Face Recognition - A One-Shot Learning Perspective

Sukalpa Chanda*^, Asish Chakrapani GV†^, Anders Brun‡, Anders Hast‡,
Umapada Pal† and David Doermann§
*Department of Information Technology, Østfold University College, Norway
sukalpa@ieee.org/sukalpa.chanda@hiof.no
† Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, India
asish.chakrapani@gmail.com, umapada@isical.ac.in
‡Centre for Image Analysis, Uppsala University, Sweden
{anders.brun, anders.hast}@it.uu.se
§Computer Science and Engineering, University at Buffalo, USA
doermann@buffalo.edu

***Abstract*—Ability to learn from a single instance is something unique to the human species and One-shot learning algorithms try to mimic this special capability. On the other hand, despite the fantastic performance of Deep Learning-based methods on various image classification problems, performance often depends having on a huge number of annotated training samples per class. This fact is certainly a hindrance in deploying deep neural network-based systems in many real-life applications like face recognition. Furthermore, an addition of a new class to the system will require the need to re-train the whole system from scratch. Nevertheless, the prowess of deep learned features could also not be ignored. This research aims to combine the best of deep learned features with a traditional One-Shot learning framework. Results obtained on 2 publicly available datasets are very encouraging achieving over 90% accuracy on 5-way One-Shot tasks, and 84% on 50-way One-Shot problems.**

***Keywords*-One-Shot Learning, Face recognition, Siamese Networks, Image Classification.**

I. Introduction

Face recognition has been extensively explored over the last several decades. Its value as a non-contact biometric authentication and in a wide variety of other digital applications like security, digital entertainment system, video analytics for marketing, video indexing from a streaming video cannot be ignored. Like any other image analysis problem, face recognition in its early days relied mainly on hand-crafted features like SIFT, SURF, Local Binary Pattern, Histogram of Gradient, Fisher vectors, but with the advent of deep-learning methodologies, there is a clear shift towards deep-learned features. During those early days, research was focused on improving the pre-processing stage, the introduction of local descriptors and feature transformation, but such techniques failed to counter the challenges of unconstrained face recognition. Hand-crafted feature-based methods were used to address changes in lighting, pose, and expression but failed in real life due to their inability to address more general pose challenges. This has changed as the deep-learning methods have evolved. Deep learning methods learn multiple levels of representations and abstractions by using a cascade of processing units for feature extraction and transformation. This leads to forming a hierarchy of abstraction/representation, and addresses changes in face pose, illumination, and expression. Even though deep-learning-based methods can tackle changes in lighting, pose, and expression while performing face recognition, one disadvantage is its demand for a huge amount of annotated data to train the system and the requirement of re-training when a new class is added. While transfer learning techniques can help mitigate such problems by freezing the first few layers and tuning pre-trained weights from the last few layers on the new data, it does not completely eradicate the problem.

One-shot algorithms, on the other hand, use a completely different philosophy for classification. One-shot algorithms are meant to perform classification seeing only a handful of the training samples. Thus a clever amalgamation of those two techniques could combine the best of both providing a rich feature representation using deep learning techniques and feeding those features to a One-Shot learning framework for classification. A widely spread strategy to implement One-Shot learning algorithms is to use a Siamese Neural Network with a triplet loss function. Our work takes a Siamese Neural Network-based approach to perform One-Shot learning and consequent classification. Deep Neural Network-based features from the "DLIB-ml machine learning toolkit" [1] are used for feature representation for all face images.

The primary contribution of this research is that a novel hybrid method combining a Siamese Neural Network with Res-Net encoded features for One-Shot face recognition task is being proposed. We also intend to publish our dataset with unconstrained face images procured from "Indian Movie Faces Database" in the near-future for One-Shot recognition task performance evaluation and benchmarking.

II. Related Work

Face detection and recognition methods have had significant importance as an image analysis research problem for almost

^ Equal contribution by the authors

978-1-7281-5686-6/19/$31.00 ©2019 IEEE
DOI 10.1109/SITIS.2019.00029

3 decades. One of the seminal articles in the early nineties is [2], where the authors represent faces using a small set 2-D Eigenvectors. Face recognition methods can be broadly divided into handcrafted features-based approaches and later deep-learning technologies deep-learned features-based approaches. The hand-crafted approaches focused mainly on high-dimensional artificial feature extraction and the reduction of features. The representative dimension reduction methods are the subspace learning methods like Principal Component Analysis [3], Linear Discriminant Analysis [4] and manifold learning methods like like Locality preserving projection [5]. With the advance of deep-learning, the representative method was to learn the discriminative face representations directly from the original image space. For example, Hu et al. [6] introduced us to the convolutional neural network applied to face recognition. It analyses the advantages and disadvantages of this method and shows the developmental roadmap in the future. This work is further explored and state-of-the-art results are obtained in [7], [8], [9], [10]. Albeit CNNs exceptional performance for some applications, such algorithms struggle to deal with many real-world applications that require learning or drawing inferences from small amounts of data, class imbalance and adjusting to a constant inflow of new class information. The problem of developing an efficient, robust face recognition system at scale is also not an exception in this context.

In the past few years, there have been several works that address this problem. To address the data imbalance problem Guo et al. [11] proposed a novel underrepresented classes promotion loss term which aligned the norms of weight vectors of underrepresented classes and normal classes thus giving the one-shot classes an equal weight-age. Work by Wang et al. [12] proposes a framework based on CNN, which deals with the deficient training data by using a balancing regularizer and shifting the center regeneration to regulate norms of weight vector into the same scale and adjusts clustering center. Insufficient training data and data imbalance, however, causes the network to perform poorly. Ding et al. [13] proposed an approach to solve the underrepresented class problem in one-shot learning, by focusing on building generative models to build extra examples. It proposed a generative model to synthesize data for one-shot classes by adapting the data variances and augmenting features from other normal classes. Another work by Jhadav et al. [14] proposed the method of deep attribute representation of faces for one-shot face recognition. They used specific attributes of human faces such as the shape of the face, hair, gender to fine-tune a deep CNN for face recognition. Their experimental results on standard datasets showed that deep attribute representations performed better in case of two one-shot face recognition techniques such as an exemplar SVM and one-shot similarity kernel. Wu et al. [15] proposed a framework with hybrid classifiers using a CNN and the nearest neighbor (NN) model. The work by Hong et al. [16] proposes a domain adaptation network to solve the One-shot task, the authors generated images in various poses using a 3D face model to train the deep model. Zhao et al. [17] proposed an enforced softmax that contains optimal dropout, selective attenuation, L2 normalization and model-level optimization which boosted the standard softmax function to produce a better representation for low-shot learning.

The concept of Siamese Networks was initially introduced by Bromley et al. [18] for the signature verification problem and further, the use of deep convolutional Siamese networks for one-shot tasks with a significant accuracy has been showcased in [19]. Face recognition usually consists of face detection, feature extraction, and recognition. We use the dlib-ml [1] toolkit which leverages image-driven neural networks to detect and extract the faces in a given image and then use a resnet based architecture to generate a feature vector to represent each face. In this paper, we propose a method which integrates the concept of Deep convolutional Siamese networks and a transfer learning strategy to produce a robust face recognition system which leverages the deep learned feature attributes.

III. Methodology

One-shot learning can be achieved in several ways. In this research we have explored two approaches: (a) Siamese Neural Network based approach; (b) a Deep-feature encoding approach followed by the nearest neighbor classification of those encoded features. We settled on a method by combining the two approaches. This improvised combined method uses the encoded features generated out of a ResNet CNN architecture as an input to the Siamese network, and the Siamese network is being trained to discriminate between two encoded feature vectors. In this combined approach a pre-trained Deep convolutional neural network (ResNet) acts as a feature extractor for a pair of an input image and then an energy function Θ is used which ties the twin networks to compute the similarity index. When the two encoded feature vectors for the input face images are obtained, the Siamese Network learns to score the similarity of those two encoded feature vectors in a range of 0-1. Where 1 is assigned if both the input images are of the same class.

A. Siamese Network

Siamese networks are a subset of deep neural network architectures that contain two identical sub-networks working in cohesion that use the same weights while taking two distinct input vectors and are joined by a comparative function. Such networks are used to determine the similarity between two distinct inputs. It is important that not only the architectures of the sub-networks are identical, but the weights are shared among them as well for the network to be called 'Siamese'. In this current study, the convolutional Siamese network is designed to learn features of the input images regardless of prior domain knowledge with very few samples from a given distribution. This model was also adopted because the twin networks share weights resulting in fewer parameters to train

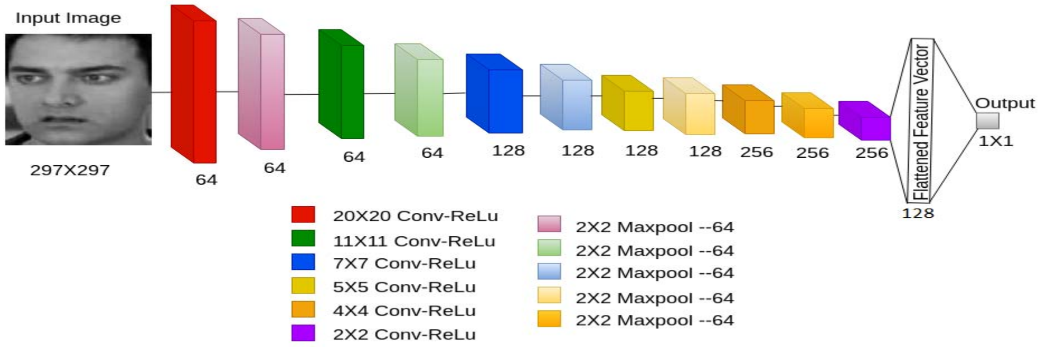

Fig. 1: Sibling of the Twin Siamese Network Architecture used in the experiment(twin network not depicted).

on and a lower tendency of over-fitting. For experiments, a small labeled support set consisting of train-validation classes and test classes were used. During training, the network takes a pair of images as the input where it learns to discriminate between two input images based on their class labels and features. The task is achieved by generating probability scores which aid in perceiving whether they belong to the same class or different classes. For evaluation of n way one-shot tasks, the network is provided with pairs of images consisting of a reference image and one sample image from each of the n unseen classes at each instance. The label from the pair with the highest probability is then given to the reference image. A pictorial diagram of our Siamese network is shown in Fig. 1.

1) Learning Details: A constant learning rate η_j is opted for all the layers whilst following a step-based decay method decaying at a uniform rate of 1% at every 500 iterations. The Validation accuracy metric is calculated after every 1000 iterations and the model with the best accuracy is saved during training. The model is trained for a maximum for 100,000 iterations. An early stopping condition was included in case the validation accuracy does not show improvement over 10,000 iterations. The momentum for each layer evolves with a predefined linear slope until it attains a final value of 0.9 and it is initialized with a value of 0.5 at the beginning. The model is trained with a batch size of 8, along with a linearly evolving layer-wise momentum μ_j for the jth layer, and L2 regularization penalization, weights for each iteration N. So the weight update rule for iteration N is:

$$\begin{aligned} W_{kj}^{N}(x_1^i,x_2^i) &= W_{kj}^{N} + \Delta W_{kj}^{N}(x_1^i,x_2^i) + 2\lambda_j |W_{kj}| \\ W_{kj}^{N}(x_1^i,x_2^i) &= -\eta_j \nabla W_{kj}^{N} + \mu_j \Delta W_{kj}^{N-1} \end{aligned} \tag{1}$$

where ΔW_{kj} is the partial derivative with respect to the weight between the j^{th} neuron in a given layer and the k^{th} neuron in the next layer.

2) Weights: The weight initialization in all the layers in the network is done using the Glorot uniform initializer. The initializer draws samples from the uniform distribution of $[-g,g]$where g is given by the equation

$$g = sqrt(\frac{6}{(fan_{in} + fan_{out})}) \tag{2}$$

Here, fan_{in} is the number of input units in the weight tensor and fan_{out} is the number of output units in the weight tensor [20]. The biases were initialized using the default setting of zeros in all the layers.

3) Loss function: The model error for the Siamese network during training is computed using a regularized cross entropy loss function. The cross-entropy function equation is as follows

$$\begin{aligned} L(x_1^i,x_2^i) = y(x_1^i,x_2^i) logP(x_1^i,x_2^i) \\ +(1-y(x_1^i,x_2^i)) log(1-P(x_1^i,x_2^i)) \\ +\lambda^N |W|^2 \end{aligned} \tag{3}$$

Here i denotes the i^{th} index of the current batch , $y(x_1^i,x_2^i)$ is a vector of length M consisting of labels. It is assumed that it equals 1 in case of same class and 0 in case of different class for iteration N.

B. ResNet

The ResNet architecture was developed to address some issues observed in its predecessor, the VGG-Net. One thing lacking in VGG-Net was it tends to lose generalization capability with an increase in the network depth. The other problem that ResNet deals with is countering the "vanishing gradient" issue which is often a problem with deeper networks. This is because gradients from the outer most layer easily shrink to zero after several applications of the chain rule, hence no weight updates are performed in the network. ResNet introduced the "skip connection" concept and by virtue of that gradients can flow directly backward from deeper layers to initial filters skipping intermediate layers. The Resnet used here is a pruned version of ResNet-34 [21].

In a pre-processing step, a CNN generates the bounding box information of a face along with a set of 68 face Landmark points [1] from an input image. The ResNet is fed with the bounding box information of the face and those set of 68 activations points inside the face region. In order to save time

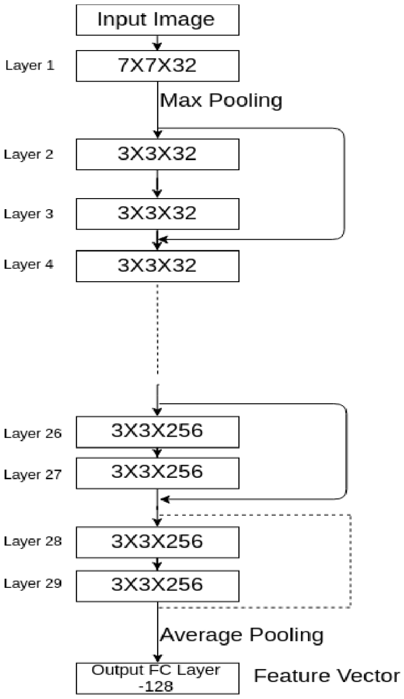

Fig. 2: Pruned ResNet Architecture used in the experiment.

and computational resources, we have used pre-trained weights from the initial layers of this network. Those weights were obtained while this network was trained from scratch on a dataset of about 3 million faces. At that time the training dataset was composed of 7845 individual face images procured from multiple sources such as the ”face scrub dataset, the VGG dataset and a large number of images scraped from the internet. This network in the 29th layer generates a 128-dimensional encoded feature for an input face image, and later that 128-dimensional encoded feature is being used for classification. This network learns the weights using a loss function called “Triplet Loss”. The pruned network architecture is shown in Fig. 2.

1) Loss function: In this current study, the ResNet architecture is uses a “Triplet Loss” function, governing by the following equation:

$$L = max(D(a, p) - D(a, n) + margin, 0)) \qquad (4)$$

The objective behind training this pruned ResNet is to generate optimal weights such that 128-dimensional feature embedding of an anchor image and positive image should be similar and feature embedding of anchor image and negative image should be much further apart. While using the “Triplet Loss” function to train the network, the 128-dimensional feature embedding from an anchor image is compared with the 128-dimensional feature embedding of both a positive sample and a negative sample. The objective here is to decrease dissimilarity between the anchor image and positive image and increase dissimilarity between the anchor image and the negative image. Here ’a’ denotes an anchor image, ’p’ denotes a positive image and ’n’ denotes a negative image. Another hyperparameter variable called margin is being added to the loss equation, that defines how far away the dissimilarities should be. For example, if the margin = 0.4 and d(a,p) = 0.3 then d(a,n) should at least be equal to 0.7.

C. A Combined Hybrid Approach

The proposed combined approach is depicted in Fig. 3. The Siamese network is taking as input the deep-learned encoded features those were generated by the pruned Res-Net CNN and learns its own set of weights intending to lower its cross-entropy loss function.To optimize the weights for our datasets, the weights of the initial convolutional layers were kept constant and the update weights are carried on the final few layers of the network with our training samples. Note that the Res-Net CNN is has its own set of weights and its corresponding loss function as well.

IV. Experimental Protocol

We used an N-way one-shot task performed on ’N’ “support classes” in a disjoint set each time for evaluating the performance in the evaluation set. For our experiments, we use 4 values of N pertaining to the set of 5,10,20,50. The efficacy of such algorithms is measured based on its performance on N-way tasks. During testing for a query sample image, a support class set S is provided consisting of ’n’ examples each from ’N’ different unseen classes. The algorithm then has to determine which of the support set classes the query sample belongs to. Two draws producing n samples each are taken, and each one of the samples produced in the first draw is taken as test images and compared against all samples of the second draw. This process was done twice for each evaluation set of n classes. We therefore perform 2N different one-shot tasks. We also observe the individual set accuracy and a mean global accuracy for the model has been reported.

A. Dataset

The experiments were conducted on two publicly available large-scale datasets: “Labeled Faces in the Wild”(LFW) [22] and “Indian Movie Face database” (IMFDB) [23]. Another popular dataset “MS-Celeb Low-Shot dataset” has not been included in the experiment for two reasons. First, some of the image samples of the dataset has been used to train the Res-Net based face recognition system, hence it would be unfair to use that database while evaluating the proposed system, Second, the dataset is unfortunately no longer publicly available. The reason for choosing “LFW” is that it is the most common dataset used for performance benchmarking of a face recognition system, and this dataset is a curated dataset with proper alignment and proper annotation. The “IMFDB” consists of unconstrained type images with much greater variability in terms of pose, illumination, and color. This variability is the reason for using IMFDB dataset in our experiments. The two datasets are complementary to each

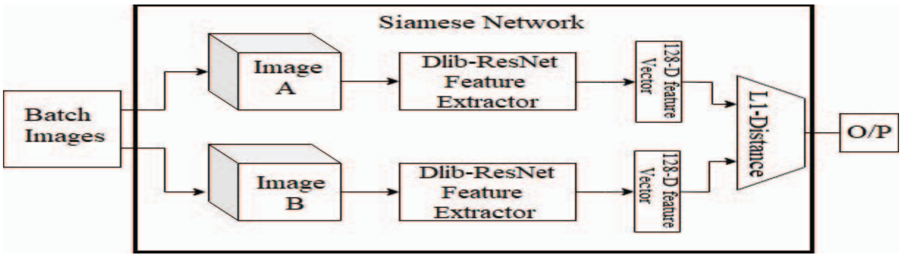

Fig. 3: Combined hybrid architecture used in the experiment.

other in that sense. The details of the respective datasets are given below.

1) LFW: - This database consists of 13,000 images of faces collected from the web. Each face has been labeled with the name of the person pictured. In this dataset, 1680 people have two or more distinct photos in the data set. To maintain the consistency and to ensure robustness we have various images for different facial positions. Further to accommodate slight variations such as facial hair and obstructions such as headgear, eyewear, we include a few samples of such images as well. Finally, for each class, we end up taking 15 images in total and due to this constraint, we remove all the classes which have 15 images or less. After this, we are left with a total of 96 classes. We use a deep funneling method to align the faces [24].

2) IMFDB: - This is a large unconstrained face database consisting of 34512 images of 100 Indian actors collected from more than 100 videos [23]. All the images are manually selected and cropped from the video frames resulting in a high degree of variability in terms of scale, pose, expression, illumination, age, resolution, occlusion, and makeup. Videos collected from the last two decades contain large diversity in age variations compared to the images collected from the Internet through a search query. IMFDB is the first face database that provides detailed annotation of every image in terms of age, pose, gender, expression and type of occlusion that may help others face-related applications. This dataset exhibits a huge degree of intra-class variability as well (Fig. 4).

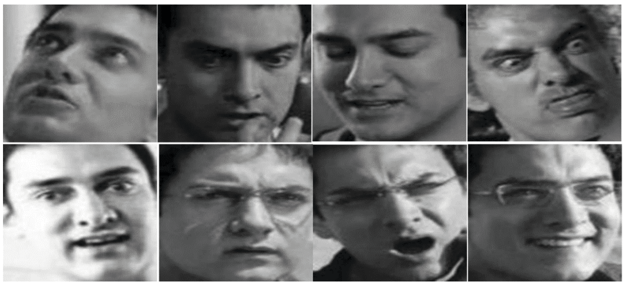

Fig. 4: Example of intra class variability in IMFDB dataset.

To maintain the variability and to ensure robustness we have various images with different facial positions. Further to accommodate slight variations such as facial hair and obstructions such as headgear, eyewear we include a few samples of such images as well. For each class, we considered 20 images in total. We have a total of 100 classes. To keep the train and test set completely disjoint and to exclude any overlap in the classes we removed 6 classes which were common to IMFDB and the dataset used to train the ResNet.

V. Results & Discussions

While conducting experiments with three different approaches, the input test and train set for each fold were same for all three experiments. This was done purposely to compare the efficacy of three approaches fairly.

For our experiments, the subset of the LFW database consisting of 96-face classes with 15 samples in each class was used. Those 96 classes were selected since the rest of the other classes have less than 15 samples. For the evaluation in face recognition we perform 3 different one-shot tasks i.e. 5, 10 and 20 way tasks so the new dataset was split into either 91-5/ 81-10 or 71-20 train-validation & evaluation classes, where the train set was further split according to an 80-20% split resulting in 72, 64 or 56 classes for training and 19, 17 or 15 classes for validation.

The set of IMFDB consisting of 94-face classes with 20 samples in each class was used. For the evaluation in face recognition we perform 5, 10 and 20 way tasks so the new dataset was split into either 89-5/ 84-10 or 74-20 train-validation & evaluation classes, where the train set was further split according to an 80-20% split resulting in 71, 67 or 59 classes for training and 18, 17 or 15 classes for validation. The evaluation was conducted using the same n-way one-shot tests on the n classes from the evaluation set.

Both the datasets contain around $\simeq$95 classes and for training and evaluation, we use a fold wise method. So the total number of folds for "n-way" is obtained as total number of classes divided by "n" the number of classes for testing with minimal re-sampling. Therefore, in the case of 5-way we get 19 folds, 10-way we get 9 folds and for 20-way we get 4 folds. Note that to frame a 50-way one-shot task, given the number of classes in each of those two datasets we could perform only two folds of train-test evaluation run where a few of the classes might be re-sampled from the previous folds. By "fold" we mean to say an unique "train-validation-test" evaluation set.The accuracy metric used here is true recognition rate for each fold in a given dataset.

A. Siamese Network-based Results

Out of three approaches, during our initial experiments, the Siamese Network-based approach performed the worst. Even while dealing with a 5-way One-Shot recognition task, it could only deliver the highest accuracy of $\approx 32.50\%$ for both datasets. To give an idea, results obtained on n-way One-Shot tasks on both datasets on 4 different folds are shown in Table I and II. Since the results are not encouraging we are not providing results for all folds with respect to different n-way tasks.

TABLE I: Accuracy of One shot Tasks on LFW dataset using Siamese Network with own feature extractor

Fold Number	5-Way Task	10-Way Task	20-Way Task
Fold 1	32.50%	28.20%	23.40%
Fold 2	27.50%	26.70%	22.60%
Fold 3	30.00%	30.20%	25.60%
Fold 4	24.60%	24.80%	22.60%

TABLE II: Accuracy of One shot Tasks on IMFDB using Siamese Network with own feature extractor

Fold Number	5-Way Task	10-Way Task	20-Way Task
Fold 1	32.80%	30.80%	24.20%
Fold 2	30.50%	27.50%	20.60%
Fold 3	28.50%	28.60%	22.80%
Fold 4	27.60%	27.60%	26.02%

B. ResNet-Based Face Recognizer Results

The ResNet architecture for face Recognition from “DLIB” has been used in our experiment. To save time and resources, a transfer learning strategy was adopted. Here a pre-trained model of the Res-Net, which was generated while training 3 million face images was initially considered in this experiment. The weights of the initial convolutional layers of this model were kept constant during training on samples from the “LFW” and “IMFDB” and weights associated with all fully connected layers were updated. The 128-dimensional feature encoding obtained from the 29th layer of an input test image is compared with 128-dimensional feature encoding vectors of all support set samples, then the class of input image is assigned to the class of nearest neighbor amongst support set samples. Results on 5-way, 10-way and 20-way One-Shot learning tasks on LFW and IMFDB dataset is depicted in Table IV and Table III respectively. Note that here also we are reporting on the same 4 folds of data that we have reported for Siamese Network. It can be noted that with the use of ResNet feature encoding there is a striking improvement in the results compared to results obtained with the Siamese Network only based approach. The accuracy is as high as 87.00% with the 20-way One-Shot tasks on LFW dataset, whereas the highest accuracy on the same dataset with Siamese Network is $\approx 26.00\%$. A similar trend can be observed in the case of IMFDB dataset as well.

TABLE III: Accuracy of One shot Tasks on IMFDB using Dlib-ResNet-29 network

Fold Number	5-Way Task	10-Way Task	20-Way Task
Fold 1	80.80%	78.60%	80.00%
Fold 2	82.40%	80.40%	76.50%
Fold 3	81.00%	79.30%	78.20%
Fold 4	83.60%	82.00%	75.40%

TABLE IV: Accuracy of One shot Tasks on LFW dataset using Dlib-ResNet-29 network

Fold Number	5-Way Task	10-Way Task	20-Way Task
Fold 1	88.20%	86.00%	85.30%
Fold 2	90.00%	84.60%	87.00%
Fold 3	89.00%	90.00%	82.00%
Fold 4	90.20%	89.00%	81.40%

C. Results obtained from Combined Hybrid Approach

The classification technique that we used to perform One-Shot learning on the encoded features from Res-Net was a naive Nearest Neighbour classification. Despite the simple classification, such high accuracies from the ResNet-based approach confirm that the encoded features generated by the ResNet were very discriminative. This motivated us to couple the discriminative feature extractor with the sophisticated discriminator function of the Siamese network architecture. In this setup, the ResNet generated encoded features were fed to the Siamese network which learns its own set of weights and hence gives much higher accuracy in the range of 80.00%-84.20% even for the 50-way one-shot task. We experimented exhaustively with this approach with all possible folds of data. The 20-way one shot results are depicted in Table V and Table VI depicts the typical results obtained by this method on 50-way one-shot learning for the two datasets.

TABLE V: Accuracy of 20-way One shot Tasks on LFW Dataset & IMFDB using combined approach

Fold Number	LFW	IMFDB
Fold 1	92.50%	70.00%
Fold 2	95.50%	72.50%
Fold 3	82.50%	72.50%
Fold 4	87.50%	80.50%

In our experiments, for the **5-way** one shot task we obtained an average accuracy of 92.44% across 19 folds on the entire subset of LFW dataset. Further, we obtained accuracy as high as 97.00% in few ocassions. The mean accuracy yielded by the **10-way** one shot tasks over a 9 fold cross-validation set

TABLE VI: Accuracy of 50-way One shot Tasks on LFW Dataset & IMFDB using combined approach

Fold Number	LFW	IMFDB
Fold 1	80.00%	80.50%
Fold 2	82.50%	84.20%

was 90.55% with best accuracy shooting as high as 97.50% in one of the fold.

Similar to the experiments conducted on the LFW dataset we also performed 5-way and 10-way tasks on the IMFDB dataset. The mean accuracy of the 5-way one-shot task for 19 folds was observed to be 82.63%. Whereas for the 10-way one-shot task the mean accuracy across 9 fold set was observed to be 79.05%. The best accuracy of the 5 and 10 way task was observed to be 92.50% and 87.50% respectively.

D. Comparison with other techniques

Though there are a large number of published results on face recognition, however, very few works like [13], [16], [14] focus on the One-Shot face recognition task. Unfortunately, we could compare the performance of our system with only [14] as the others have used the "MS-Celeb Low Shot" dataset meant for One-Shot recognition task and that dataset is not available from any legitimate source. In [14], the authors did experiments for One-Shot recognition using the "LFW" dataset and we have compared our results with them in Table VII. Note that our method has outperformed the method proposed in [14] especially in the case of 10-way and 20-way one-shot tasks. We plan to preserve and publish the train and test split of images that we have used for our experiments from the other dataset "IMFDB", for benchmarking performance evaluation of One-Shot face Recognition task.

TABLE VII: Accuracy comparison of One shot Tasks on LFW

Method	5 Way	10 Way	20 Way
Deep attribute, Jadhav at al. [14]	94.00%	93.75%	88.87%
Dlib-Siamese Net , Proposed Method	**97.00%**	**97.50%**	**95.50%**

VI. Conclusions & Future Work

This article proposes a new hybrid approach of fusing Res-Net features along with a Siamese-Network classifier to handle face recognition task in a One-Shot learning framework. The proposed hybrid network shows impressive performance even while dealing with 50-way One-Shot recognition tasks on two publicly available datasets. Future research plan is to use more sophisticated discriminator function to combat 100-way One-Shot recognition task.

References

[1] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 07 2009.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[3] W. Zhao, R. Chellappa, and A. Krishnaswamy, "Discriminant analysis of principal components for face recognition," *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 336–341, 1998.

[4] L.-F. Chen, H.-y. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, "New lda-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, pp. 1713–1726, 10 2000.

[5] Y. C. Tan, Y. Zhao, and X. Ma, "Contourlet-based feature extraction with lpp for face recognition," *2011 International Conference on Multimedia and Signal Processing*, vol. 1, pp. 122–125, 2011.

[6] G. Hu, Y. Yang, D. Yi, J. Kittler, W. Christmas, S. Li, and T. Hospedales, "When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition," 12 2015, pp. 384–392.

[7] C. Lu and X. Tang, "Surpassing human-level face verification performance on LFW with gaussianface," *CoRR*, vol. abs/1404.3840, 2014.

[8] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," *CoRR*, vol. abs/1406.4773, 2014.

[9] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," *CoRR*, vol. abs/1412.1265, 2014.

[10] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," 09 2014.

[11] Y. Guo and L. Zhang, "One-shot face recognition by promoting underrepresented classes," *CoRR*, vol. abs/1707.05574, 2017.

[12] L. Wang, Y. Li, and S. Wang, "Feature learning for one-shot face recognition," *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2386–2390, 2018.

[13] Z. Ding, Y. Guo, L. Zhang, and Y. Fu, "One-shot face recognition via generative learning," 05 2018, pp. 1–7.

[14] A. Jadhav, V. P. Namboodiri, and K. S. Venkatesh, "Deep attributes for one-shot face recognition," in *ECCV Workshops*, 2016.

[15] Y. Wu, H. Liu, and Y. Fu, "Low-shot face recognition with hybrid classifiers," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[16] S. Hong, W. Im, J. Ryu, and H. S. Yang, "SSPP-DAN: deep domain adaptation network for face recognition with single sample per person," *CoRR*, vol. abs/1702.04069, 2017.

[17] J. Zhao, Y. Cheng, Z. Wang, Y. Xu, J. Karlekar, S. Shen, and J. Feng, "Know you at one glance: A compact vector representation for low-shot learning," 09 2017.

[18] J. Bromley, I. Guyon, Y. LeCun *et al.*, "Signature Verification using a "Siamese" Time Delay Neural Network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 04, p. 669688, 1993.

[19] G. Koch, R. Zemel, and R. Salakhudtdinov, "Siamese Neural Networks for One-shot Image Recognition," in *Proceedings of the 32 nd International Conference on Machine Learning*, vol. 37, Lille, France, Jul. 2015.

[20] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, 2010, pp. 249–256.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[22] 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," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[23] S. Setty, M. Husain, P. Beham, J. Gudavalli, M. Kandasamy, R. Vaddi, V. Hemadri, J. C. Karure, R. Raju, B. Rajan, V. Kumar, and C. V. Jawahar, "Indian Movie Face Database: A Benchmark for Face Recognition Under Wide Variations," in *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, Dec 2013.

[24] G. B. Huang, M. A. Mattar, H. Lee, and E. Learned-Miller, "Learning to align from scratch," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12, USA, 2012, pp. 764–772.