Deep Domain Adaptation for Face Recognition using images captured from surveillance cameras

Samik Banerjee[1], Avishek Bhattacharjee[2], Sukhendu Das[3]

Abstract: Learning based on convolutional neural networks (CNNs) or deep learning has been a major research area with applications in face recognition (FR). However, performances of algorithms designed for FR are unsatisfactory when surveillance conditions severely degrade the test probes. The work presented in this paper has three contributions. First, it proposes a novel adaptive-CNN architecture of deep learning refurbished for domain adaptation (DA), to overcome the difference in feature distributions between the gallery and probe samples. The proposed architecture consists of three components: feature (FM), adaptive (AM) and classification (CM) modules. Secondly, a novel 2-stage algorithm for Mutually Exclusive Training (2-MET) based on stochastic gradient descent, has been proposed. The final stage of training in 2-MET freezes the layers of the FM and CM, while updating (tuning) only the parameters of the AM using a few probe (as target) samples. This helps the proposed deep-DA CNN to bridge the disparities in the distributions of the gallery and probe samples, resulting in enhanced domain-invariant representation for efficient deep-DA learning and classification. The third contribution comes from rigorous experimentations performed on three benchmark real-world surveillance face datasets with various kinds of degradations. This reveals the superior performance of the proposed adaptive-CNN architecture with 2-MET training, using Rank-1 recognition rates and ROC and CMC metrics, over many recent state-of-the-art techniques of CNN and DA.

Keywords: Face Recognition; DA; Deep Learning; Low-Resolution; Denoising Auto-encoders.

1 Introduction

Deep learning (DL) has attracted several researchers in the field of computer vision due to its ability to perform face and object recognition tasks with higher accuracy than the traditional shallow learning systems. For biometric authentication, face recognition (FR) has been preferred due to its passive nature. Recent works using deep networks [PVZ15, Ta14, SKP15, Su15] for FR follow a purely data-driven approach, where the representations are directly learned from the pixels of the face. However, most solutions of FR fail to achieve higher accuracies when the training and the testing conditions vary. For face images captured using surveillance cameras, which are highly degraded, most FR systems fail to perform satisfactorily even with near-frontal test probes, since the gallery is obtained in controlled laboratory settings.

In a variation of transfer learning methods, domain adaptation tasks [Pa11, GGS13, Sa10, Go12a] attempt to minimize the discrepancy in the probability distributions of the source (gallery) and target (probes) domains. In this paper, we propose a novel adaptive-CNN architecture termed ‘deep-DA’, to perform FR efficiently using images from surveillance cameras. The network with 3 modules is trained using a novel 2-stage Mutually Exclusive Training (2-MET) process to minimize the disparity of the gallery and probe samples. The first stage of the 2-MET trains the network, while the final stage amends the network trained at stage 1 to accomplish the task of domain

[1] Visualization and Perception Lab, Department of CS&E, IIT Madras, Chennai, India, samik@cse.iitm.ac.in
[2] Visualization and Perception Lab, Department of CS&E, IIT Madras, Chennai, India, avi@cse.iitm.ac.in
[3] Professor, Visualization and Perception Lab, Department of CS&E, IIT Madras, Chennai, India, sdas@iitm.ac.in

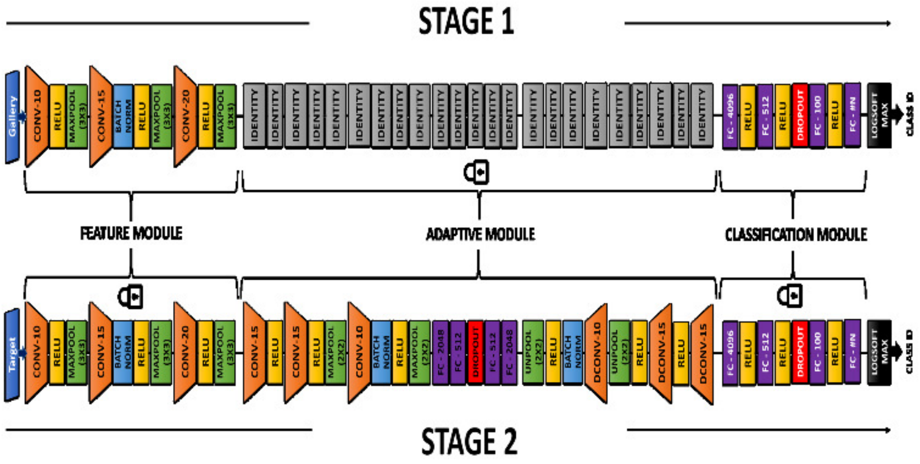

Fig. 1: Adaptive-CNN Architecture with 3 modules for 2-MET training (best viewed in color). The “lock” icon indicates that the module is frozen at that stage. ‘#*N*’ indicates the number of subjects in the dataset used.

adaptation (DA), such that the disparity in appearance between the gallery and probe samples is minimized. The adaptive-CNN architecture, reforms itself to adapt to the change in the gallery & probe samples, which are pre-processed [BSD14] to obtain tightly cropped face [As14] images.

A mutually exclusive 2-stage training of deep-DA, using 2-MET turns out to be significant for achieving high accuracy, where the design of the AM is inspired by stacked denoising autoencoders [Vi08]. This structure has the capacity to adapt to the testing environment, and in addition overcomes noise and aliasing artifacts present in the probe images acquired with surveillance cameras. This paper has three major contributions: (a) it proposes a novel adaptive-CNN architecture; (b) learning process consists of a novel 2-stage Mutually Exclusive Training (2-MET); (c) rigorous experimentations performed using 3 real-world datasets captured using surveillance cameras to exhibit superior performance of our proposed method.

2 Deep Domain Adaptation (Deep-DA)

For the task of DA, we are given a training (source) domain $D_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ with n_s labeled data, and a distinct test (target) domain $D_t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ that contains a small amount of labeled data, denoted by n_t, with varied characteristic probability distributions, q. The aim of deep-DA is to bridge the cross-domain discrepancy, and build a classifier $y = \theta(x)$ which can minimize the target risk $\varepsilon_t(\theta) = Pr_{(x,y)\sim q}[\theta(x) \neq y]$ using target supervision.

2.1 Adaptive-CNN Architecture

The adaptive-CNN architecture proposed in this paper is a deep CNN based on 3 modules, namely: Feature module (FM), Adaptive module (AM) & Classification module (CM); shown in figure 1.

2.1.1 Feature Module (FM)

The feature module (FM) mainly focuses on the generalized holistic features on the face. FM consists of 3 convolutional layers, interlaced with *Rectified Linear Unit (ReLU)* [ZF13] and a *MAXPOOLING* layer. As the data flows through a deep network the weights and parameters alter them, sometimes making the data too big or too small, referred to as ”internal covariate shift”. By normalizing the data in each mini-batch, this problem is largely avoided, adapted from the method proposed in [IS15], $y_i = BN_{\gamma,\beta}(x_i) \equiv \gamma \hat{x_i} + \beta$, where the normalization term is given by

$$\hat{x_i} = \mu_0 + \sqrt{\frac{\sigma_0^2 (x_i - \mu_B)^2}{\sigma_B^2}}, \quad \text{if } \hat{x_i} \geq \mu_B; \text{ or, } \hat{x_i} = \mu_0 - \sqrt{\frac{\sigma_0^2 (x_i - \mu_B)^2}{\sigma_B^2}}, \quad \text{if } \hat{x_i} < \mu_B \tag{1}$$

where, μ_0 and σ_0 is the overall mean andstandard deviation of the training set. Refer [IS15] for more details of the parameters. The shifted (normalized) values of y are passed to the subsequent layers.

2.1.2 Adaptive module (AM)

During stage-1 of 2-MET AM is frozen, passing the data unattenuated for input to the next layers, *i.e.*feed-forward without any modifications. DA involves the transformation of the extracted features from the target to the source domain. In line with this, we have positioned the AM module after the FM module which acts as the feature extraction module in CNN architecture. During stage-2 of training, the AM works as a stacked denoising auto-encoder modified from Vincent *et al.*[Vi08]: the shrinkage (encoder) and expansion (decoder) sub-modules. AM comprises of 3 convolution layers in the shrinkage sub-module interlaced with the ReLU and 2 maxpooling layers ($[p_l, s_l] = Pool(z_l)$, where z_l is the feature map fed to the layer l, p_l is the pooled map and s_l is the stride of the pooling). One convolutional layer having a 1×1 kernel is incorporated, as inspired by [SZ14], to introduce more non-linearity into the model. The shrinkage sub-module also contains 2 fully-connected (fc) layers. A dropout layer, with 50% probability, is also kept at the conjunction of the shrinkage and expansion sub-modules, to prevent overfitting.

The expansion sub-module in the AM is constructed as the mirror image of that of shrinkage. The unpooling layers correspond to each of the pooling layers in the previous sub-modules, as, $z_l = U_{s_l} p_l$ [ZTF11], where U_{s_l} is the unpooling layer corresponding to stride s_l for the layer l. The three deconvolutional layers aim to reconstruct the images corresponding to the convolutional layers of the shrinkage sub-module. The reconstruction [ZTF11] of $\hat{y_l}$ (comprising of c color channels) is formed by convolving each of the 2-D feature maps, $z_{k,l}$, with filters $f_{k,l}^c$ and summing them as: $\hat{y_l^c} = \sum_{k=1}^{K_1} z_{k,l} * f_{k,l}^c$, where $*$ is the 2D convolution operator.

2.1.3 Classification Module (CM)

The classification module (CM) has four fc layers which are mainly tailored for a particular task [Yo14]. Each fc layer learns a non-linear mapping, $h_i^l = f^l(W^l h_i^{l-1} + b_l)$, where h_i^l is the hidden representation of point x_i at the l-th layer, W^l and b^l are the weights and bias of the l-th layer, and f^l is the activation at rectifier units (ReLU), as: $f^l(x) = max(0,x)$ for hidden layers, or *logsoftmax* units, as: $f^l(x) = \frac{1}{a}\exp(x)$ where, $a = \sum_{j=1}^{|x|}\exp(x_j)$ for the output layers. If $\Theta = \{W^l, b^l\}_{l=1}^{|l|}$ denotes the set of all CNN parameters in CM, the empirical risk of CNN is

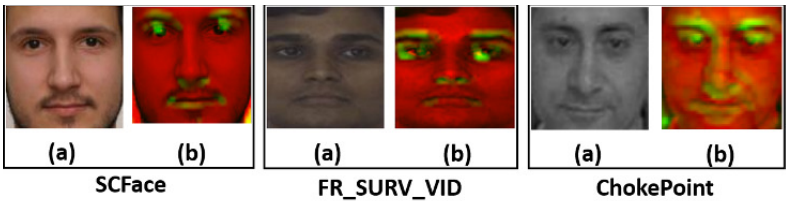

Fig. 2: For three real-world surveillance datasets: (a) The input gallery images, (b) The output showing the selected activation of one of the filters at *CONV20* layer (see figure 1) after stage 1 of training. The green shaded areas show the discriminative areas of the face, while the red indicates the non-discriminative areas.

represented as $\min_{\Theta} \frac{1}{n_a} \sum_{i=1}^{n_a} \mathscr{C}(\theta(x_i^a), y_i^a)$, where $\mathscr{C}$ is the cross-entropy loss function, and $\theta(x_i^a)$ is the conditional probability that the CNN assigns a label y_i^a to x_i^a. The training of AM is done exclusively at stage 2 of 2-MET, which is described below.

2.2 2-stage Mutually Exclusive Training (2-MET)

To achieve the outcome of superior domain-invariant recognition, a novel training algorithm is designed in two stages, outlined in sub-sections 2.2.1 and 2.2.2. The two stages involve mutually exclusive updates of different parameter sets. The parameters of the FM and CM are updated in stage 1, where the AM appears as Identity layers (frozen, with no parameter update). The output of trained module from stage 1 is then passed onto stage 2, where the Identity layers of the AM are replaced by the layers similar to a stacked denoising auto-encoder. The parameters of the FM and CM at this stage are frozen, and the update of the parameters is done only for the AM. Thus, the parameter updates take place for the modules in a mutual exclusion mode in 2-MET. This exclusive mode of training is necessary, else the deep layers (in FM and CM) will fail to overcome (map) the discrepancies in the source and target domains.

2.2.1 Stage 1 (for training FM and CM)

This stage of the adaptive-CNN network is trained using the gallery samples. The model is trained using SGD with standard backpropagation [Le89] using a batch size of 200 samples, momentum of 0.9 and weight decay of 0.005. Weight decay here is not merely a regularizer; it reduces the models training error [KSH12]. We initialized the weights in each layer using a zero-mean Gaussian distribution with standard deviation of 0.01. We initialized the neuron biases in the convolutional as well as fully-connected hidden layers with a constant 1.

The outputs of the FM are directly transferred to the input of CM by the stack of identity layers of the AM. The update of parameters in FM and CM helps the convolutional layers to automatically learn the discriminative features of the face, as shown in figure 2, where one sample from each of the three real-world surveillance datasets (described later in section 3) are shown in (a), while in (b) the output of a filter is shown, where greener the area more discriminative it is.

2.2.2 Stage 2 (for training AM)

A pre-trained model obtained from stage 1 is fine-tuned for adaptation under target supervision at this stage. Our aim is to transfer the trained model to adapt for the target task without updating a large set of its parameters.

In our proposed model, the AM unit attempts to map the disparity in the distributions between the source and the target domains to overcome DA. At stage 2 of training, the output of a FM unit, $z_P = FM(x_P)$ (with target samples, $x_P \in P_T$, available for DA, as input) is fed at the input layer of AM; while the same, $z_S = FM(x_S)$ (output of another identically pre-trained FM) but with the set of corresponding gallery samples, $x_S \in D_S$, given subject/class-wise (C^i) as input, is subsequently available at the output layer of AM (see figure 1(b)), for comparison with the output ($\tilde{z} = AM(z_P)$) of AM. The training process at stage-2 involves the minimization of the objective function ($\mathscr{J}$) (as in [Zh15]) by back-propagation, where

$$\mathscr{J} = J_r(z_S, \tilde{z}) + \alpha \mathscr{L}(\theta, \xi_P) + \beta \mathscr{T}(\kappa_S, \kappa_P) \tag{2}$$

where, α and β are the coefficients providing relative importance of each term. The first term in the objective function is the reconstruction error between z_S and $\tilde{z}$, which is defined in SSD form as:

$$J_r(z_S, \tilde{z}) = \sum_{i=1}^{n_P} \|z_S^i - \tilde{z}^i\|^2 \tag{3}$$

We assume here that n_P contains the target probe samples including those obtained by data augmentation.

The second term in equation 2 is the loss function of *softmax* regression used to perform the task of classification by the *softmax* layer at the end of CM (pre-trained at stage-1). Specifically, this term is:

$$\mathscr{L}(\theta, \xi_P) = -\frac{1}{n_P} \sum_{i=1}^{n_P} \sum_{j=1}^{C} 1\{y_P^i = j\} \log \frac{e^{\theta^{jT} \xi_P^i}}{\sum_{l=1}^{C} e^{\theta^{lT} \xi_P^i}} \tag{4}$$

where, ξ_P^i is the output of the layer preceding the soft-max regression layer, $\theta^{jT} (j \in C)$ is the parameter set corresponding to the j-th node of the *softmax* layer, and y_P^i is the predicted label. The minimization of this term implicitly helps to preserve the class labels for the features of the target samples in stage-2.

Let, κ_S & κ_P be the probability density functions (*PDF*s) of q_S & q_P respectively, where q_S & q_P are the flattened [KSH12] feature vectors z_S and $\tilde{z}$, respectively. The third term in equation 2 is the KL-divergence between κ_S (feature distribution of gallery samples) & κ_P (feature distributions of the transformed target probe samples). Thus, the third term can be expressed as:

$$\mathscr{T}(\kappa_S, \kappa_P) = KLD(\kappa_S, \kappa_P) + KLD(\kappa_P, \kappa_S) \tag{5}$$

where $KLD(T,S)$ is defined in equation 6. Minimization of this term in the objective reduces the gap between the gallery and the probe samples, as in DA. The training criterion for denoising auto-encoders used in TM, is based on the KLD (Kullback-Leibler Divergence) measure [KL51], given as:

$$KLD(T,S) = \sum_{x_S \in D_S, x_P \in P_T} T(x_P) \log \frac{T(x_P)}{S(x_S)} \tag{6}$$

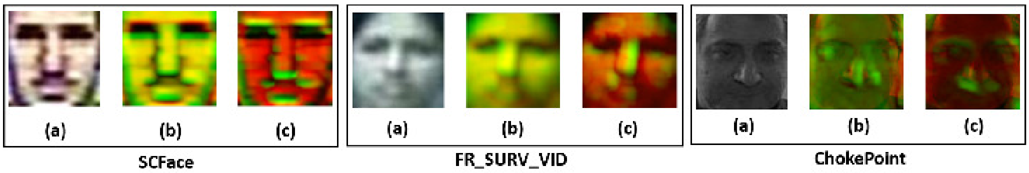

Fig. 3: (a) The input probe images, (b) The output of the selected activation of one of the filters at *CONV20* layer before stage 2 of training, (c) The same for one of the filters at *DCONV15* layer (see figure 1) after stage 2 of training (color code as described in figure 2) (best viewed in color).

where, $T(x_P)$ and $S(x_S)$ represent the target and source distributions, with a constraint that x_P and x_S represent a pair of target and source samples from class C^i. Regularization of the parameters (as done in [Zh15]) is implicitly incorporated in the *Tensorflow* based implementation of AM.

Figure 3(a) shows a few probe (target) samples, (b) shows the output of filters trained at stage 1 on the target samples, & (c) shows the output of the filters after stage 2 of 2-MET. Larger discrimination is visible by green labeled areas in figure 3(c). This justifies our claim for stage 2 of training in 2-MET. The AM at this stage also takes care of the background noise and aliasing effect present in the image, being an inherent property of the stacked denoising auto-encoder [Vi08]. This stage needs lesser computational time than stage 1.

3 Datasets

The proposed technique is evaluated over 3 real-world face datasets obtained using surveillance cameras namely, SCFace [GDG11], ChokePoint (CP) [Wo11] and FR_SURV_VID [RD11]; with many state-of-the-art methods based on DA and DL techniques used recently for FR.

The SCFace dataset [GDG11] (a standard dataset for evaluating FR under surveillance, with gallery and probe samples captured indoor) consists of 130 subjects. The training set consists of 9 mugshot images per sample as gallery and 15 probe samples per subject, captured using 5 different cameras at 3 different distances. While the average cropped gallery samples are 250 × 250 pixels, the probe images range from 15 × 15 to 45 × 45 pixels, at an average.

The ChokePoint (CP) dataset [Wo11] contains the faces of 54 subjects in two profiles, captured using three surveillance video cameras in an indoor environment. This is also a benchmark dataset for testing the performance of surveillance FR. In total, the dataset consists of 48 video sequences and $64,204$ face images. In our experimentations, the images taken by the camera $C1$ are considered as the training set, while that of the other pair ($C2$ and $C3$) are considered as the test samples. This results in a average of 500 face images per subject in training and 2500 in the testing pool. The datasets contains images of the same resolution averaging 80 × 80 pixels for the cropped face images.

The third dataset has the highest complexity among all these datasets. This dataset is a mild expanded version of the FR_SURV dataset [RD11], called FR_SURV_VID (FSV). The complexity of the dataset lies in the fact, the gallery images are captured indoor, but unlike other afore-mentioned datasets, the probe images are captured outdoor at uncontrolled environmental conditions with poor illumination, contrast, aliasing, large blur and low resolution. The training set has 250 face images (frontal pose) per subject on an average, with an average resolution of 150 × 150 pixels when cropped to get the face region. The testing set (video frames), captured outdoor using a surveillance video camera has 700 samples per subject, with the average cropped face image having a resolution of 33×33 pixels. The dataset has face images of 51 subjects. All these three datasets have no occlusion and negligible variations in expression variation and face pose.

Most real-world face datasets provide few samples in the gallery for training shallow transfer learning algorithms [PY10, BSD14]. This may be generally inadequate for training deep-CNN models. Hence, we use three different data augmentation techniques to artificially increase the size of the dataset, as proposed in [CMS12, KSH12], by three label-preserving techniques. The number of such target samples per subject, used for DA in each dataset, is artificially increased to a few thousands for training the deep-DA model (see table 1).

3.1 Preparing the data for the task

The gallery and the probe images vastly differ in their quality. For all three rows in table 1, the probe images are obtained using surveillance video cameras. They suffer from low resolution, low contrast, poor illumination, aliasing, blur and background noise, all predominantly present in FSV [RD11] dataset, making it the hardest of the lot. The gallery images have minimal background, but the probe samples suffer from background variations. We observed largely unsatisfactory performance (mentioned the face regions) to the learning/adaptation stages of deep-DA. All large CNN architectures failed to directly bridge the gap in source and target domains even satisfactorily. Hence, we relied on pre-processing compulsorily to obtain a reliable FR performance. To boost the performance of the FR, face detection was hence followed by a pre-processing stage.

We obtain a tightly cropped image based on the *Chehra* proposed by Asthana *et al.*[As14]. The tightly cropped face image eliminates any background information present in the face image. To cope with the low contrast and poor illumination setting present in the probe images of real-world surveillance datasets, the tightly cropped face samples are passed through a contrast-stretching stage, using the Power Law Transformation [Fa01]. The difference is resolution is overcome by applying a face hallucination technique on the probe images, proposed by Jin and Bougannis in [JB15]. The gallery samples are downsampled to match the resolution of the probe images. Further pre-processing of the gallery samples to match the probe samples includes degradation of the gallery samples using a Gaussian blur kernel followed by illumination normalization of the gallery and the probe images performed based on the method proposed by Xu and Savvides [JXS15]. This pre-processing is applied only for SCFace and FSV datasets, as the gallery and probe samples in all other datasets are similar in resolution. This pre-processed data is used for training, fine-tuning and testing.

Table 1 gives the number of training, testing and target samples used for experimentations, where the total number of samples used for training/target-adaptation incorporates those obtained by data augmentation [CMS12, KSH12] (*i.e.*additionally, several synthetic samples were generated). The limited number of labeled test samples forces us to use a small fraction of them (see 3^{rd} column under 'Target' label) for training the AM, which do not overlap with test probes used for performance analysis. About 5-15% of the test samples were used for fine-tuning in DA. In our case, the minimal number of samples used as the target set was empirically determined, based on the criteria that increasing the same does not significantly improve the performance of our method. This experimental condition was kept same, as done in most DA based applications [BD16, Go12a, Go12b] published in the recent past.

4 Experimental Details and Performance Analysis

Experimental setup is implemented with *Keras* using *Tensorflow*-backend, and run on a machine with i7-6720K 3 GHz processor and dual Nvidia Titan X GPU, with 64 GB RAM. In the experiments we start with a learning rate 0.03 which is gradually reduced through the training process. Random initialization of weights is used and the model runs for 50 to 70 hours for training. The margin α is set to 0.35. # samples used for training, adaptation and testing are given in table 1. The inputs were sent in a mini-batch of 200 for training, and batch-normalization is carried on the mini-batch. The input size to the network is $100 \times 100 \times 3$ (see figure 1). In the following, results of the performance analysis are discussed for our proposed deep-DA, compared with that of several recently published CNN & DA (shallow) models. The rigorous set of experimentations are broadly divided into 3 categories as: **1**. *Experimentation on Real-world Datasets* (section 4.1); **2**. *Unbiased Training* (section 4.2) and **3**. *Cross-dataset Adaptation* (section 4.3).

Tab. 1: The number of samples, used for experimentation. The target and test probes never overlap.

Datasets	Training (Gallery)	Target	Testing (Probes)
SCFace [GDG11]	2000	200	3000
ChokePoint [Wo11]	5000	1000	10000
FR_SURV_VID	4000	800	10000

4.1 Experimentation on Real-world Datasets

The comparison of the performance of our proposed deep-DA technique with recent state-of-the-art techniques for three real-world surveillance datasets is shown in table 2, using Rank-1 Recognition rates. Number of samples per subject for training, adaptation and testing are as given in table 1. The results in bold show the best performance accuracies. Existing CNN methods in rows $1-4$ of table 2 are adapted to the target domain by freezing the convolutional layers and training only

Tab. 2: Rank-1 Recognition Rates for different methods over 3 real-world surveillance face datasets. Results in bold, exhibit the best performance.

Sl.	Algorithm	SCface	CP	FSV
1	DeepFace [PVZ15]	52.67	73.26	40.32
2	DeepID 3 [Su15]	47.51	78.79	43.98
3	FaceNET [SKP15]	69.08	82.72	54.68
4	VGG-19 [SZ14]	59.35	80.14	46.32
5	Naive	35.24	61.59	18.62
6	FV_DCNN [Ch16]	63.78	80.62	52.32
7	SML_MFKC [BD16]	79.86	85.59	58.31
8	DAN [LW15]	74.57	83.86	55.43
9	LSDA [Ho14]	77.65	87.76	53.29
10	deepMSDA [Ch14]	73.21	88.97	57.25
11	K-NN at $FC100$	78.72	86.25	69.71
12	Deep-DA (ours)	**86.74**	**93.41**	**72.33**

(fine-tuning pre-trained models) the fc layers on the target samples. FaceNet [SKP15] performs the best among these four methods, which uses triplet-loss function for learning. The naive method (row 5) executes only the stage 1 of training (involving FM and CM), using a concatenation of the target and gallery samples as the training set, and then testing with the probes. It is noticeable that the methods (rows $6-10$) which incorporate DA techniques for adapting hand-crafted or convolutional features to the target domain, fare quite better in general than the other methods. The others (rows $1-4$) suffer from the fact that the specific or higher layers do not easily adapt to the target domain. All the other deep-transfer learning methods, in rows $8-10$ of table 2 use source supervision for adaptation, while variations of our proposed techniques in the last 2 rows $(11-12)$ rely on the target supervision. The second best performing method at row 16, ”K-NN at $FC100$”, indicates that a K-NN classifier is used with the feature maps obtained at the $FC100$ block of final module of CM. Finally, our proposed method (deep-DA) achieves the best and a significantly higher accuracy (see last row) than all other competing methods. The stacked denoising auto-encoder module helps to minimize the noise and aliasing effect in the probe images, which also boosts the performance. The third best (on an average) performing method is based on a shallow technique, SML_MFKC [BD16], which also does a source to target transformation for DA. To strengthen our claim, we also provide the CMC and ROC curves for the three datasets. Figure 4 shows the CMC and ROC plots for only the 5 best performing methods under comparison, for better visibility. The red curve in each sub-plot depicts the performance obtained by our method, which is superior to all other competing techniques.

4.2 Unbiased Training

Finally, to show the effectiveness of fine-tuning the deep-DA architecture using target samples in 2-MET training, we validate the effectiveness of our algorithm by unifying the gallery samples of many datasets for training, while adapting using target samples from only a particular dataset before testing on the same. A chimeric dataset is formed for training as given in left column of table 3, which can be considered as a large unbiased dataset not overfitted to any particular environment of acquisition. The results are reported in table 3.

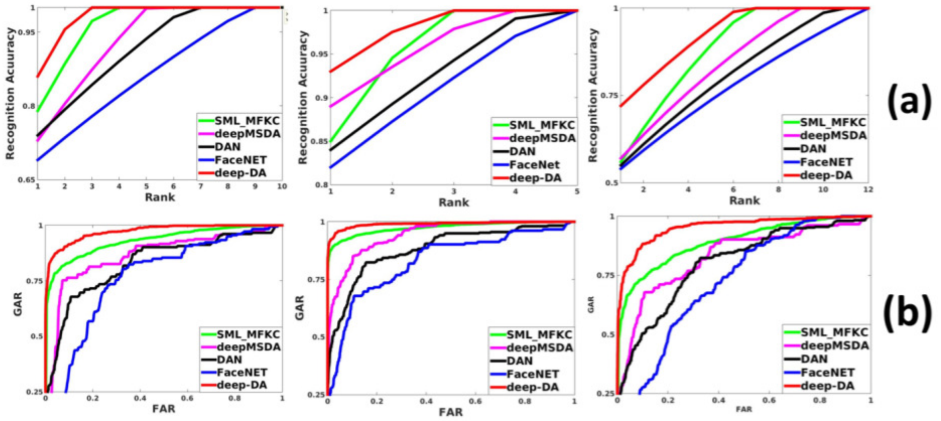

Fig. 4: (a) The CMC and (b) the ROC curves, showing superiority of our model, on the three datasets, from left to right, SCFace, ChokePoint and FR_SURV_VID. The curves marked in red show the results of our proposed method. Performances are shown only for the next 5 best performing methods (from those in table 2) used for comparison, as: SML_MFKC [BD16], deepMSDA [Ch14], DAN [LW15] and FaceNet [SKP15].

Tab. 3: Recognition Rate (%) when trained with two different large scale Chimeric Datasets. The degradation in performance range from $1-5\%$, when compared to last rows of table 2.

Training Datasets	Adapting & Testing Dataset	Recognition Rate (%)
SCFace + FR_SURV_VID + ChokePoint	SCFace	82.11
	FR_SURV_VID	69.11
	ChokePoint	**90.32**

None of the performances are better than those in the last rows of table 2, when compared dataset-wise respectively. This simply indicates an obvious fact that the best performances for each of the datasets under experimentation at at the bottom rows of table 2, when the same dataset is used for training/testing/target-adaptation. The amount of degradation in performance, when training is done using an unbiased, large chimeric dataset, range from $1-5\%$ (compare last rows of table 2, with the last column of table 3).

4.3 Cross-dataset Adaptation on surveillance datasets

In this mode of experimentation, the training dataset used at stage 1 of 3-MET is different than that used for stage 2 (adaptation). Specifically, training at stage 1 of 2-MET is done using gallery samples from a dataset, while the model is adapted to the target domain in stage 2 using target samples from a different dataset. The probes for test dataset are chosen to be either of the ones used for training at stage 1 or adaptation at stage 2 (latter being mostly a relevant use). The results showing the performance analysis for all the different pairs of combinations of training and adaptation datasets, are reported in table 4. Performance appears as a mixed bag. It appears that the FSV dataset is the toughest dataset to adapt or learn (when comparing row-wise average performances; also see lower rates at the last column of table 4). ChokePoint (CP) seems to be the simplest among the three to learn, as a combination of CP in adaptation and test probes gives the best accuracy (in general, 2^{nd} column from right has higher rates on an average). Our method is highly sensitive to the combination of training and adaptation datasets. In all cases, the performance degrades considerably if the training and adaptation datasets differ (compared to the accuracies reported in the last two rows of table 2). Exhibiting such findings has been the main purpose of this part of the experimentation.

Tab. 4: Rank-1 recognition rate (in %) of deep-DA for cross-dataset adaptation.

Sl.	Training	Adapt-ation	Test Probes		
			SCface [GDG11]	CP [Wo11]	FSV
1	SCFace	CP	70.46	72.36	-
2	SCFace	FSV	75.16	-	61.25
3	CP	SCFace	71.81	79.26	-
4	CP	FSV	-	78.65	59.17
5	FSV	SCFace	**78.24**	-	**66.87**
6	FSV	CP	-	**80.16**	63.02

5 Conclusion

The proposed deep-DA method efficiently transforms the source data to the target domain under limited target supervision. The three major contributions of the paper are: (a) it proposes a novel adaptive-CNN architecture, called deep-DA; (b) training done with a novel 2-stage Mutually Exclusive Training (2-MET); (b) rigorous experimentations performed on three real-world degraded face datasets show the superiority of our method. The fine-tuning of the model at stage 2 of 2-MET boosts the performance of FR. The 2-MET algorithm proposed in this paper maintains the principle of DA, where the source model remains unaltered during training. Our method outperforms all other recent state-of-art techniques for the 3 benchmark face datasets. Scalability of deep-DA may be verified with large real-world degraded face datasets when available to researchers.

References

[As14] Asthana, Akshay; Zafeiriou, Stefanos; Cheng, Shiyang; Pantic, Maja: Incremental face alignment in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1859–1866, 2014.

[BD16] Banerjee, Samik; Das, Sukhendu: Soft-Margin Learning for Multiple Feature-Kernel Combinations With Domain Adaptation, for Recognition in Surveillance Face Dataset. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) on Biometrics. pp. 169–174, 2016.

[BSD14] Banerjee, Samik; Samanta, Suranjana; Das, Sukhendu: Face Recognition in Surveillance Conditions with Bag-of-words, using Unsupervised Domain Adaptation. In: Indian Conference on Computer Vision Graphics and Image Processing (ICVGIP). 2014.

[Ch14] Chen, Minmin; Weinberger, Kilian Q; Sha, Fei; Bengio, Yoshua: Marginalized Denoising Auto-encoders for Nonlinear Representations. In: International Conference on Machine Learning (ICML). pp. 1476–1484, 2014.

[Ch16] Chen, J. C.; Zheng, J.; Patel, V. M.; Chellappa, R.: Fisher vector encoded deep convolutional features for unconstrained face verification. In: IEEE International Conference on Image Processing (ICIP). pp. 2981–2985, Sept 2016.

[CMS12] Ciregan, Dan; Meier, Ueli; Schmidhuber, Jürgen: Multi-column deep neural networks for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3642–3649, 2012.

[Fa01] Farid, H.: Blind inverse gamma correction. IEEE Transactions on Image Processing (TIP), 10(10):1428–1433, Oct 2001.

[GDG11] Grgic, Mislav; Delac, Kresimir; Grgic, Sonja: SCface–surveillance cameras face database. Multimedia Tools and Applications, 51(3):863–879, 2011.

[GGS13] Gong, Boqing; Grauman, Kristen; Sha, Fei: Connecting the Dots with Landmarks: Discriminatively Learning Domain-Invariant Features for Unsupervised Domain Adaptation. In: International Conference on Machine Learning (ICML). pp. 222–230, 2013.

[Go12a] Gong, Boqing; Shi, Yuan; Sha, Fei; Grauman, Kristen: Geodesic flow kernel for unsupervised domain adaptation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2066–2073, 2012.

[Go12b] Gopalan, Raghuraman; Taheri, Sima; Turaga, Pavan; Chellappa, Rama: A blur-robust descriptor with applications to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 34(6):1220–1226, 2012.

[Ho14] Hoffman, Judy; Guadarrama, Sergio; Tzeng, Eric S; Hu, Ronghang; Donahue, Jeff; Girshick, Ross; Darrell, Trevor; Saenko, Kate: LSDA: Large scale detection through adaptation. In: Advances in Neural Information Processing Systems (NIPS). pp. 3536–3544, 2014.

[IS15] Ioffe, Sergey; Szegedy, Christian: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[JB15] Jin, Yonggang; Bouganis, Christos-Savvas: Robust multi-image based blind face hallucination. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5252–5260, 2015.

[JXS15] Juefei-Xu, Felix; Savvides, Marios: Encoding and decoding local binary patterns for harsh face illumination normalization. In: IEEE International Conference on Image Processing (ICIP). pp. 3220–3224, 2015.

[KL51] Kullback, Solomon; Leibler, Richard A: On information and sufficiency. The Annals of Mathematical Statistics, 22(1):79–86, 1951.

[KSH12] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS). pp. 1097–1105, 2012.

[Le89] LeCun, Yann; Boser, Bernhard; Denker, John S; Henderson, Donnie; Howard, Richard E; Hubbard, Wayne; Jackel, Lawrence D: Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541–551, 1989.

[LW15] Long, Mingsheng; Wang, Jianmin: Learning transferable features with deep adaptation networks. CoRR, abs/1502.02791, 2015.

[Pa11] Pan, Sinno Jialin; Tsang, Ivor W; Kwok, James T; Yang, Qiang: Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks (NN), 22(2):199–210, 2011.

[PVZ15] Parkhi, Omkar M; Vedaldi, Andrea; Zisserman, Andrew: Deep face recognition. In: British Machine Vision Conference (BMVC). volume 1, p. 6, 2015.

[PY10] Pan, Sinno Jialin; Yang, Qiang: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering (KDE), 22(10):1345–1359, 2010.

[RD11] Rudrani, Shiva; Das, Sukhendu: Face recognition on low quality surveillance images, by compensating degradation. In: International Conference Image Analysis and Recognition (ICIAR). pp. 212–221, 2011.

[Sa10] Saenko, Kate; Kulis, Brian; Fritz, Mario; Darrell, Trevor: Adapting visual category models to new domains. In: European Conference on Computer Vision (ECCV). pp. 213–226, 2010.

[SKP15] Schroff, Florian; Kalenichenko, Dmitry; Philbin, James: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 815–823, 2015.

[Su15] Sun, Yi; Liang, Ding; Wang, Xiaogang; Tang, Xiaoou: Deepid3: Face recognition with very deep neural networks. arXiv preprint arXiv:1502.00873, 2015.

[SZ14] Simonyan, Karen; Zisserman, Andrew: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

[Ta14] Taigman, Yaniv; Yang, Ming; Ranzato, Marc'Aurelio; Wolf, Lior: Deepface: Closing the gap to human-level performance in face verification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1701–1708, 2014.

[Vi08] Vincent, Pascal; Larochelle, Hugo; Bengio, Yoshua; Manzagol, Pierre-Antoine: Extracting and composing robust features with denoising autoencoders. In: Int'l Conference on Machine learning (ICML). 2008.

[Wo11] Wong, Yongkang; Chen, Shaokang; Mau, Sandra; Sanderson, Conrad; Lovell, Brian C: Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) on Biometrics. pp. 74–81, 2011.

[Yo14] Yosinski, Jason; Clune, Jeff; Bengio, Yoshua; Lipson, Hod: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems (NIPS). pp. 3320–3328, 2014.

[ZF13] Zeiler, Matthew D; Fergus, Rob: Stochastic pooling for regularization of deep convolutional neural networks. arXiv preprint arXiv:1301.3557, 2013.

[Zh15] Zhuang, Fuzhen; Cheng, Xiaohu; Luo, Ping; Pan, Sinno Jialin; He, Qing: Supervised Representation Learning: Transfer Learning with Deep Autoencoders. In: International Joint Conference on Artificial Intelligence (IJCAI). pp. 4119–4125, 2015.

[ZTF11] Zeiler, Matthew D; Taylor, Graham W; Fergus, Rob: Adaptive deconvolutional networks for mid and high level feature learning. In: International Conference on Computer Vision (ICCV). pp. 2018–2025, 2011.