provided for each person in the rest 1,000 celebrities (Novel Set). Given the dispersion in the number of training samples between Base Set and Novel Set, it is hard to build a single classifier that works well for both sets. To solve this problem, we propose a framework with multiple classifiers. This decomposes a single classifier for all data into multiple classifiers that each works well for a part of data. To be more specific, a Deep Convolution Neural Network (CNN) is utilized for Base Set and a Nearest Neighbor (NN) model is applied to Novel Set. The final prediction is based on a fusion of CNN results and NN results. Extensive experiments on MS-Celeb-1M Low-shot face dataset demonstrate the superiority of the proposed method. Our solution achieves 92.64% Coverage@Precision=0.99 in Novel Set while maintaining 99.58% top-1 accuracy in Base Set. This result wins the challenge in the track of without external data. Moreover, it is worth to note our result even surpasses some models using external data and can achieve the third place if compared with all participants.

1. Introduction

Convolutional Neural Networks(CNN) [10, 8, 11] has led a series of breakthroughs in large-scale image classification [17], object detection[16] and many other applications [12, 4]. Face recognition also has witnessed great success with CNN [19, 18]. However, in building a large-scale face recognizer, there might be very limited number of training samples for some people. The major challenge is how to build the classifier that can work well for people with both enough and limited samples. To study this problem, Guo *et al*. [5] introduced a benchmark dataset with 21,000 persons. This benchmark dataset is a subset of MS-Celeb-1M [6], in which there are total 99,891 people. These 21,000 people are divided into two parts, *i.e*., Base Set and Novel Set, regards of the number of available training samples. Base Set consists of 20,000 people, with an average of 58 training samples per person. Novel Set has the rest 1,000 people, of which each comes with 1, 2 or 5 training images.

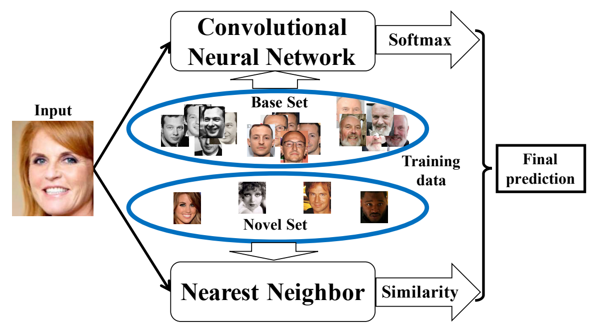

Figure 1. The framework for hybrid classifiers. Convolutional Neural Network and Nearest Neighbor models are utilized to classify Base Set and Novel Set, respectively. Final predictions are made based on the ensemble of the two models.

The shortage of training samples in Novel Set brings the difficulty to build a single Convolutional Neural Network (CNN) with a classifier on top for both Base Set and Novel Set. A CNN model works well when large amount of labeled images are available [7]. But poor generalization abilities are observed in these classes with few training samples [5]. Thus, the CNN model can be applied in Base Set with enough samples, but not Novel Set with few images.

In the Nearest Neighbor (NN) classifier [2, 3], samples are classified based on the class of their nearest neighbor. Compared with CNN, NN is one of the simplest classification methods yet effective. We argue that NN fits well in the classification problem of Novel Set under the low-shot learning scenario for two reasons. First, NN is a non-parametric method [14] that is robust with few training samples. Second, given the limited samples, the computation cost for NN is low. However, parameters of an NN model grows with the amount of training data [14]. Given the large amount training images in Base Set, NN will be slow and inaccurate compared with CNN.

Based on that CNN works well for Base Set and NN fits Novel Set, in this paper, we propose a framework with **hybrid classifiers** to ensemble different inferences from a CNN model and an NN model. Instead of using a single classifier for 21,000 celebrities, we decompose the problem into two sub-problems, recognizing 20,000 people in Base Set using CNN and recognizing 1,000 people in Novel Set using NN. By doing this, we utilize the advantages that CNN can achieve high accuracy with massive data while NN could handle the problem of few training samples.

To merge the two different kinds of classifiers into a unified framework, a fusion strategy is introduced. This strategy utilizes the property of high accuracy in CNN when a high top-1 confidence is achieved. Thus, we set a high threshold θ for CNN and the prediction is made if the top-1 confidence score is greater than or equal to θ. Otherwise, we utilize the output score in NN with 1,000 people in Novel Set. However, since the NN model only has 1,000 classes labels. The predictions made by NN are constrained into these 1,000 classes, which decreases the recognition accuracy in Base Set. To increase the accuracy in Base Set when making predictions using NN, we set another threshold β to filter out these samples that has lower confidence score in NN and assigned these samples with previous CNN predicted labels. The threshold θ and β are set manually based on the statical information on the validation set.

In summary, our contributions are summarized as:

- We propose a framework with hybrid classifiers using CNN and NN for low shot face recognition.
- We introduce a strategy to fuse two different kinds of classifiers into a unified classifier.
- Extensive experiments on MS-Celeb-1M Low-shot dataset demonstrate the superiority of the proposed method.

In MS-Celeb-1M Low-shot Learning Challenge: Know You at One Glance, our solution achieves state-of-the-art with 92.64% Coverage@Precision=0.99 in Novel Set while maintains 99.58% top-1 accuracy in Base Set. This result **wins** in the track of **without external data**. Moreover, it is worth to note our result even surpasses some models using external data and can achieve the third place if compared with all participants.

2. The Proposed Approach

In this section, we first introduce the Convolutional Neural Network(CNN) model, then show the Nearest Neighbor (NN) model. Last, hybrid classifiers based on CNN and NN are given.

2.1. Convolutional Neural Network (CNN)

The Convolutional Neural Network (CNN) model we used is a ResNet 34-layer model [8]. The network structure is the same with original model except the last fully-connected layer with softmax. The number of nodes in the last fully-connected layer are set to the number of classes.

Given an input image I with label y, the deeply learned feature $\mathbf{x} \in \mathbb{R}^d$ of the network can be calculated as:

$$\mathbf{x} = \phi(I), \tag{1}$$

where $\phi(\cdot)$ is the forward operation from the input layer to last average pooling layer in the ResNet 34-layer model. d is the dimension of $\mathbf{x}$ and is equal to the number of output in the last average pooling layer. After feature extraction, the fully-connected layer with softmax function is on top to generate the probability distribution for all classes. The output of the softmax is denoted as following:

$$\mathbf{p} = f(\mathbf{x}) = f(\phi(I)), \tag{2}$$

where $f(\cdot)$ representation the operation in the last fully connected layer with softmax function. The $f(\cdot)$ can also be viewed as a classifier for all classes given the feature $\phi(I)$. Cross entropy loss is utilized to train the network from scratch and end-to-end:

$$-\mathbf{y} \log \mathbf{p}, \tag{3}$$

where $\mathbf{y}$ is the one-hot encoding of y.

After training, we could utilize the classifier $f(\phi(I))$ to infer if an input image I belongs to training classes, as well as the deeply learned feature extraction $\phi(\cdot)$ as general face features.

During inference, the top-1 output of the network can be formulated as:

$$s_c, y_c = \max(\mathbf{p}). \tag{4}$$

The $\max$ function will return the top-1 score s_c and corresponding label y_c in the softmax output $\mathbf{p}$.

2.2. Nearest Neighbor (NN) Classifier

In the Nearest Neighbor (NN) classifier, we first extract deep features $\mathbf{x} = \phi(I)$ for every image I learned from the CNN model as general face features. L2 normalization is applied to every feature before calculating the similarity between every pair of images. Given an annotated image set S and a test image $\hat{I}$, we first selected the nearest sample (I^*, y^*) from all samples in the set:

$$\begin{aligned}(I^*, y^*) &= \underset{(I,y)\in S}{\operatorname{argmin}}(||\frac{\phi(\hat{I})}{||\phi(\hat{I})||_2} - \frac{\phi(I)}{||\phi(I)||_2}||_2) \\ &= \underset{(I,y)\in S}{\operatorname{argmin}}(||\frac{\hat{\mathbf{x}}}{||\hat{\mathbf{x}}||_2} - \frac{\mathbf{x}^*}{||\mathbf{x}^*||_2}||_2).\end{aligned} \tag{5}$$

Then, the prediction is made by assigning y^* as the label of the NN classifier for the new image $\hat{I}$:

$$y_n = y^*. \tag{6}$$

To measure the confidence of the NN classifier, we first calculate the vector difference of two normalized features between the new image $\hat{I}$ and the found nearest neighbor I^*, which can be formulated as:

$$\mathbf{z} = \frac{\hat{\mathbf{x}}}{||\hat{\mathbf{x}}||_2} - \frac{\mathbf{x}^*}{||\mathbf{x}^*||_2}. \tag{7}$$

The confidence score can be computed by:

$$s_n = 1 - \frac{\mathbf{z}^{\mathrm{T}} \cdot \mathbf{z}}{2}. \tag{8}$$

By doing this, score s_n will be close to 1 if $\hat{I}$ and I^* belong to the same person, otherwise s_n will be close to $1 - \frac{\sqrt{2}}{2}$, since all features are positive due to the relu activation before the last averaging pooling layer.

2.3. Hybrid Classifiers

To build a hybrid classifier with CNN and NN, two thresholds $\theta (0 \leq \theta \leq 1)$ and $\beta (0 \leq \beta \leq 1)$ are set to switch the output result between CNN and NN. Parameter θ is used to select the confident output in the CNN model, meanwhile β is utilized to select the confident output in the NN model. These parameters are set through the static information on the validation set. The algorithm is shown in Algorithm 1. If the CNN output s_c is greater than or equal to θ, we directly use the output from CNN (s_c, y_c) as the final prediction. If s_c is lower than θ, the output from NN (s_n, y_n) will be taken as the final result when s_n is greater than or equal to β or s_n is greater than or equal to s_c . Otherwise, the confidence score from NN (s_n) and the predict label from CNN (y_c) will be combined as (s_n, y_c) that will be the final output. How to set parameters θ and β will be discussed in Section 3.4.

3. Experiments

In this section, we introduce the datasets and evaluation metric in the beginning. Analysis about every single model and hybrid classifiers are given later, followed by results and analysis on validation sets and test sets.

3.1. Datasets and Evaluation Metric

We evaluate our method on MS-Celeb-1M low shot face dataset [5] that has 21,000 classes. All models are trained on provided training data and evaluate on the validation set. Details of training data are shown in Table 1. The validation set, shown in Table 2, has total 25,000 images, of which 20,000 images are in Base Set and 5,000 images are in Novel Set.

Algorithm 1 Hybrid classifier

procedure PREDICT(CNN score s_c, CNN label y_c, NN-Novel score s_n, CNN label y_n, θ and β)
  if $s_c \geq \theta$ **then**
    Output: (s_c, y_c)
  else
    if $s_n \geq \beta$ or $s_n \geq s_c$ **then**
      Output: (s_n, y_n)
    else
      Output: (s_n, y_c)

Table 1. Training Set Information

Set	Classes	Images	AVG Images
Base	20,000	1,155,175	58
Novel	1,000	1,000	1
Total	21,000	1,156,175	55

Table 2. Validation Set Information

Set	Classes	Images	AVG Images
Base	20,000	20,000	1
Novel	1,000	5,000	5

We also obtain a final result on 100,000 test images, reported by challenge organizers. Aligned data are used in all experiments. We do not apply any face detection and alignment methods.

Performance are evaluated on both Base Set and Novel Set. On Base Set, the top-1 accuracy is required to be greater than 98%, which means you can not sacrifice too much performance on Base Set to get better results on Novel Set. On Novel Set, recognition coverage at precision 99% is utilized to evaluate the performance. Coverage and Precision are defined as:

$$\text{precision} = j/m, \tag{9}$$
$$\text{coverage} = m/k, \tag{10}$$

where j images are correct in the recognized m images and k is the total number of images in the measurement set.

3.2. Convolutional Neural Network (CNN)

We first investigate the single Convolutional Neural Network (CNN) model with all classes. The node number of the last fully connected layer is set to 21,000, which is the same with the total number of classes in Base Set and Novel Set. We start with this model to see how the CNN model works on both Base Set and Novel Set. To alleviate the unbalanced data problem, we employ a simple up-sampling strategy for data in Novel Set. Each image in Novel Set is up-sampled 30 times by copy-paste.

Before feeding images into the ResNet 34-layer model, we apply the inception-like data augmentation method [20] to each image. The batch size is set to 256. Total training epochs are 90. The learning rate is initially set to 0.1

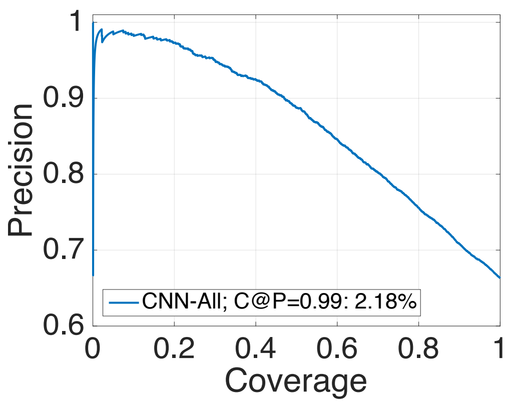

Figure 2. Precision-Coverage curve on Validation Novel Set using a single CNN model trained with all training data (CNN-All).

Table 3. Top-1 accuracy on Validation Set using a single CNN model trained with all training data.

	Base Set	Novel Set
CNN-All	99.92%	66.3%

and decreases to 1/10 of the previous learning rate every 30 epochs. The model is trained from scratch with two Pascal Titan X GPUs and takes about 50 hours to finish. The toolbox we used is tensorpack[1] on tensorflow [1].

After training, we directly test the classifier with softmax output. The confidence score and predicted label in Eq. (4) are taken. For testing, one single center crop from the original image is used as input.

Results of top-1 accuracy on validation are shown in Table 3. We can see the top-1 accuracy on Base Set reaches 99.92%, which is very promising. However, compared with Base Set, the top-1 accuracy on Novel Set is only 66.3%. There is a big gap on top-1 accuracy between Base Set and Novel Set due to the difference in number of training samples. Moreover, the Precision-Coverage curve on the Novel Set is shown in Figure 2. We can observe that the coverage@precision=0.99 can only get 2.18%, which also demonstrates that the CNN has a bad performance on Novel Set compared with Base Set.

3.3. Nearest Neighbor (NN)

To investigate the performance of Nearest Neighbor classifier on Base Set and Novel Set, we conduct two experiments. We first build a Nearest Neighbor classifier with all samples (NN-All) in Base Set and Novel Set. The second is a Nearest Neighbor classifier on means of all classes (NN-Mean) [13]. The implementation here is based on scikit-learn package [15]. Top-1 accuracy on validation set is shown in Table 4. From the result, we can see that the NN-All and NN-Mean are worse than CNN on Base Set. NN-All has similar performance with CNN on Novel Set. NN-Mean is better than NN-All and CNN with an improvement

[1] https://github.com/ppwwyyxx/tensorpack

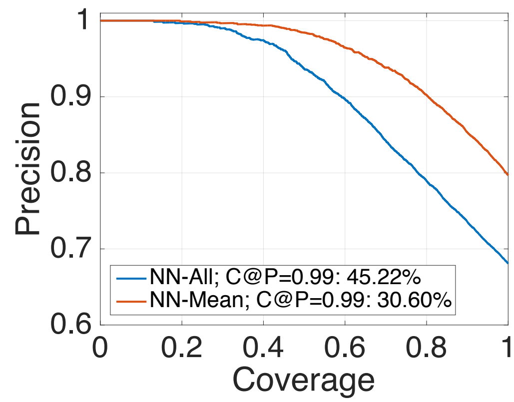

Figure 3. Precision-Coverage curve with NN-All and NN-Mean models using all training data. NN-All takes all training samples to build the Nearest Neighbor classifier. NN-Mean is a Nearest Neighbor classifier that is built on means of every classes in training data.

Table 4. Top-1 accuracy on validation set with NN-All and NN-Mean models using all training data.

	Base Set	Novel Set
NN-All	99.73%	66.08%
NN-Mean	99.73%	79.68%

of 13% on top-1 accuracy. The precision-coverage curves of NN-All and NN-Mean on Novel Set are shown in Figure 3. From the figure, we can see that NN-Mean is better than NN-All on Novel Set at the coverage@Precesion=0.99, which is consistent with top-1 accuracy in Table 4. Compared with CNN shown in Figure 2, the NN-All improves the coverage@Precesion=0.99 from 2.18% to 30.60%. Further, NN-Mean can achieve 45.22%. We can see that given the low-shot learning problem, NN is better than CNN on Novel Set while CNN is better than NN on Base Set.

3.4. Hybrid Classifier and Parameter Analysis

Thus, we can conclude that the performance is not satisfied if we only use either CNN or NN. In this subsection, we investigate each component in hybrid classifiers and how to ensemble CNN and NN.

3.4.1 CNN with Base Set

We did another set of experiments that the CNN model is trained on Base Set, that covers 20,000 Base Set classes. Data on Novel Set are not used. During testing, all samples in validation set are tested, which means that predictions on Novel Set are all wrong. However, we can still get a prediction for samples in Novel Set from the 20,000 classes with a confidence score.

We show the coverage-threshold curve in Figure 4, in which coverage at a threshold is defined as the percentage of samples that have higher confidence scores than the threshold. From the figure, we can see that a high coverage can

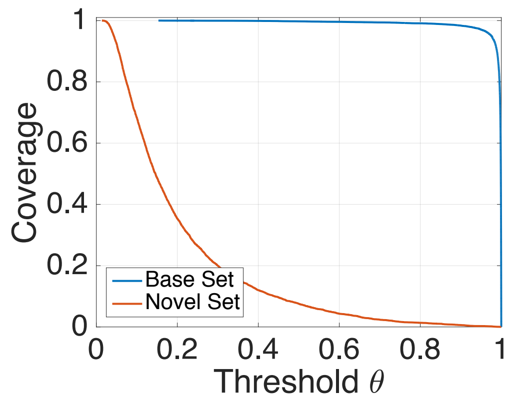

Figure 4. Coverage-threshold θ curve on Validation Base Set and Validation Novel Set using a single CNN model trained with data only in Base Set.

Table 5. Statical information on Validation Base Set and Validation Novel Set using a single CNN model under different threshold θ. The CNN model is trained with data only in Base Set.

	Novel Set		Base Set	
θ	Precision	Coverage	Precision	Coverage
0.80	0.0%	1.34%	99.99%	99.10%
0.85	0.0%	1.02%	100.00%	98.85%
0.90	0.0%	0.62%	100.00%	98.37%
0.95	0.0%	0.28%	100.00%	97.05%

be achieved even with a high threshold in Base Set. However, in Novel Set, the coverage is extremely lower than Base Set when the threshold is high. Some precision coverage number with different thresholds are shown in Table 5. When $\theta = 0.95$, the coverage on Novel Set is only 0.28%. But we can get 100.00% accuracy with 97.05% coverage on Base Set. This also demonstrates the good performance with CNN on Base Set. And the coverage results show that we could use a high threshold θ to filter out most Novel set samples given the low coverage.

3.4.2 NN with Novel Set

Next we check the performance with an NN classifier that is built only on Novel Set (NN-Novel). The deep features we used here are from the CNN model trained on Base Set. The result on the Novel Set in validation is shown in Figure 6. From the figure, we can see that NN-Novel classifier could achieve 91.14% coverage given the 99% precision. Since NN-Novel classifier does not cover the classes in Base Set. The accuracy of NN-Novel on Base Set is 0%. We further show the different coverage of NN-Novel on Base Set and Novel Set in Figure 5. Similar to the analysis in the section 3.4.1, when $\beta = 0.5$, the coverage on Base Set is only 7.90%. But we can get 98.81% accuracy with 92.48% coverage on Novel Set. Thus, we can find a large part of samples from Base Set if the scores from NN is lower than 0.5.

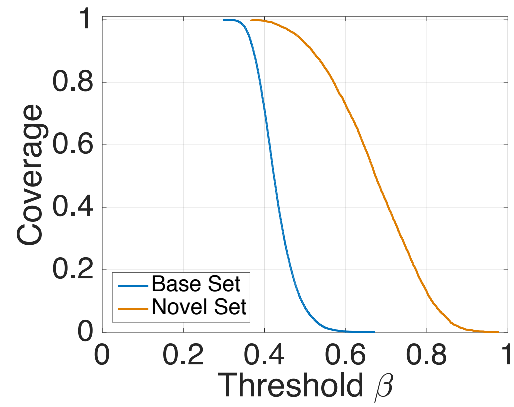

Figure 5. Coverage-threshold β curve of Validation Base Set and Validation Novel Set using a single NN model trained with training data only in Novel Set (NN-Novel).

Table 6. Statical information on Validation Base Set and Validation Novel Set using a single NN model under different threshold β. The NN model is trained with training data only in Novel Set.

	Novel Set		Base Set	
β	Precision	Coverage	Precision	Coverage
0.4	96.78%	99.62%	0.0%	71.24%
0.5	98.81%	92.48%	0.0%	7.90%
0.6	99.86%	72.72%	0.0%	0.17%

3.4.3 Hybrid Classifier

Given the observations in last two sections, we set $\theta = 0.95$ and $\beta = 0.5$ in Algorithm 1. The CNN model used here is the model trained with only Base Set. And the NN model is trained with only Novel Set.

Results of top-1 accuracy on validation are shown in Table 7. We can see the top-1 accuracy on Base Set can keep 99.63% compared with 99.92% in the CNN with Base Set (Table 3). And top-1 accuracy on Novel Set achieves 95.68%, which are much better than any single model in Section 3.2 and 3.3. Moreover, the Precision-Coverage curve on the Novel Set is shown in Figure 7. We

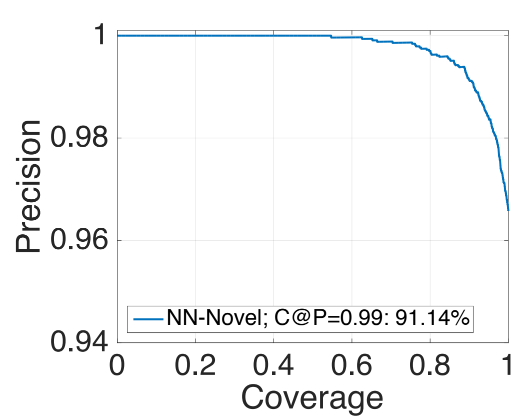

Figure 6. Precision-Coverage curve on Validation Novel Set using the NN model trained with data only in Novel Set (NN-Novel).

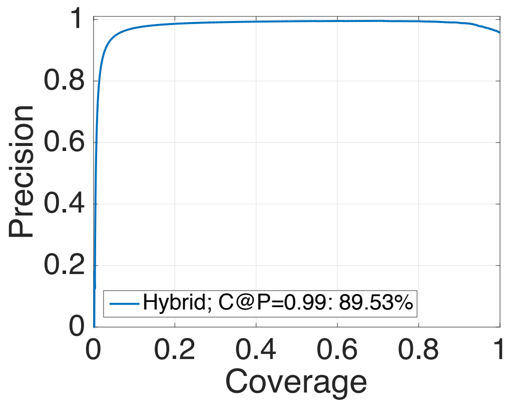

Figure 7. Precision-coverage curve on Validation Novel Set with a hybrid classifier. The hybrid classifier consists of a CNN model trained with data in Base Set and an NN model trained with data in Novel Set.

Table 7. Top-1 accuracy on Validation Base and Novel Set with the hybrid classifier.

Set	Base	Novel
Top-1 Accuracy	99.63%	95.68%

Table 8. Precision-coverage curve on Validation Novel Set with the new hybrid classifier (Hybrid-New) and the ensemble results (Ensemble).

Set	Base	Novel
Hybrid-New	99.62%	95.98%
Ensemble	99.68%	96.42%

can observe that the coverage@precision=0.99 can achieve 89.00%, which demonstrates that the hybrid classifier can perform well on both Base Set and Novel Set.

3.5. Further improvement

In this subsection, we illustrate several improvements based on the hybrid classifier, including replacement of CNN model for Base Set in section 3.4.1 with the CNN model for both Base and Novel Set in section 3.2 and using multi-crop testing.

From Figure 7, we can see the prevision-coverage curve starts from $(0, 0)$ and goes up with the increase of coverage in the beginning. The reason why it starts from $(0, 0)$ is because the CNN model in section 3.4.1 used in hybrid classifiers makes mistake even the threshold is set to a very high value 0.95. Thus we replace the CNN model for Base Set in section 3.4.1 with the CNN model for both Base and Novel Set in section 3.2. This model is denoted as Hybrid-New and final results on Novel Set are shown in Figure 8 and Table 8. Hybrid-New achieves 91.32% coverage@precision=0.99 compared with 89.53% in the original hybrid classifier.

Multi-crop testing [9, 21] has been demonstrated that can boost the performance of Convolutional Neural Networks. For the CNN model, predictions are based on an average of outputs from original and flipped images. For

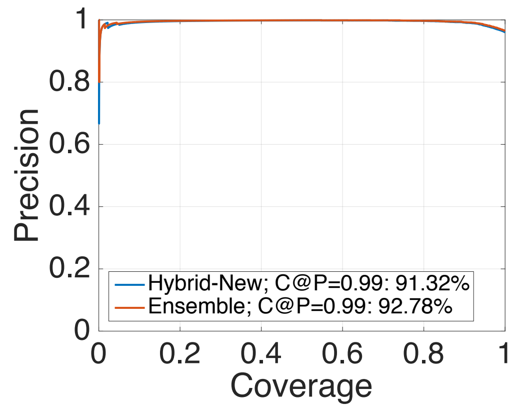

Figure 8. Precision-coverage curve on Validation Novel Set with the new hybrid classifier (Hybrid-New) and the ensemble results (Ensemble). Hybrid-New includes a CNN model training with all training data and an NN model trained with data in Novel Set. Ensemble is Hybrid-New with multi-crop testing strategy.

the NN model, we extract both features from original and flipped images and concatenate them to form the new feature. All other parts are kept the same as Hybrid-New. This model is denoted as Ensemble and results are shown in Figure 8 and Table 8. Model ensemble gets 92.78% coverage@precision=0.99 on Novel Set and 99.68% top-1 accuracy on Base Set.

3.6. Comparisons

In this section, we compare our approach with other algorithms, including other participants in the challenge and state-of-the-art methods [5, 7]. Results are shown in table 9. Our approach achieves the state-of-the-art without using external data. The coverage@precision=0.99 on Novel Set is 92.64%, which is 15% better than the second best UP [5]. It is worth to note our result even surpasses some models using external data (*e.g.* BUPTFR, Orion and FaceSecret), and can achieve the third place if compared with all results.

4. Conclusion

In this paper, we proposed a hybrid classifier model to solve the low-shot face recognition problem. The model decomposed a single classifier for all data into multiple classifiers that each worked well for a part of data. The good performance in MS-Celeb-1M low-shot face recognition challenge demonstrated the superiority of our method.

5. Acknowledge

We would like to acknowledge Yandong Guo and Zicheng Liu for helpful discussions in the comparison experiments after the challenge .

This research is supported by the ONR Young Investigator Award N00014-14-1-0484.

Table 9. Comparison on Test Set. As shown in the table, our method achieves the state-of-the-art without external data.

Team Name	External Data	Base Set, Top 1 Accuracy	Novel Set, C@P=0.99
NUS-Panasonic	Yes	99.74%	**99.01%**
Turtle	Yes	99.79%	97.61%
BUPTFR	Yes	99.08%	80.53%
Orion	Yes	99.90%	57.57 %
FaceSecret	Yes	97.98%	89.13 %
SIS	No	99.90%	73.86%
CNC240	No	89.72%	63.17%
KATE	No	97.83%	61.21%
SGM [7]	No	99.80%	27.23%
UP [5]	No	99.80%	77.48%
Ours	No	99.58%	**92.64%**

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] P. Cunningham and S. J. Delany. k-nearest neighbour classifiers. *Multiple Classifier Systems*, 34:1–17, 2007.

[3] Z. Ding and Y. Fu. Robust multi-view subspace learning through dual low-rank decompositions. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 1181–1187, 2016.

[4] Z. Ding, M. Shao, and Y. Fu. Deep robust encoder through locality preserving low-rank dictionary. In *European Conference on Computer Vision*, pages 567–582. Springer, 2016.

[5] Y. Guo and L. Zhang. One-shot face recognition by promoting underrepresented classes. *arXiv preprint arXiv:1707.05574*, 2017.

[6] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *European Conference on Computer Vision*. Springer, 2016.

[7] B. Hariharan and R. Girshick. Low-shot visual recognition by shrinking and hallucinating features. *arXiv preprint arXiv:1606.02819*, 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] A. G. Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[12] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[13] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2624–2637, 2013.

[14] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[16] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[18] 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.

[19] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1891–1898, 2014.

[20] 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.

[21] Y. Wu, J. Li, Y. Kong, and Y. Fu. Deep convolutional neural network with independent softmax for large scale face recognition. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 1063–1067. ACM, 2016.