Neurocomputing 310 (2018) 299–315

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Mutual variation of information on transfer-CNN for face recognition with degraded probe samples

Samik Banerjee, Sukhendu Das*

Department of CS&E, IIT Madras, Chennai, India

ARTICLE INFO

Article history:
Received 21 December 2017
Revised 2 May 2018
Accepted 4 May 2018
Available online 7 June 2018

Communicated by Dr Zhen Lei

Keywords:
Face recognition
Transfer-CNN
Mutual variation of information
Domain adaptation
Deep-DA
TM
KLD
SGD
3-MET

ABSTRACT

Learning based on convolutional neural networks (CNNs) or deep learning has been a major research area with applications in face recognition (FR). Under degraded conditions, performance of FR algorithms severely degrade. The work presented in this paper has three contributions. First, it proposes a transfer-CNN architecture of deep learning tailor-made for domain adaptation (DA), to overcome the difference in feature distributions between the gallery and probe samples. The proposed architecture consists of three units: base convolution (BCM), transfer (TM) and linear (LM) modules. Secondly, a novel 3-stage algorithm for Mutually Exclusive Training (3-MET) based on stochastic gradient descent, has been proposed. The initial stage of 3-MET involves updating the parameters of the BCM and LM units using samples from gallery. The second stage involves updating the parameters of TM, to bridge the disparity between the source and target distributions, based on mutual variation of information (MVI). The final stage of training in 3-MET freezes the layers of the BCM and TM, for updating (fine-tuning) only the parameters of the LM using a few probe (as target) samples. This helps the proposed transfer-CNN to provide enhanced domain-invariant representation for efficient deep-DA learning and classification. The third contribution comes from rigorous experimentations performed on three benchmark real-world degraded face datasets captured using surveillance cameras, one real-world dataset with non-uniform motion blur and three synthetically degraded benchmark face datasets. This exhibits superior performance of the proposed transfer-CNN architecture with 3-MET training, using Rank-1 recognition rates and ROC and CMC metrics, over many recent state-of-the-art techniques of CNN and DA. Experiments also include performance analysis under unbiased training with two large-scale chimeric face datasets.

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 high accuracy than the traditional shallow learning systems. The convolutional layers present in the deep learning systems help to successfully capture the distinctive features of the face. For biometric authentication, face recognition (FR) has been preferred due to its passive nature. Most solutions of FR fail to achieve higher accuracies when the training and the testing conditions vary. For face images acquired using surveillance cameras, which are highly degraded, most FR systems fail to perform satisfactorily even with near-frontal test probes, since the rich gallery samples are obtained in controlled laboratory settings.

Convolutional Neural Network (CNN) architectures proposed earlier for FR, deal with the scenario where the gallery and the probe samples are acquired in similar environmental conditions. These approaches fail to achieve an acceptable accuracy when the gallery and probe samples differ in resolution, contrast and sharpness, due to the difference in camera parameters and environmental conditions in which they are captured. The downside of these techniques is their inability to cope with the changes in the distributions of training and testing conditions. One of the main limitations of CNN-based DL methods is its inability to conveniently adapt for transfer learning applications [3–5]. On the contrary, our proposed approach of transfer-CNN with 3-MET (3 stage mutually exclusive training) algorithm trains on the gallery samples and then embeds a knowledge transfer using a few test probes as target samples. Fig. 1 shows the variability in the appearance of the gallery and probe samples. The image pairs shown have extreme variations in appearance, in addition to significant degradations in the probe samples. Our aim is to design a method to overcome such variations efficiently.

* Corresponding author.
E-mail addresses: samik@cse.iitm.ac.in (S. Banerjee), sdas@iitm.ac.in (S. Das).

https://doi.org/10.1016/j.neucom.2018.05.038
0925-2312/© 2018 Elsevier B.V. All rights reserved.

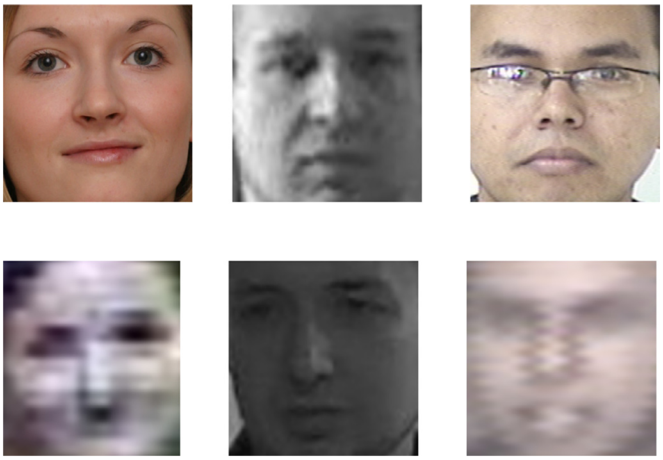


Fig. 1. The bottom row shows the degraded probe images, while the top row shows the crisp gallery samples, one from each of the three datasets, SCFace [1], ChokePoint [2] and FR_SURV_VID (placed from left to right), respectively.

In this paper, we designed a transfer-CNN architecture termed ‘deep-DA’ for FR, which performs efficiently under such degraded conditions. Henceforth, we will use the terms transfer-CNN and deep-DA interchangeably in the manuscript. The network with three separable units is trained using a novel 3-stage Mutually Exclusive Training (3-MET) process to minimize the disparity of the gallery and probe samples. The first stage of the 3-MET process performs pre-training of the Base Convolutional module (BCM) and Linear module (LM) using only the gallery (source) samples, while the second stage exclusively trains only the Transfer module (TM) to accomplish the task of domain adaptation (DA), such that the disparity in feature distributions between the gallery and probe samples is minimized, based on the mutual variation of information (MVI). The proposed TM unit needs training with a few probe (target) samples, to acquire information about the feature distribution due to varied environmental (testing) conditions and camera parameters. The transfer-CNN architecture, reforms itself to adapt to the change in the appearance of the gallery and probe samples, which are pre-processed to obtain tight crops of the face area in the images. The third stage of training involves minor update (fine-tuning) of the weights in LM unit.

A mutually exclusive 3-stage training of deep-DA, using MVI achieves high accuracy, where the design of the TM is inspired by stacked denoising auto-encoders [6]. 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 versatile transfer-CNN architecture with three units designed for DA; (b) the learning process consists of a novel design of a 3-stage Mutually Exclusive Training (3-MET), based on mutual variation of information; (c) rigorous experimentations performed using 4 real-world degraded and 3 synthetically degraded large face datasets, to exhibit superior performance of our proposed method over other deep-CNN and DA methods published in literature.

As the probe images are taken using surveillance cameras, the images suffer from severe degradation due to low resolution, low contrast, aliasing effect, background noise (low SNR) and large blur (for SCFace [1], ChokePoint [2] and FR_SURV_VID datasets). The discriminatory features of the face even after super-resolving [7], were found to be inadequate. The contracting nature of the stacked denoising auto-encoder helps to focus the attention of the model on the discriminative features, thus suppressing the non-discriminative parts. Moreover, this representation provides an additional advantage of invariance (tolerance) to noise and aliasing artifacts of the probe images, implicitly by the network itself (no additional pre-processing required). For other datasets used for adaptation, the probe images in TIP dataset [8] suffer from non-uniform motion blur, while the test data from FERET [9], PIE [10] and LFW-S [11] datasets have been synthetically blurred. We assume that test probes are of near-frontal pose and do not have occlusions and much of expression variations.

In the rest of the paper, Section 2 gives a literature review in the relevant work; Section 3.1 describes the proposed transfer-CNN architecture and the novel 3-MET algorithm. Section 4 gives details of the datasets used for experimentations. In Section 5 quantitative results of our experiments, showing the effectiveness of our proposed method is reported. Finally, the paper concludes in Section 6.

2. Related work

Recent works using deep networks [12–17] for FR follow a purely data-driven approach, where the representations are directly learned from the pixels of the face. The databases experimented in such cases contain labeled faces with variations. The authors resort to the vastness of the datasets to attain invariance to pose and occlusion. The LFW database [11] is a benchmark dataset to test the recognition system’s performance in an unconstrained environment. This has proved to be much harder than many other constrained datasets (e.g. YaleB [18] and Multi-PIE [19]). Recent DL techniques [12–15,20–23] have shown surprising improvements of FR in the LFW and YoutubeFaces [24] datasets. Zhao et al. [25,26] has studied several techniques on FR and face modeling, published in the recent past. Specifically, Zhou et al. [27] also studied several shallow techniques available for unconstrained face recognition which have failed to produce satisfactory results. Recent works on degraded faces in surveillance scenarios has also been studied using shallow techniques [28,29], without considerable success.

The works proposed in [15,20,21] mainly deal with a multi-stage complex systems, which take the convolutional features obtained from their model and then use PCA (Principal Component Analysis) for dimensionality reduction, followed by classification using SVM. Zhu et al. [20] tries to warp faces into a canonical frontal view using a deep network, for efficient classification. PCA on the network output in conjunction with an ensemble of SVMs is used for the face verification task. Taigman et al. [15] propose a multi-stage approach that aligns faces to a general 3D shape model combining with a multi-class (deep) network which is trained to perform the FR task. The compact network proposed by Sun et al. [14,21,22] uses an ensemble of 25 of these networks, each operating on a different face patch. The FaceNet proposed by Schroff et al. [13] uses a deep CNN to directly optimize the embedding itself, based on the triplet loss formulated by a triplet mining method. The DeepFace approach proposed by Parkhi et al. [12] uses a deep network to train on the large-scale face datasets. DeCAF [30] is an open-source implementation of deep convolutional activation features, along with all associated network parameters to enable vision researchers to be able to conduct experimentation with deep representations across a range of visual concept learning paradigms.

The work done by Chen et al. in [31] shows that deep neural networks learn non-linear representations that can provide invariance to the different variations amongst data samples. The authors had used deep neural networks for exploring DA, with the use of marginalized stacked auto-encoder for source supervision [32]. Long et al. [32] proposed a Deep Adaptation Network (DAN) for DA using MK-MMD (multiple kernel variant of maximum mean discrepancies). In both the cases, results are shown on only object and character recognition tasks. Deep Reconstruction Classification

Network (DRCN) [33] jointly learns a shared encoding representation for two tasks: i) supervised classification of labeled source data, and ii) unsupervised reconstruction of unlabeled target data. For such an approach, the learnt representation not only preserves discriminability, but also encodes useful information from the target domain. This paper does not talk about the DA for classification of the target domain data. Alain and Bengio [34] generalized the result of any parameterization in the encoder and decoder with squared reconstruction error and Gaussian corruption noise. They show that as the amount of degradation due to noise approaches zero, such models estimate the true score of the underlying data generating distribution. Finally, Bengio et al. [35] show that any denoising auto-encoder is a consistent estimator of the underlying data generating distribution within some family of distributions. The above holds good under the following scenarios: for any parameterization of the auto-encoder; for any type of information-destroying corruption process with no constraint on the noise level except being positive; and for any reconstruction loss expressed as a conditional log-likelihood. The consistency of the estimator [35] is achieved by associating the denoising auto-encoder with a Markov chain whose stationary distribution is the distribution estimated by the model, and this Markov chain can be used to sample from the denoising auto-encoder.

Domain adaptation is a fundamental problem in machine learning and has gained a lot of traction in natural language processing, statistics, machine learning, and, most recently, in the computer vision (CV) [36] field. In a variation of transfer learning methods, domain adaptation tasks [4,5,33,37–43] attempt to minimize the discrepancy in the probability distributions of the source (gallery) and target (probes) domains. Most of the existing methods learn a new shallow representation model in order to reduce the disparity between the source and target domains. Duan et al. [44] proposed a new learning method for heterogeneous domain adaptation (HDA), in which the data from the source and target domains are represented by heterogeneous features with different dimensions. Using two different projection matrices, they first transformed the data from two domains into a common subspace in order to measure the similarity between the data from two domains. With recent advances in the field of Generative Adversarial Networks (GANs) [45–47], a few attempts have been made to transfer the knowledge from one domain to another, but not with faces as objects. In our earlier work [48], an optimal feature-kernel combination had been used for adaptation to the target domain. Results were shown on three benchmark real-world degraded face datasets captured using surveillance cameras. These shallow methods are very task-specific with hand-crafted features, which do not completely overcome the domain-specific factors, resulting in residual disparity in feature distributions even after transformations of the data. Instead, our proposed approach relies on suitable training of the models for the transferability of the deep features.

Our proposed deep-DA model with 3-MET learning paradigm overcomes these drawbacks and makes CNN competent to handle DA for FR approaches, which are yet to be thoroughly explored in literature. Details of the design are in the next section.

3. Deep domain adaptation (Deep-DA)

For the task of DA, we are given a source (gallery) domain $D_S = \{(x_S^i, y_S^i)\}_{i=1}^{n_S}$ with n_S labeled data, and a distinct test (probes) domain $D_T = \{(x_T^i, y_T^i)\}_{i=1}^{n_T}$ that consists of n_T labeled (ground-truth) data, with varied characteristic probability distributions. Our proposed deep-DA model assumes that a small amount ($n_P << n_T$) of the labeled probe data as $P_T \subset D_T$ with $n_P = |P_T|$, is available from the test domain for training (as in [3,41]). We call this set as target probes used for DA. P_T does not overlap with D_T. The aim of deep-DA is to bridge the cross-domain discrepancy, and build a classifier $y = \theta(x)$ which can minimize a risk $\epsilon_t(\theta) = Pr_{(x,y)\sim q}[\theta(x) \neq y]$ using target supervision [32]. Let, C^i be the i^{th} subject in a dataset, $\forall C^i \in \{1, \ldots, C\}$, where C denotes the total number of classes.

Flowcharts illustrating for the training and testing phases of deep-DA architecture are shown in Fig. 2. In the following subsections, the details of the architectures of BCM, TM and LM units as elaborated in Fig. 3 are described with details in Section 3.1, while the stages 1, 2 and 3 of training phase of deep-DA architecture are detailed in Fig. 4. The stage-wise training procedures of 3-MET are detailed in Section 3.2. As per the diagram in Fig. 2, the training stage uses the entire gallery samples as source domain (D_S) and a few samples from probes as target domain (P_T). Since our mode of operation is supervised DA, the class-IDs are also used during training, which has three stages (mutually exclusive) of cost minimization on appropriately chosen units of deep-DA. During testing, all probes from test domain (D_T) are used for observing the performance of our deep-DA model. Hence, the data flow is in a sequence from BCM through TM and then to LM for obtaining the class IDs. TM is the main (pivot) unit responsible for providing support for DA, and works in a denoising auto-encoder type mode.

3.1. Transfer-CNN architecture

The transfer-CNN architecture proposed in this paper is a deep convolutional network based on three units, namely: Base Convolutional module (BCM), Transfer module (TM) and Linear module (LM) - all are described in the following subsections. The three principle units of the detailed architecture are shown in Fig. 3. BCM unit is mainly responsible for low-level feature extraction, TM is responsible for transfer of knowledge from source to target domains (in our case, gallery to probe faces), whereas LM is responsible for high-level feature extraction and classification. Instead of training these units together, we have proposed a 3-MET algorithm, which performs exclusive training in 3 stages: BCM and LM at stage-1 for feature extraction; TM at stage-2 to overcome DA (transfer of knowledge) across features sets (source to target) with varying distributions; and finally LM at stage-3 for classification. The details (analytics) of the 3 stages of 3-MET training are explained later in Section 3.2. According to the traditional DA techniques [4,5,37,38], a small subset of target (in our probes) samples are available for fine-tuning or re-training. It turns out to be disastrous if one attempts to train a deep-CNN architecture, using a small subset of samples (even enhanced by data augmentation). Cross dataset applications of existing deep-DA methods published in literature [45,49,50] use a completely different dataset to fine-tune a pre-trained model consisting of a large set of samples. To avoid computational hazards (vanishing gradients, lack of convergence, etc.), we only train a particular component of our proposed deep-DA model at each stage of 3-MET. This we found to be logical as fine-tuning the TM component using a small set of target samples available for DA helps to transfer the knowledge from low-level features in BCM to high level ones in LM across domains.

The outputs of the last few deconvolutional layers of TM are combined into a 1D feature vector and passed into the first fully-connected layer of LM. Structural details of transfer-CNN architecture are given in Appendix A. Details of operations in BCM, TM and LM units follow.

3.1.1. Base convolutional module (BCM)

The base convolutional module (BCM) mainly focuses on the generalized holistic features of the face. BCM consists of three convolutional layers, interlaced with *Rectified Linear Unit (ReLU)* [51] and a *MAXPOOLING* layer [52]. ReLU induces non-linearity and sparsity in the network, due to its activation function $h = max(0, a)$, where $a = \mathbf{W}x + \mathbf{b}, \mathbf{W}$ and $\mathbf{b}$ are the weights and bias of the layer. Max-pooling is a form of non-linear down-sampling,

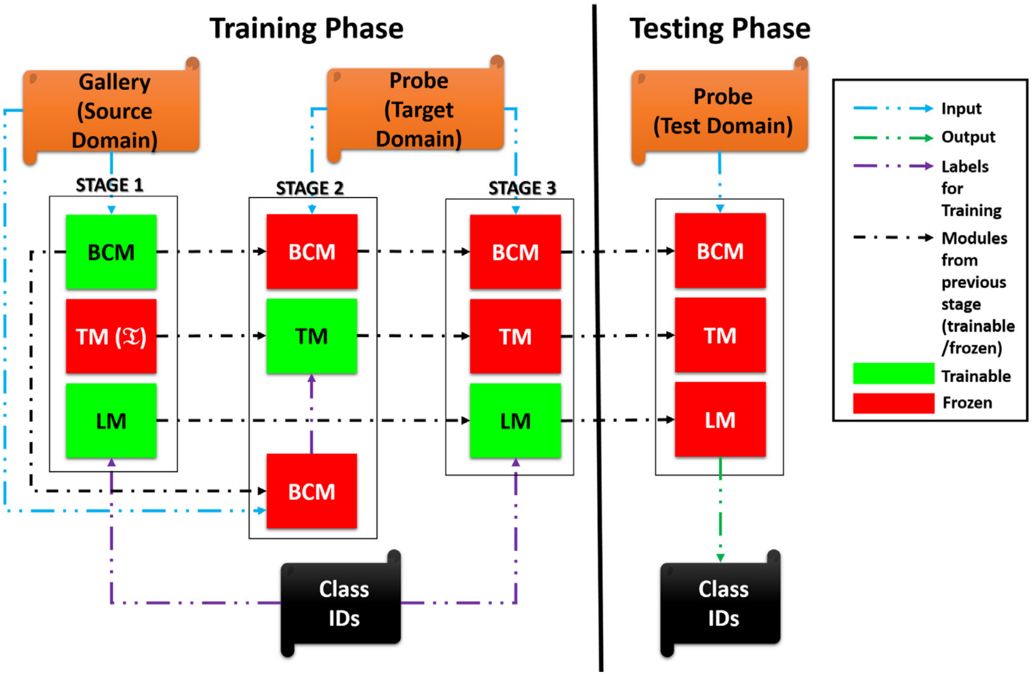


Fig. 2. Flowchart showing the training and testing phases of Deep-DA architecture (best viewed in color). Also see Figs. 3 and 4 and the detailed discussions in Sections 3.1 and 3.2 for details. '𝔗' indicates that the layers in the TM unit at stage-1 act as Identity layers.

which helps in eliminating the non-maximal values to reduce the computation of the higher layers and provides a form of translational invariance. As the data flow 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”. Using normalization of the data in each mini-batch, this problem is largely avoided. The batch-normalization layer is adapted from the method proposed by Ioffe and Szegedy [53], as:

$$g^i = BN_{\gamma,\beta}(x^i) \equiv \gamma\hat{x^i} + \beta \tag{1}$$

where the normalization term is given by,

$$\hat{x^i} = \mu_0 + \sqrt{\frac{\sigma_0^2 (x^i - \mu_B)^2}{\sigma_B^2}}, \quad \text{if } \hat{x^i} \geq \mu_B$$

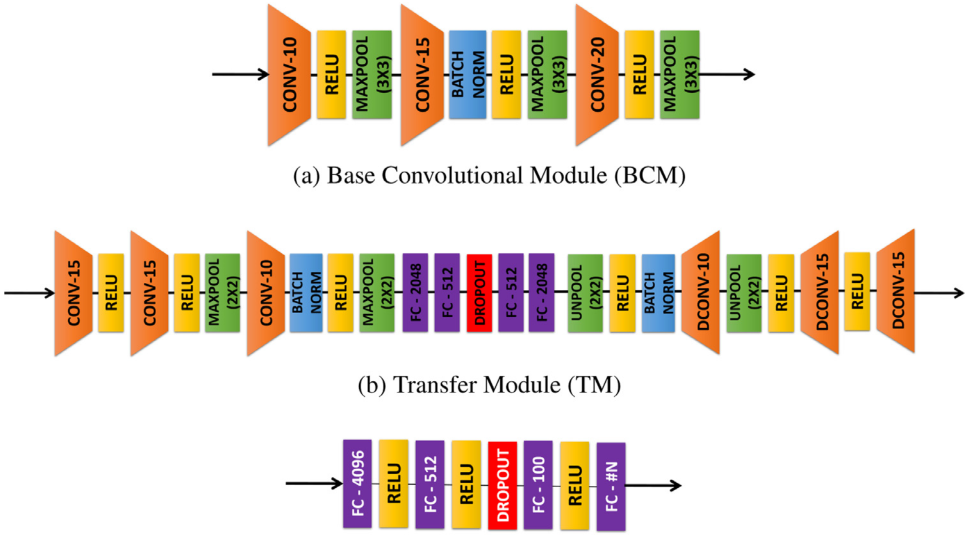
$$\hat{x^i} = \mu_0 - \sqrt{\frac{\sigma_0^2 (x^i - \mu_B)^2}{\sigma_B^2}}, \quad \text{if } \hat{x^i} < \mu_B$$

$$\text{where, } \mu_0 \text{ and } \sigma_0 \text{ are the overall mean and}$$
$$\text{standard deviation of the training set, } D_S. \tag{2}$$

Refer [53] for more details of the parameters of Eq. (1). The normalization function given in Eq. (1) has been modified from that given in [53] for better performance. The shifted (normalized) values of g, as given in Eq. (1), are passed to the subsequent layers.

3.1.2. Transfer module (TM)

The transferability gap grows with domain discrepancies at the higher layers, which is very large [49]. The fully-connected (*fc*) layers are tailored to their original task at the expense of the

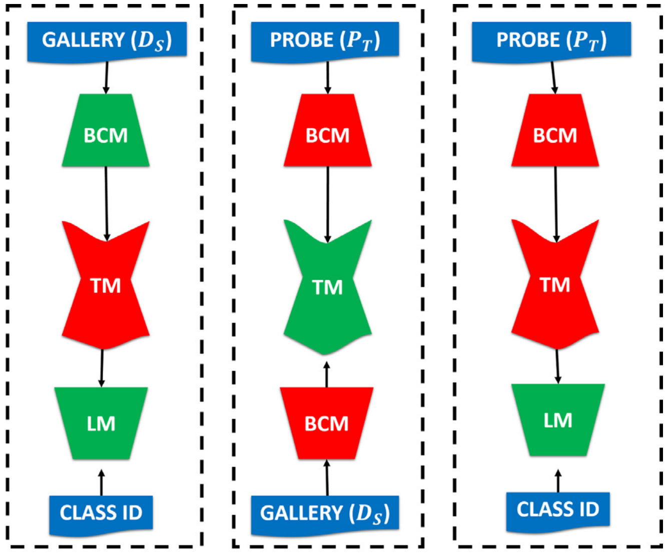


(a) Base Convolutional Module (BCM)

(b) Transfer Module (TM)

(c) Linear Module (LM)

Fig. 3. Architectural details of the three units in the Transfer-CNN architecture (best viewed in color).



(a) Stage I (b) Stage 2 (c) Stage 3

Fig. 4. Block Diagram illustrating the three stages of learning in 3-MET process (best viewed in color). The units in red indicate those which are the frozen (passive) with no weight update, while the green units indicate active parts for parameter update at a particular stage (best viewed in color). (a) Stage 1 – Base training using D_S, (b) Stage 2 – DA using class-wise pairing of D_S and P_T, and (c) Stage 3 – Fine-tuning using P_T. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

degraded performance of the target task. Hence these *fc* layers cannot be easily transferred to the target domain with limited target supervision [32], or fine-tuning (in our case). This drawback will be illustrated later (in Section 5.1), where the performance of the deep-DA model will be empirically verified with benchmark face datasets (Table 2). Thus, we incorporate a transfer (adapting) module between the generic (BCM) and specific linear modules (LM), which forms the backbone of our proposed architecture.

In our problem domain (of DA), the state-of-the-art statistical classifiers and end-to-end deep convolutional networks were unable to adapt to the degradation in the face quality of probe samples, as their training algorithms were not designed to overcome the disparity in distributions of gallery and probe sets. This motivated us to use an exclusive unit (TM) as autoencoder to accomplish the task of DA. The TM unit with the 3-MET gives the power to the overall architecture to adapt and implicitly learn the degradation function (even non-linear) which naturally occurs with the degraded probes.

During stage-1 of 3-MET training the TM was frozen, passing the data unattenuated for input to the next unit, *i.e.* feed-forward without any modifications. The TM consisting of variants of auto-encoders and decoders, updates its parameters only at the second stage of training. The module is inspired by the stacked denoising auto-encoder, proposed by Vincent et al. [6]. DA in our case, involves the transformation of the extracted features from the target to the source domain. In line with this, we have positioned the TM after the BCM unit which acts as the feature extraction module in CNN architecture.

An autoencoder is characterized by an encoder (E) and a decoder (D), which is represented as (similar to [54]):

$$\mathbf{E}\textbf{:}\ \phi = f(\mathbf{W}_1 z + \mathbf{b}_1),\ \mathbf{D}\textbf{:}\ \tilde{z} = f(\bar{\mathbf{W}}_1 \phi + \bar{\mathbf{b}}_1) \qquad (3)$$

where, f is the activation function, $\mathbf{W}_1, \mathbf{b}_1, \bar{\mathbf{W}}_1, \bar{\mathbf{b}}_1$ are the weights and the biases of the hidden layers, z and $\tilde{z}$ are inputs and outputs of the auto-encoder, and the reconstruction error J_r is given by

$$\min \sum_i \|\tilde{z} - z\|^2 \qquad (4)$$

By minimizing the reconstruction error on the deep-CNN, the property of the denoising auto-encoder is implicitly imposed on our model.

During stage-2 of training, the TM works as a denoising auto-encoder [6,55] with the shrinkage (encoder) and expansion (decoder) sub-modules. Inline with [6,55]. we specify ”the input is stochastically corrupted (considering the probes as the degraded version of gallery images), but the uncorrupted input (gallery images) is still used as target for the reconstruction”. Intuitively, a denoising auto-encoder does two things: (i) encodes the input (preserves the information about the input), and (ii) reverses the effect of degradation, stochastically applied to the input of the auto-encoder. In DA [3,41], the latter is usually done by capturing the statistical dependencies between the inputs. Also, ”the stochastic degradation (as in non-uniform blur) process involves randomly setting some of the inputs (as many as half of them) to zero (represented by the *DROPOUT* layer in the TM). Hence the denoising auto-encoder predicts the missing values from the non-missing values, for randomly selected subsets of missing patterns”. The TM unit is thus fed with combinations of feature pairs of probes and gallery, formed class-wise. The pre-trained BCM is used to derive these features, which itself (as well as LM) is not updated at this stage.

TM comprises of three convolution layers in the shrinkage sub-module interlaced with the ReLU activation layer and two max-pooling layers ($[p_l, s_l] = Pool(m_l)$, where m_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 [56], to introduce more non-linearity into the model. The shrinkage sub-module also contains two 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 TM is constructed as the mirror image of that of shrinkage. The unpooling layers corresponding to each of the pooling layers in the shrinkage sub-modules, are characterized by, $m_l = U_{s_l} p_l$ [57], 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 [57] (for all color channels) is formed by convolving each of the 2-D feature maps, m_l^k, with filters F_l^k and summing them as: $G_l = \sum_{k=1}^{K_1} m_l^k * F_l^k$, where $*$ is the 2D convolution operator.

3.1.3. Linear module (LM)

The linear module (LM) has four *fc* layers which are mainly tailored for a particular task [49]. Each *fc* layer learns a non-linear mapping, $h_l^i = f_l(\mathbf{W}_l h_{l-1}^i + \mathbf{b}_l)$, where h_i^l is the hidden representation of point x^i at the lth layer, $\mathbf{W}_l$ and $\mathbf{b}_l$ are the weights and bias of the lth layer, and f_l is the activation function at rectifier units (ReLU), as: $f_l(x) = max(0, x)$ for hidden layers, or *softmax* layers for the output layers. The last layer of LM is always a fully-connected layer with one output unit per class, used for the recognition task, with a *softmax* activation function, defined as:

$$y = f_{SM}(\xi) = \max_j \frac{e^{\theta^{jT}\xi}}{\sum_{u=1}^{C} e^{\theta^{uT}\xi}} \qquad (5)$$

such that each neuron's output activation can be an interpretation of the belongingness of a particular face image for that subject, where ξ denotes the input to the *softmax* regression layer, θ^j represents the parameter set corresponding to the jth node in the *softmax* layer, with y being the class-labels in $\{1, \ldots, C\}$.

If $\Theta = \{\mathbf{W}_l, \mathbf{b}_l\}_{l=1}^{|l|}$ denotes the set of all CNN parameters in LM, the empirical risk of CNN is $\min_\Theta \frac{1}{n_a} \sum_{i=1}^{n_a} J(\theta(x_a^i), y_a^i)$, where J is the cross-entropy loss function, and $\theta(x_a^i)$ is the conditional probability that the CNN assigns a label y_a^i to x_a^i. The training of BCM, TM and LM are done exclusively at the different stages of 3-MET, as described in the following.

3.2. 3-stage mutually exclusive training (3-MET)

Adaptation under source supervision had been proposed by Long and Wang [32] using MK-MMD. Yosinski et al. [49] also proved that the *fc* layers cannot be directly transferred to the target domain under limited target supervision. To overcome this, we have proposed a TM unit (see Fig. 3) in the architecture, which can be used to transfer the knowledge in the entire network to adapt for the target domain, under limited target supervision [32] without altering the parameters of the modules trained on the source domain.

To achieve this outcome of superior domain-invariant recognition, a novel training algorithm is designed in three stages (see Fig. 4), described in Sections, 3.1.1–3.1.3. The three stages involve mutually exclusive updates of different parameter sets. The parameters of the BCM and LM are updated in stage 1, where the TM is frozen. The outputs of two separate BCM units identically trained at stage 1, are then used at stage 2, where the layers of the TM are similar to a stacked denoising auto-encoder. The parameters of the BCM and LM are frozen at this stage, and only the parameters for the TM are updated. In the third and final stage, a fine-tuning of the parameters of the pre-trained LM unit is done with a limited set of target samples (P_T). Thus, the parameter updates take place for the modules in a mutual exclusion mode in 3-MET. This exclusive mode of training is necessary, as the deep layers (in BCM and LM) fail to overcome (map) the discrepancies in the source and target domains.

3.2.1. Stage 1 (for pre- (or base-) training of BCM and LM)

Here, the two units (BCM and LM) of the transfer-CNN architecture are trained together (end-to-end, with TM suppressed) using the gallery samples (D_S). The training (batch mode) is done using Stochastic Gradient Descent (SGD) with standard backpropagation [58,59], using a batch size of 500 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 [60]. The update rule for the weight w is given as:

$$\begin{aligned} v^{i+1} &:= 0.9 \cdot v^i - 0.005 \cdot \epsilon \cdot w^i - \epsilon \left\langle \frac{\delta L}{\delta w} \Big|_{w^i} \right\rangle_{B^i} \\ w^{i+1} &:= w^i + v^{i+1} \end{aligned} \tag{6}$$

where, i is the iteration index, v is the momentum variable, ϵ is the learning rate and $\left\langle \frac{\delta L}{\delta w} |_{w^i} \right\rangle_{B^i}$ is the average of the derivatives of the objective function with respect to w, evaluated at w^i [60] over the ith batch, B^i. We initialized the weights in each layer using a zero-mean Gaussian distribution with standard deviation 0.01. We initialized the neuron biases in the convolutional layers, as well as in the fully-connected hidden layers, with a constant 1.

The outputs of the BCM are directly transferred to the input of LM by the frozen layers of the TM (initialized as layers with identity weights). The update of parameters in BCM and LM helps the convolutional layers to automatically learn the discriminative features of the face, as shown in Fig. 5. In Fig. 5, one sample from each of the three real-world degraded datasets (described later in Section 4) are shown in (a), while in (b) the output of the activations of a filter in BCM are shown, where more greener the area more discriminative it is. In Fig. 5, green areas show high values of activation functions at the outputs of *CONV20* layer in BCM after stage 1 of training. Hence these regions must be more discriminative across subjects. This illustration follows a similar approach as that given in [61–63]. To reduce the computational complexity of the network, we chose a maximum of 20 filters to capture the discriminative features of the face, compared to the 384 filters in AlexNet [60], where most of the filter outputs for the probe images contain negligible or no information. BCM and LM units pretrained at stage-1 are next used for the subsequent stages of learning, as discussed below.

3.2.2. Stage 2 (for DA-based training of TM)

At stage 1 of training, the BCM and LM units were trained using the gallery (D_S) samples, where BCM acts as the feature extractor and the LM acts as the regressor. Thus LM can regress the gallery (source) distribution. Now, the outputs of two identical pre-trained BCM units, obtained separately for both the gallery (D_S) and probe (P_T) samples, are then fed class-wise to the TM (a stacked denoising auto-encoder) unit at stage 2 of training (see Fig. 3(b)).

Despite various attempts [31,64] to address the problem of DA, researchers in this field tend to update the trained model parameters only at the higher levels. Our aim is to transfer the trained model to adapt for the task of DA without updating a large set of its parameters. To implement this, the process of training the transfer module (TM) at stage 2 has been adopted from the concept of stacked denoising auto-encoder [6]. In our case, since the source and target distributions are very dissimilar, the stacked denoising auto-encoder is successful in mapping and hence overcoming the disparity between them.

In our proposed model, the TM 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 BCM unit, $z_P = BCM(x_P)$ (with target samples, $x_P \in P_T$, available for DA, as input) is fed at the input layer of TM; while the same, $z_S = BCM(x_S)$ (output of another identically pre-trained BCM) 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 TM (see Fig. 3(b)), for comparison with the output ($\tilde{z} = TM(z_P)$) of TM. The training process at stage-2 involves the minimization of the objective function ($\mathcal{J}$) (modified from that proposed in [65]) by back-propagation, where

$$\mathcal{J} = J_r(z_S, \tilde{z}) + \alpha \mathcal{L}(\theta, \xi_P) + \beta \mathcal{T}(\kappa_S, \kappa_P) + \gamma MVI(z_S, \tilde{z}) \tag{7}$$

where, α, β and γ 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:

$$J_r(z_S, \tilde{z}) = \sum_{i=1}^{n_P} \| z_S^i - \tilde{z}^i \|^2 \tag{8}$$

We assume here that n_P contains the target probe samples including those obtained by data augmentation. This term is the auto-encoder loss as given in Eq. (4). Authors of [65] used an $L1-norm$ for this component of the objective function.

The second term in Eq. (7) is the loss function of *softmax* regression used to perform the task of classification by the *softmax* layer at the end of LM (pre-trained at stage-1). Specifically, this term is:

$$\mathcal{L}(\theta, \xi_P) = -\frac{1}{n_P} \sum_{i=1}^{n_P} \sum_{j=1}^{C} 1\{y_P^i = j\} \log \frac{e^{\theta^{jT} \xi_P^i}}{\sum_{l=1}^{C} e^{\theta^{lT} \xi_P^i}} \tag{9}$$

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 (using Eq. (5)). 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 and κ_P be the probability density functions (*PDFs*) of q_S and q_P respectively, where q_S and q_P are the flattened [60] feature vectors z_S and $\tilde{z}$, respectively. The third term in Eq. (7) is the normalized KL-divergence between κ_S (feature distribution of gallery

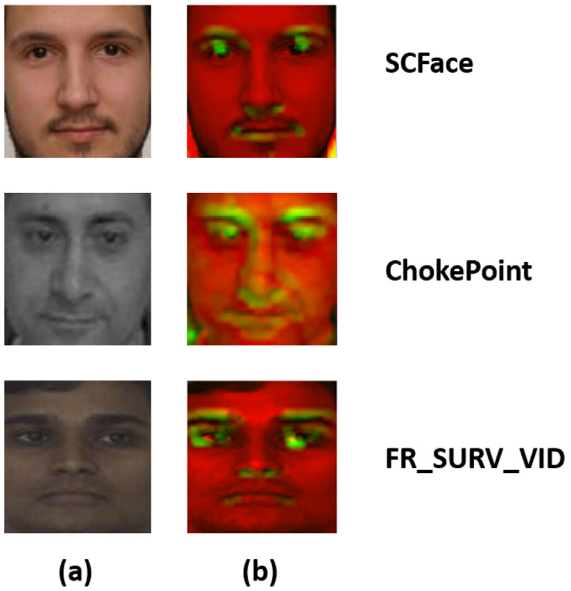


Fig. 5. For three real-world degraded datasets: (a) The input gallery images, (b) The output showing the selected activation of one of the filters at *CONV20* layer of BCM unit (see Fig. 3) after stage 1 of training. The green shaded areas show the discriminative areas of the face, while the red shades indicate the non-discriminative areas (best viewed in color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

samples) and κ_P (feature distributions of the transformed target probe samples). Thus, the third term can be expressed as:

$$\mathcal{T}(\kappa_S, \kappa_P) = \frac{KLD(\kappa_S, \kappa_P) + KLD(\kappa_P, \kappa_S)}{2 \times \max(KLD(\kappa_S, \kappa_P), KLD(\kappa_P, \kappa_S))} \tag{10}$$

where $KLD(T, S)$ is defined in Eq. (11) based on the KLD (Kullbeck–Leibler Divergence) measure [66], given as:

$$KLD(T, S) = \sum_{x_S \in D_S, x_P \in P_T} T(x_P) \log \frac{T(x_P)}{S(x_S)} \tag{11}$$

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. Minimization of this term in the objective reduces the gap between the gallery and the probe samples, as in DA. Authors of [65] did not use the symmetrical normalized form as in Eq. (10).

The last (fourth) term in the optimization function gives the normalized mutual variation of information (MVI), which is a metric modified from that used in [67]. The normalized MVI between q_S and q_P is given by:

$$MVI(q_S, q_P) = \frac{(H(q_S, q_P) - I(q_S; q_P))I(q_S; q_P)}{H(q_S, q_P)\sqrt{H(q_S)H(q_P)}} \tag{12}$$

where, $I(\cdot, \cdot)$ is the mutual information function, defined as:

$$I(q_S; q_P) = \sum_{a \in q_S, b \in q_P} p(a, b) \log \frac{p(a, b)}{p(a)p(b)} \tag{13}$$

$H(\cdot)$ is the entropy function, given as:

$$H(q_S) = \sum_{a \in q_S} p(a) \log p(a) \tag{14}$$

and $H(\cdot, \cdot)$ is the joint entropy, defined as:

$$H(q_S, q_P) = \sum_{a \in q_S, b \in q_P} p(a, b) \log p(a, b) \tag{15}$$

such that, $p(\cdot)$ is the probability, and $p(\cdot, \cdot)$ is the joint probability. Regularization of the TM parameters (as done in [65]) is implicitly incorporated in the *Tensorflow* based implementation of TM.

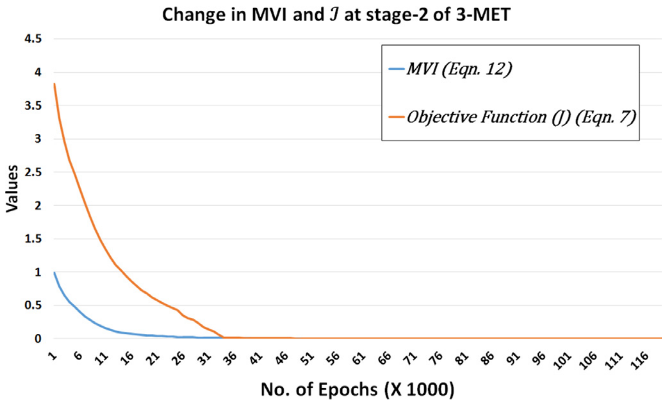


Fig. 6. Graphs showing the drop of MVI loss (Eq. (12)) and the objective function $\mathcal{J}$ (Eq. (7)) with increasing epochs, at stage 2 of 3-MET process for FR_SURV_VID [68] dataset, thus exhibiting the accomplishment of DA technique (best viewed in color).

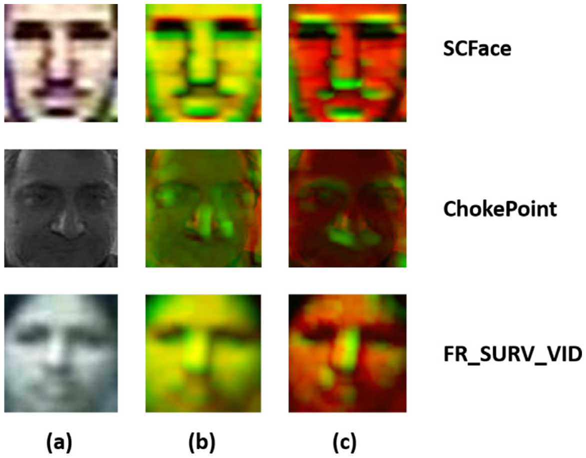


Fig. 7. (a) Input probe images, one from each 3 degraded face datasets, (b) The output map of the selected activation of one of the filters at the *CONV20* layer of the BCM, before stage 2 of training, (c) The same for one of the filters at *DCONV15* layer (see Fig. 3) of the TM unit, after stage 2 of 3-MET (color code as described in Fig. 5, best viewed in color).

The learning is done in batch mode here, with a significantly large set of combinations created by all possible pairings of gallery and few target probes (available for DA) used for training class-wise, based on the minimization of the objective $\mathcal{J}$ (Eq. (7)). This helps the auto-encoder in TM to learn the transformation of the features from probes to that of gallery, enabling DA for classification.

The TM thus learns to transform the target (probe) domain features to approximate the source (gallery) domain features, to be later fed to the LM module where a minor weight update of the LM unit is done at the stage 3 of 3-MET. The difference in the distributions between transformed probes and gallery samples is measured at stage 2 using the MVI loss criteria (Eq. (12)). Significant drops in the values of the objective function $\mathcal{J}$ is reflected in the graph shown in Fig. 6. This justifies our claim to bridge the distribution gap between D_S and P_T domains for DA, as the main purpose of stage 2 of 3-MET. The parameter updates of BCM and LM are frozen, while only the layers of the TM perform a parameter update using Eq. (6). The training procedure remains the same as described in Section 3.1.1.

Fig. 7(a) shows a few probe (target) samples given as input to BCM, and (b) shows the corresponding outputs of filters in BCM trained at stage 1 on the gallery samples, while (c) shows the corresponding outputs of the filters of TM after stage 2 of 3-MET. Bet-

Table 1
The number of samples (including real-world and the synthetic set obtained using data augmentation), from each dataset used for experimentation. The set of target and test probes never overlap (are distinct).

Datasets	Training Gallery (D_s)	Target set	Testing Probes (D_t)
SCFace [1]	2000	200	3000
ChokePoint [2]	5000	1000	10000
FR_SURV_VID	4000	800	10000
TIP-D [8]	1500	1000	4500
FERET [9]	10000	1500	5000
PIE [10]	8000	4000	12000
LFW-S [11]	2000	1500	9000

ter discrimination is visible at the salient (selective) green labeled areas in Fig. 7(c) which signifies greater activation at these areas of the face. The TM 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 [6].

3.2.3. Stage 3 (for fine-tuning the LM)

Although the stage 2 of the 3-MET reduces the gap between the source and the target domains considerably, the LM unit was however tuned at stage 1 to classify based on only the gallery samples. There is a minor disparity in the distributions of the gallery (source) and transformed probe (target) domains (output of TM after stage 2). To overcome this minor disparity, we fine-tune the pre-trained LM unit using the limited set (P_T) of target samples (see Fig. 4(c)) as used for DA at stage 2. Fine-tuning of parameters does not involve a major update of weights in the LM unit. Hence, the training of the LM unit can be carried on the limited set of target (P_T) domain samples. The parameters in the BCM and TM units are frozen at this stage, and act as feature extractor and DA processing blocks. The overall deep-DA trained model obtained at the end of stage 3 is used for testing with samples from D_T. The number of training, adaptation and testing samples used (per subject) for different datasets are as mentioned in Table 1. Detailed discussions on datasets used, pre-processing, evaluation protocol and performance analysis are discussed in the following sections.

4. Datasets and pre-processing

The proposed technique is evaluated over 4 real-world degraded face datasets and 3 synthetically degraded large scale real-world datasets. We compare the performance of our method with many state-of-the-art methods based on transfer learning and DL techniques used recently for FR.

The SCFace dataset [1] (a standard dataset for evaluating FR with degraded probe samples, 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 [2] 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 low-resolution 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 an average of 500 face images per subject in training and 2500 in the testing pool. The datasets contain 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 [68], 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.

The TIP-Dataset (TIP-D) proposed by Punnapurath et al. [8] consists of 50 subjects, consisting of a single mugshot per subject in the gallery and an average of 44 probe images per subject, with different non-uniform motion blur. Hence this scenario is a single sample per person (SSPP) condition for experimentation. Though the image samples in the gallery are of crisp and high quality than the probe samples, the tightly cropped face images are available at a resolution of 64 × 64 pixels in the database. The probe images suffer from varying amounts of blur, variations in illumination and pose, and with small degree of occlusion and facial expression changes. Although the blur was predominantly due to camera shake, no restriction was imposed on the movement of the subjects during image capture, and, therefore, a subset of these images could possibly have both camera and object motion.

For experimentation on large scale datasets, image samples of three benchmark datasets for FR have been synthetically degraded, namely: FERET [9], PIE [10] and LFW-S [11]. The *fa* folder of FERET containing 1364 images of 994 individuals, is used as the gallery set; while the *fb* folder containing 1358 images of 993 individuals, blurred using Gaussian kernel of $\sigma = \{0, 2, 4, 8\}$ and size $4\sigma + 1$, is used as probe. The PIE dataset consists of 68 individuals. To study the effect of illumination and blur together we consider the given image with a frontal pose (c_{27}) and good illumination (f_{21}) as our gallery (hence, can also be referred as an SSPP scenario), while amongst the rest of the images in c_{27}, the Bad Illumination (BI) set as described in [8] consisting of f_{13}, f_{14}, f_{15}, f_{16}, f_{17} and f_{22}, are used as probes, which are blurred using Gaussian kernels of $\sigma = \{0, 2, 4, 8\}$ and size $4\sigma + 1$. The LFW [9] database contains 13,233 target face images. Some images contain more than one face, but it is the face that contains the central pixel of the image which is considered the defining face for the image. The database contains images of 5749 different individuals. Of these, 560 subjects having two or more images, with minimal occlusion and pose-variations in the database are used in our work. This subset of images used will henceforth be called as LFW-S. The images are available as 250 × 250 pixels JPEG images. All these images are cropped to a fixed size using Viola-Jones face detector [69]. Only one image is considered as the gallery image and the rest used as probe images. Data augmentation techniques [60,70,71] have been used to increase the dataset size for training/adaptation/fine-tuning. The probe images are blurred synthetically similar to that done for FERET and PIE databases. Test probes of FERET, PIE and LFW-S have minor variations in pose (tilt, out of plane rotation) and expression.

Most real-world face datasets provide few samples in the gallery for training shallow transfer learning algorithms [5,72]. 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 [60,70,71], 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).

4.1. Preparing the data for the task

The gallery and the probe images vastly differ in their quality, as shown earlier in Fig. 1. For the first 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 [68] dataset, making it the hardest of the lot. The gallery images have uniform background, but the probe samples suffer from background variations. We observed largely unsatisfactory performance of all large state-of-the-art CNN architectures, which 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. [73]. The tightly cropped face image eliminates any background information present in it. To cope with the low contrast and poor illumination setting present in the probe images of real-world degraded datasets, the tightly cropped face samples are passed through a contrast-stretching stage, using the Power Law Transformation [74]. The difference in resolution is overcome by applying a face hallucination technique on the probe images, proposed by Jin and Bougannis in [7]. The gallery samples are downsampled to match the resolution of the probe images. Further pre-processing of the gallery samples to match (minimal reduction of disparity in quality) the probe samples include 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 [75]. 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 (for adaptation) samples used for experimentations, where the total number of samples used for training, adaptation and testing incorporates those obtained by data augmentation [60,70,71] (*i.e.* additionally, several synthetic samples were generated). The limited number of labeled test (probe) samples are available as a small fraction of the overall probe samples, for training the TM, which do not overlap with test probes used for performance analysis. About $5-15\%$ of the real-world test (probe) samples (P_T) 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 (but, within a small range) the same does not significantly improve the performance of our method. This experimental condition was kept same, as done in most DA based applications [41,48,76] published in the recent past.

5. Experimental details and performance analysis

Experiments are performed on a machine with i7-6720K 3 GHz processor and dual Nvidia Titan X GPU, with 64 GB RAM. The implementations are all coded in *Keras* using *TensorFlow* backend. In most of the experiments we start with a learning rate 0.03 which is gradually reduced as we progress through the training process. With random initialization of weights the training phase runs on these GPUs for 70 to 100 hrs. The inputs were sent as normalized mini-batch of 200 for training. The input size to the network is $100 \times 100 \times 3$ (see Fig. 3). The values for α, β and γ (refer Eq. (7)) are empirically determined as 0.5, 1.25 and 0.8 respectively. Further details of each of the layers are given in Appendix A.

In the following, results of the performance analysis are discussed of our proposed transfer-CNN architecture, compared with that of several recently published CNN and DA (shallow) models. The rigorous set of experimentations are broadly divided into three categories and discussed, as: **A**. *Experimentation on Real-world Degraded Datasets* (Section 5.1); **B**. *Experimentation on large datasets with synthetic blur* (Section 5.2); and **C**. *Unbiased Training* (Section 5.3). In Section 5.1, we use three real-world degraded face datasets, viz. SCFace [1], ChokePoint [2] and FR_SURV_VID to exhibit the performance analysis. In Section 5.2, we report the accuracy of our method and compared that with the recent state-of-the-art techniques discussed earlier, using benchmark face datasets, viz. TIP-D [8], FERET [9], PIE [10] and LFW-S [11], where the probes are synthetically degraded using a uniform blur kernel. Finally, in Section 5.3, we design an unbiased setting of training, where two vast chimeric datasets are formed for training, while adapted and tested individually on the each of the other constituent datasets. Additional results of performance analysis under varied scenarios are reported in Appendices B–F..

5.1. Experimentation on real-world degraded datasets

Rigorous experimentations were carried on the three real-world degraded datasets.

The comparison of the performance of our proposed deep-DA technique with recent state-of-the-art techniques for three real-world degraded 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 $1-7$ of Table 2 are first pre-trained using the augmented gallery samples, and then the regressor (the higher *fc*-) layers (for details, see [49]) have been fine-tuned using the target probe samples used for DA, where the lower layers are kept unaltered. The number of parameters in the architecture and the mode of experimentations are kept the same as provided by the authors, for all the CNN-based models used for comparisons. We have ensured that uniform experimental conditions are maintained for all cases, keeping in mind that we do not modify the training process of existing methods as suggested by the respective authors. All the values in Table 2 are obtained after pre-processing the input samples, except that mentioned in *row* 15. All experimentations have been performed using the pre-processed image probe samples in order to enhance the clarity of the highly degraded probe images.

FaceNet [13] performs the best (see row 5) among these seven CNN-based methods, which uses triplet-loss function for learning, where the triplets are drawn online. The naive method (row 8) executes only the stage 1 of training, using a concatenation of the target and gallery samples as the training set, and then testing with the probes. This involves BCM and LM units, with the TM fixed, *i.e.* TM acts as an Identity layer, which transfers the data directly from the BCM to the LM, without any modification. The FV_DCNN method [77] uses a Fisher vector encoded Deep convolutional network, where the FV features are adapted using EDA [5]. SML_MFKC [48] is a very recent shallow DA technique earlier proposed by us, that has shown encouraging results on low-resolution face datasets. The DeCAF$_6$ feature is adapted for DA using an eigen-domain based transformation in [5]. It is noticeable that the methods in rows $9-11$ (among $1-14$), which incorporate exclusive DA techniques for adapting hand-crafted or convolutional features to the target domain, fare quite better in general than the CNN-based methods in rows $1-7$ of Table 2. These CNN-based methods (rows $1-7$) suffer from the fact that the specific or higher layers do not easily adapt to the target domain. All other deep-transfer learning methods, in rows $12-14$ of Table 2 use source supervision [32] for adaptation, while variations of our proposed techniques in the last three rows $(15-17)$ of Table 2 rely on exclusive target supervision.

Our proposed method (deep-DA) achieves the best and a significantly higher accuracy (see last row) than all other competing

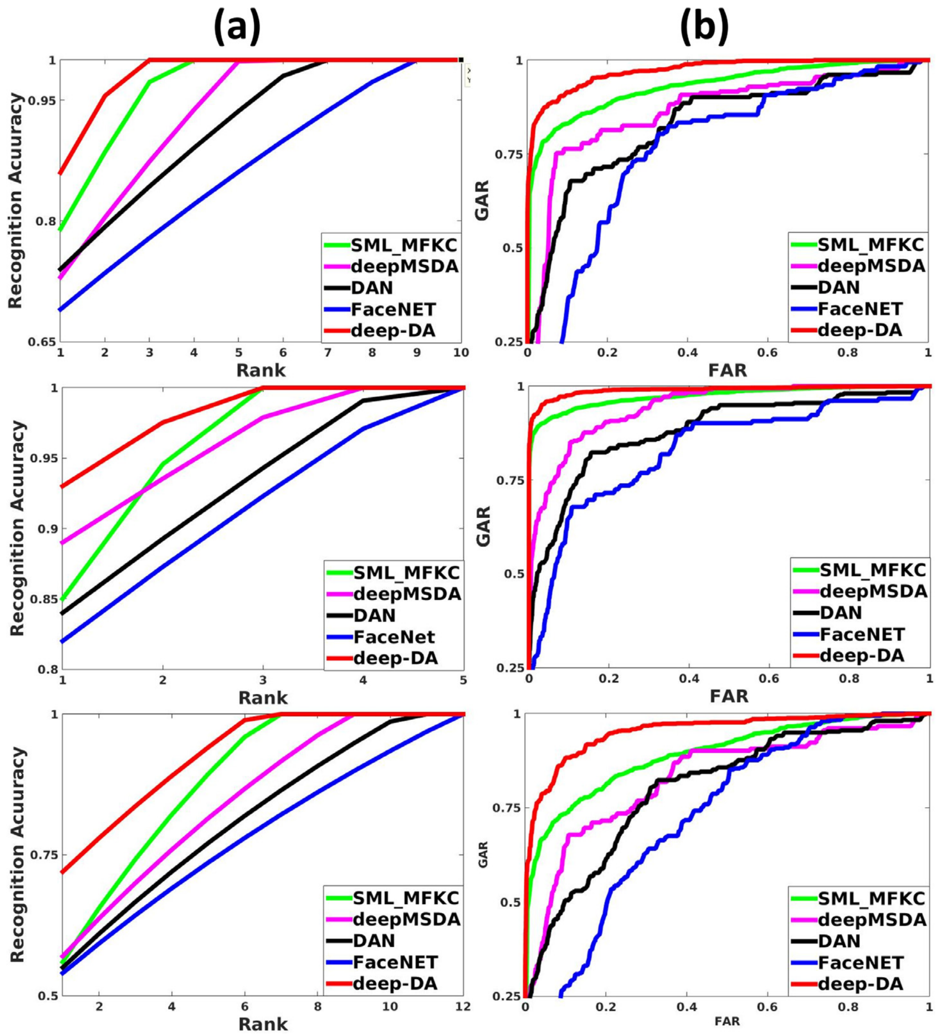


Fig. 8. (a) The CMC and (b) the ROC curves, showing the superiority of our proposed deep-DA model, on the three datasets, from top to bottom: SCFace, ChokePoint and FR_SURV_VID. The curves marked in red show the results of our proposed method. Performances are shown only for the next 5 best performing methods (from those in Table 2) used for comparison, as: SML_MFKC [48], deepMSDA [31], DAN [32] and FaceNet [13] (best viewed in color). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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 second best performing method at row 16, "K-NN at *FC*100", indicates that a K-NN classifier is used with the feature maps obtained at the *FC*100 block of the final layer of LM unit. The third best (on an average) performing method is our earlier work on a shallow technique, SML_MFKC [48], which also does a source to target transformation for DA. The deepMSDA [31] method also has a marginal stacked auto-encoder which reduces the noise and aliasing effect in the target images and is the fourth-best (when comparisons are done on an average over three datasets). A considerable degradation in the performance of deep-DA is reflected in row 15 of Table 2, when no pre-processing (as described in Section 4.1) is done on the data before training/testing. Similar fall in performances was also noticed for other methods (rows 1 – 14) when the data was used for training/testing/target-adaptation without pre-processing (discussed later in Appendix G). Implementations of all the CNN-based architectures using off-the-shelf and raw architectures were fine-tuned separately for each dataset partitions as per Table 1.

To strengthen our claim, we also provide the CMC and ROC curves for the three datasets. Fig. 8 shows the CMC and ROC plots for only the five best performing methods under comparison, for better visibility. The red curve in each sub-plot depicts the performance obtained by our deep-DA method, which is superior to all other competing techniques. The proposed method works better than most competing deep-CNN and DA methods, due to the adaptive denoising auto-encoder present in the TM, which has been exclusively trained (stage-2 of 3-MET) for DA to overcome the degradation from gallery to probe samples for all degraded FR datasets. The objective function in the TM has 3 different components to minimize: Reconstruction error, Softmax regression and KLD loss; which enables support for mapping (transfer) the face features across domains. The BCM and LM units of the proposed deep-DA architecture also support efficient feature extraction and classification respectively.

5.2. Experimentations on synthetically blurred datasets

Further experimentations have been carried on synthetically blurred face images of FERET, PIE, TIP-D and LFW-S datasets. The performance of our proposed architecture is compared with all other techniques as listed in Table 2. The results showing the effectiveness of our method, when compared with several state-of-the-art methods, is now reported in Table 3. Number of samples per subject for training, adaptation and testing are as given in Table 1.

Table 2
Rank-1 Recognition Rates for different methods over 3 real-world degraded face datasets. Results in bold exhibit the best performance, as appears for our proposed 3-MET method for deep-DA in the last three rows.

Sl.	Algorithm	SCface	CP	FSV
1	AlexNET [60]	48.25	73.24	35.89
2	DeepFace [12]	52.67	73.26	40.32
3	DeepID 2 [21]	44.65	78.12	43.02
4	DeepID 3 [22]	47.51	78.79	43.98
5	FaceNET [13]	69.08	82.72	54.68
6	VGG-19 [56]	59.35	80.14	46.32
7	DeepFace 2 [15]	53.62	75.86	48.71
8	Naive	35.24	61.59	18.62
9	FV_DCNN [77]	63.78	80.62	52.32
10	DeCAF$_6$ [30]	76.51	83.43	52.78
11	SML_MFKC [48]	79.86	85.59	58.31
12	DAN [32]	74.57	83.86	55.43
13	LSDA [78]	77.65	87.76	53.29
14	deepMSDA [31]	73.21	88.97	57.25
15	Deep-DA (without Pre-processing)	81.62	84.21	67.94
16	K-NN at *FC*100	79.65	88.74	71.25
17	Deep-DA (ours)	**87.23**	**94.57**	**74.11**

Table 3
Rank-1 Recognition Rate for different methods on 3 synthetically blurred (Gaussian PSF, with $\sigma = 4$) face datasets and TIP-D. Results in bold, exhibit the best performance.

Sl.	Algorithm	FERET	PIE	LFW-S	TIP-D
1	AlexNET [60]	68.37	71.45	71.38	61.96
2	DeepFace [12]	72.43	74.98	74.21	68.14
3	DeepID 2 [21]	64.32	68.86	70.09	62.97
4	DeepID 3 [22]	65.43	69.17	73.82	65.53
5	FaceNET [13]	76.64	79.12	78.36	79.18
6	VGG-19 [56]	75.15	77.34	77.28	77.25
7	DeepFace 2 [15]	73.52	75.67	74.33	76.36
8	Naive	51.76	64.88	69.67	42.49
9	FV_DCNN [77]	74.71	76.89	77.42	70.19
10	DeCAF$_6$ [30]	80.29	81.71	82.21	79.97
11	SML_MFKC [48]	82.65	83.54	84.51	81.76
12	DAN [32]	74.57	77.57	78.52	74.31
13	LSDA [78]	76.32	80.61	81.31	76.28
14	deepMSDA [31]	81.11	82.45	82.34	78.73
15	K-NN at *FC*100	83.88	83.96	88.13	84.21
16	Deep-DA (ours)	**84.42**	**86.02**	**89.91**	**84.19**

Details of experimentations as discussed in Section 5.1, remains unaltered for this part of the work too. However, the results are reported separately here, since the probes of these four datasets appear degraded only due to blur (non-uniform motion blur in case of TIP-D and synthetically fused Gaussian PSF blur, $\sigma = 4$ for the other three). FERET, PIE and LFW-S do not suffer from other degradations (as like the real-world degraded datasets) mentioned previously, such as low resolution, illumination and contrast, noise, or aliasing. For large-scale FERET [9], PIE [10] and LFW-S [11] (as well as TIP-D) datasets our proposed deep-DA technique performs the best providing acceptable levels of accuracies $\approx 83 - 90\%$, which proves the effectiveness of the transfer-CNN architecture.

In Table 4, we report the results of our method on TIP-D and PIE datasets with different levels of blur, where the performance is compared with several techniques as reported in [8,79]. Values reported in Table 4 are quoted from [8,79], except that in the last row. The probe set of PIE dataset is blurred with different values of the Gaussian parameter, $\sigma = \{0, 1, 2, 3\}$ of the PSF (kernel). In all the cases we can see that our method outperforms the competing methods by a considerable margin, specifically for larger values of σ.

Table 4
Performance using Rank-1 Recognition rate for increasing blur over PIE and TIP-D datasets. Results of other methods are quoted only for the best performing ones from [8,79]. '–' indicates unavailability of data.

	PIE				TIP-D
	Kernel Size (σ)				
	0	1.0	2.0	3.0	
rIRBF	95.10	92.70	88.20	81.36	–
LPQ	99.10	96.08	73.04	27.70	–
MOBILAP [8]	–	–	–	–	76.27
Deep-DA (ours)	**99.81**	**99.12**	**95.24**	**84.51**	**84.19**

5.3. Unbiased Training with very large chimeric datasets

It is necessary, to ascertain if a pre-learned FR system trained with a large dataset and then fine-tuned on the environment (illumination, blur, etc.) with limited samples [80,81], performed equivalently or better than an exclusive training done with gallery samples on-site. 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. For this purpose, two chimeric datasets are formed as gallery for training, as given in the 2*nd* column of Table 5, which can be considered as large unbiased datasets not overfitted to any particular environment of acquisition. This is to ascertain if exclusive training by a gallery of a particular dataset (biased training) is more effective than doing the same thing on a large chimeric dataset (unbiased training). Furthermore, we have also trained our deep-DA network (stage-1) with a large public dataset MS-CELEB [82] and fine-tuned it using the target datasets as mentioned in 3*rd* column, and the results are also reported in Table 5. In top-two cases of Table 5, the 2nd column shows the list of datasets used to form two large chimeric datasets: one using the three real-world degraded datasets providing 235 subjects; while the other combines this with four other benchmark face datasets, yielding a total of 1907 subjects. The number of probe samples used for adaptation and testing are obtained by combining them respectively from those as given in the 3rd and 4th columns of Table 1. In case of MS-CELEB [82], at the bottom set of rows (case C) in Table 5, the number of subjects is 99,892.

Performances in none of the scenarios are better than those in the last rows of Tables 2 and 3, when compared dataset-wise respectively. This simply indicates an obvious fact that the best performances for each dataset under experimentation are at the last rows of Tables 2 and 3, when the same dataset is used for training/testing/target-adaptation. Hence comparing results of Table 5 with Tables 2 and 3, one can identify that a large scale dataset used for training the transfer-CNN architecture does not help to improve the accuracy. Training exclusively with gallery samples of any one dataset provides better performance using than a large (chimeric or public) face dataset for the case of transfer-CNN or deep-DA. The amount of degradation in performance, when training is done using an unbiased, large chimeric dataset, range from $1 - 5\%$ (compare last rows of Tables 2 and 3, with the last column in Table 5, cases A and B). For the large public dataset MS-CELEB [82], the results reported in the bottom rows (case C) of Table 5 show a deterioration in performance of $4 - 6\%$, compared to that obtained using the two other chimeric datasets. This is again due to the limited number of target samples available for fine-tuning, in case of training with large public dataset (MS-CELEB). Since the target classes also form a small subset of the training subjects (classes) used for the other two large chimeric datasets (for cases A and B), performance is better. The worst performances are always obtained in cases where FSV is used for

Table 5
Rank-1 Recognition Rate (%) of proposed deep-DA when trained separately with two different large scale Chimeric Datasets. The degradation in performance range from 1 – 5%, when compared with the last rows of Tables 2 and 3.

Cases	Training dataset	Dataset for adaptation and testing	Recognition rate (%)
A	SCFace + FR_SURV_VID + ChokePoint	SCFace	83.07
		FR_SURV_VID	70.59
		ChokePoint	92.87
B	SCFace + FR_SURV_VID + ChokePoint + TIP-D + FERET + PIE +LFW-S	SCFace	81.59
		FR_SURV_VID	68.71
		ChokePoint	88.82
		TIP-D	81.47
		FERET	81.36
		PIE	82.79
		LFW-S	85.98
C	MS-CELEB [82]	SCFace	76.74
		FR_SURV_VID	64.02
		ChokePoint	84.29

testing/adaptation, reinstating that this is the hardest among all datasets used in degraded surveillance conditions, in spite of not being of large scale.

6. 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 transfer-CNN architecture, called deep-DA; (b) training done with a novel 3-stage Mutually Exclusive Training (3-MET) algorithm, incorporating an auto-encoder based objective minimization at stage-2; (b) rigorous experimentations performed on three real-world degraded face datasets, one real-world motion-blurred datasets and three synthetically blurred real-world benchmark face datasets, show the superiority of our method. The fine-tuning of the model at stage 2 of 3-MET boosts the performance of FR. Our method outperforms all other recent state-of-art techniques for the seven benchmark face datasets. One major drawback of the system, is the unavailability of a large amount of target data for training the TM unit for DA. Hence, the system can get trapped in a local optimal value. However, inspite of this our model provides the best performance across all datasets. Further experimentations may be done in future on cross-domain Object Recognition tasks and OCR using variations of architecture and training of our proposed model, to verify the generalization of the model for several other applications of deep-DA tasks. GAN, a recent model, can also be exploited for knowledge transfer (performing the task of DA) to solve FR in unconstrained conditions. Scalability of deep-DA may also be verified with large scale real-world degraded face datasets when available to researchers.

Appendix A. Structural details of the transfer-CNN architecture

The following Table A.6 gives different parameters for each of the layers used in the deep-DA architecture (see Fig. 3).

Appendix B. Cross-dataset adaptation on degraded 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 3-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 B.7. Performance appears as a mixed bag. It appears that the FSV dataset is the most toughest dataset to adapt or learn (when comparing row-wise average performances; also see lower rates at the last column of Table B.7). ChokePoint (CP) seems to be the most 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

Table A.6
The structural details of the proposed transfer-CNN architecture (see Fig. 3). The ReLU activation layer is omitted as the size-in and size-out are same.

Layer	Size-in	Size-out	Kernel
CONV10	100 × 100 × 3	100 × 100 × 10	3 × 3
POOL	100 × 100 × 10	34 × 34 × 10	3 × 3
CONV15	34 × 34 × 10	34 × 34 × 15	3 × 3
NORM	34 × 34 × 15	34 × 34 × 15	–
CONV20	34 × 34 × 15	34 × 34 × 20	3 × 3
POOL	34 × 34 × 20	12 × 12 × 20	3 × 3
CONV15	12 × 12 × 20	12 × 12 × 15	3 × 3
CONV15	12 × 12 × 15	12 × 12 × 15	1 × 1
POOL	12 × 12 × 15	6 × 6 × 15	2 × 2
CONV10	6 × 6 × 15	6 × 6 × 10	3 × 3
NORM	6 × 6 × 10	6 × 6 × 10	–
POOL	6 × 6 × 10	3 × 3 × 10	2 × 2
FC2048	90	2048	–
FC512	2048	512	–
DROPOUT	512	512	0.5
FC512	512	2048	–
FC2048	2048	90	–
UNPOOL	3 × 3 × 10	6 × 6 × 10	2 × 2
NORM	6 × 6 × 10	6 × 6 × 10	–
DCONV10	6 × 6 × 10	6 × 6 × 15	3 × 3
UNPOOL	6 × 6 × 15	12 × 12 × 15	2 × 2
DCONV15	12 × 12 × 15	12 × 12 × 15	1 × 1
DCONV15	12 × 12 × 15	12 × 12 × 20	3 × 3
FC4096	24880	4096	–
FC512	4096	512	–
DROPOUT	512	512	0.5
FC100	512	100	–
FC51	100	51	–
LSM	51	51	–

Table B.7
Rank-1 recognition rate (in %) of deep-DA for cross-dataset adaptation.

Sl.	Training	Adaptation	Test Probes		
			SCface [1]	CP [2]	FSV
1	SCFace	CP	70.73	72.64	–
2	SCFace	FSV	75.98	–	61.67
3	CP	SCFace	72.03	79.58	–
4	CP	FSV	–	79.13	59.82
5	FSV	SCFace	**78.69**	–	**67.06**
6	FSV	CP	–	**80.96**	63.75

Table C.8
Results showing the performance of FR for different statistical classifiers with CNN features, for the three degraded face datasets. Best results are in bold (identical to row 16 of Table 2).

Sl.	Method/Classifier	Features	Test Probes		
			SCface	CP	FSV
1	KNN	*FC*100 (stage-2)	**79.65**	**88.74**	**71.25**
2	EDA [5]	*FC*100 (stage-1)	74.52	78.68	63.91
3	SML_MFKC [48]	*FC*100 (stage-1)	75.08	83.25	64.63
4	SVM	*FC*100 (stage-2)	77.48	84.94	68.01

Table D.9
Rank-1 Recognition rate (in %) for the proposed deep-DA model with off-frontal face samples, with PIE and FERET datasets.

Dataset	Rank-1 Recognition Rate (%)
PIE	61.74
FERET	63.59

Table D.10
The number of off-frontal samples (real-world and synthetic using data augmentation) per subject, used for experimentation. The set of target and test probes never overlap.

Datasets	Training (Gallery)	Target	Testing (Probes)
FERET	20000	2000	10000
PIE	15000	5000	18000

Table E.11
Rank-1 Recognition rates (in %) for the proposed method, with increase in percentage of probe images used in deep-DA, for FR_SURV_VID dataset.

Amount(%) of probe samples	Rank-1 Recognition Rate (%)
10	71.93
20	74.07
30	75.09
40	76.98
50	82.87

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.

Appendix C. Testing using the features obtained from *fc*-layers of the LM unit (see Fig. 3)

Researchers in the recent past [30,77] have used feature vectors from the last *fc* layer for recognition/categorization. In this subset of experiments of deep-DA, features were obtained from the *FC*100 layer of LM unit in the proposed transfer-CNN architecture, but the identification is done using conventional statistical classifiers. We verified the performance using three different classifiers. In one case, KNN classifier was used for training, with the features extracted from *FC*100 layer after stage 3 of 3-MET method. The result showing the performance of KNN appears in the first row of Table C.8. Middle two rows of Table C.8 show results of experiments performed using two DA techniques [5,48], on the 100*D* feature vector obtained from *FC*100 layer of stage 1. Both DA methods (EDA [5] and SML_MFKC [48]) transform the source domain features to the target domain using shallow DA techniques. The final layers of the Linear module are unfrozen to match the number of classes in the target dataset. Overall, the first row of Table C.8 (when using KNN classifier) shows the best performing results, where the domain-adapted CNN features obtained at stage 3 marginally boosts the performance compared to two of our recently published shallow methods. This best result is identical to that given in Table 2, row 16.

Appendix D. Results of deep-DA using off-frontal samples

We have further verified the performance of our proposed model using off-frontal samples for training, keeping the same experimental conditions as described in Section 5. The training and test samples are both off-frontal. Since the off-frontal samples are only available in PIE [10] and FERET [9] datasets, the results reported in Table D.9 show the performance of deep-DA only for both these datasets. Results show a significant drop (compare Tables D.9 and 3) in accuracy, leading us to believe that our proposed method is well-suited for near-frontal samples. Table D.10 shows the number of samples (overall) used per subject for the two datasets, for performance analysis using off-frontal face images.

Appendix E. Effect of target sample size used for DA

Conventional DA based classification assumes that a small percentage of probe data is available during training, for the learning process to acquire minimal knowledge of the disparity in distributions between source (gallery) and target (probe) domains. To understand the impact of increasing the percentage of probe data available for training, we have conducted deep-DA training using 3-MET (our proposed model) with increasing levels ($10-50\%$) of samples (probes) available for training. Results are shown in Table E.11 where the recognition accuracy increases at a marginal rate with the rise in percentage of probe data available for DA. It's only appreciably good when 50% of probes are used for DA. This results in a substantial reduction in the number of probes used for testing, making the experimental setup lighter for any trained system.

Appendix F. Discussion and analysis on fine-tuning for deep-DA

Fine-tuning of a model refers to the re-training of a pre-trained model [83]. In DA, fine-tuning is used when a pre-trained model is updated using data with a different distribution. In most cases, fine-tuning a trained model update only a subset of its parameters. The popular fine-tuning framework [84] either takes the output of the last layer of the network as a feature and performs additional training (usual approach) for the new tasks, or performs fine-tuning on the entire original network without dropping the original objective function [83]. Fine-tuning of the parameters for adaptation over existing CNN methods listed in Table 2 (rows 1–7), other than our deep-DA, shallow and naive methods, has been performed as follows: (a) initially, pre-training using only the gallery samples (see Table 1, 2nd column); and then (b) fine-tune the *fc*- layers of this pre-trained model using a few target samples (Table 1, 3rd column) to adapt to the target domain, where all the other (spatial convolutional) layers are kept frozen. We will refer to this usual approach (two stages) of pre-training and fine-tuning [83,84] as the ’mode 1’ scenario. In mode-1 of training (used for methods in rows $1-7$ of Tables 2 and 3), the BCM and LM units are first ’pre-trained’ using the gallery samples at stage-1, where the TM is kept inactive (acts as fixed Identity layers or bypassed). Then, at the stage-2 of training in mode-1, considered as ’fine-tuning’, weights are only updated on the *fc*-layers in LM using test probes. Hence, the mode-1 of training can be considered as the operation of our proposed transfer-CNN architecture with the operation of TM completely suppressed. For our deep-DA model (in

Table F.12
Results of enhancement (boost) in Rank-1 recognition rate (%) of mode 2 (3-MET) over mode 1 scenario (baseline strategies), with fine-tuning of the model across (the three degraded face) databases. All gallery and probe faces have been pre-processed in this case.

Trained model (Stage 1)	Fine-tuning (both modes)	Dataset for testing	Enhancement of accuracy (%)
SCFace	SCFace	SCFace	**15.23**
SCFace	CP	SCFace	4.85
CP	SCFace	SCFace	3.68
SCFace	FSV	SCFace	10.42
FSV	SCFace	SCFace	9.74
CP	CP	CP	**12.96**
SCFace	CP	CP	5.71
CP	SCFace	CP	8.62
CP	FSV	CP	9.05
FSV	CP	CP	10.18
FSV	FSV	FSV	**17.85**
SCFace	FSV	FSV	7.26
CP	FSV	FSV	9.27
FSV	SCFace	FSV	8.97
FSV	CP	FSV	10.74

rows 15 – 17 of Table 2), our proposed 3-MET process described in Section 3.2, will be referred here as ’mode 2’ of fine-tuning.

Methods in rows 8 – 14 of Table 2 give the performances of methods including deep transfer networks, shallow DA classifiers, naive as well as CNN-feature based statistical classification. Application of either ’mode 1’ or ’mode 2’ of fine-tuning would have changed the basic protocol of training as specified by the authors in their work in these specific cases and hence avoided.

F1. Performance boost by fine-tuning of TM at stage 2 of 3-MET

This sub-section presents results of the improvement in performance of our proposed deep-DA method involving ’mode 2’ of fine-tuning (as 3-MET) over ’mode 1’ (baseline strategies), when exposed to different datasets. Note that for results reported for methods in rows 1 – 7 of Table 2, only the ’mode 1’ of fine-tuning has been used.

This experimental setup also illustrates the significance of 3-MET process to help our proposed deep-DA model to adapt across datasets or different distributions of the same dataset (gallery vs probes). The experiments are conducted in two modes of fine-tuning for adaptation, for our proposed deep-DA model. As mentioned earlier, in mode 1 the convolutional layers (in BCM) pre-trained using gallery samples are later frozen when only the higher fc-layers (in LM) are fine-tuned using a few target samples (used for DA) from the dataset as mentioned in the 2*nd* column of Table F.12. Whereas, in mode 2, the proposed 3-MET protocol is followed, as discussed in Section 3.2.

The enhancement in recognition accuracy for mode 2 of fine-tuning w.r.t. that for mode 1 is reported in Table F.12. The datasets used for training at stages 1 and 2 (see column 1 of Table F.12), fine-tuning (see column 2 of Table F.12) for both modes of training and testing (see column 3 of Table F.12) are varied, to observe the benefit of training all the three stages of 3-MET process (also referred as mode 2 of fine-tuning). The testing is performed separately for both modes of the fine-tuned models, pre-trained using the dataset as mentioned in Table F.12 (first column). While selecting the combinations of three datasets for training, fine tuning and testing, a constraint is ensured that the testing dataset is identical to one of those used for either training and fine tuning. The accuracies reported in all the experimentations use the Rank-1 Recognition Rate. The results of enhancement in this performance of mode 2 over mode 1 are reported in the last column of Table F.12.

All results in Table F.12 show very clear improvement brought by mode-2 over mode-1 training operations. In other words, this improvement in performance is clearly due to TM (inactive for mode-1), which is exclusively fine-tuned during training at stage-2 of the proposed 3-MET (Section 3.1.2) process. The final layers of LM are unfrozen in both these modes to match the number of classes in the target dataset. Accuracy of mode 2 is consistently better than that of mode 1, exhibiting the importance of our proposed 3-MET process. In Table F.12, the values in bold exhibit that the best results are achieved only when the same dataset (for obvious reasons) is used in all three cases of training, fine-tuning and testing (*i.e.* not cross-database, but gallery and probes of the same dataset). Interestingly over the three uniform conditions presented (see results in bold) in Table F.12, the best improvement in the performance accuracy due to fine-tuning (mode-2 over mode-1) appears for the case of FSV dataset, which is the toughest (see consistently least performance over all methods, in Tables 2, 5, C.8, F.12 and G.13) among the three real-world degraded face datasets.

Table G.13
Rank-1 Recognition Rates for different methods without pre-processing, over 3 real-world degraded face datasets. Results in bold, exhibit the best performances.

Sl.	Algorithm	SCface	CP	FSV
1	AlexNET [60]	29.46	58.31	22.16
2	DeepFace [12]	39.68	56.27	28.51
3	DeepID 2 [21]	27.81	60.97	26.48
4	DeepID 3 [22]	31.42	63.76	28.29
5	FaceNET [13]	51.38	67.56	42.83
6	VGG-19 [56]	45.68	64.19	32.54
7	DeepFace 2 [15]	41.18	60.98	37.29
8	Naive	30.68	54.32	15.43
9	FV_DCNN [77]	56.81	68.29	38.39
10	DeCAF$_6$ [30]	61.34	59.94	30.67
11	SML_MFKC [48]	69.85	64.27	48.59
12	DAN [32]	58.44	70.24	39.73
13	LSDA [78]	62.65	69.38	36.85
14	deepMSDA [31]	59.24	70.92	46.51
15	K-NN at *FC*100	75.16	76.68	66.27
16	Deep-DA (ours)	**81.35**	**84.97**	**68.09**

Appendix G. Performance enhancements due to TM unit *vs.* pre-processing of images

Results of all the competing methods, when used without pre-processing of face samples, are shown in Table G.13. When compared with Table 2, Table G.13 shows a considerable degradation in performance, when pre-processing is not applied on the gallery and probe samples. Although deep-DA performs the best (shown in last row of Table G.13), the accuracy values are much lower than that in Table 2. This highlights the importance of pre-processing

Table G.14
Comparison of enhancement in accuracy due to TM (for DA) with or without pre-processing and vice-versa. The values of performance enhancements are obtained as: **Col. 2**: mode-2 (3-MET) over mode-1 (with pre-processing of faces), where values are directly obtained from Table F.12 (rows 1, 6 and 11); **Col. 3**: mode-2 over mode-1 (without pre-processing); **Col. 4**: 3-MET (mode-2) with pre-processing over 3-MET without pre-processing. Values are obtained as {Table 2 (row 17) – Table G.13 (row 15)}; and **Col. 5**: mode-1 with pre-processing over mode-1 without pre-processing.

Degraded Face Datasets	Enhancement of Accuracy due to			
	TM (for DA)		Pre-processing	
	With Pre-processing	*Without Pre-processing*	*With TM*	*Without TM*
SCFace	15.64	25.52	5.26	2.36
ChokePoint	12.42	21.28	9.78	5.09
FR_SURV_VID	18.23	31.24	5.81	1.79

the data in case of low-resolution FR, with deep-DA, deep-CNN and shallow (DA) methods.

Next, we show the explicit comparison of performance enhancements due to the TM unit (for DA) *vs.* pre-processing of face image samples. Table G.14 provides the performance enhancements under four different conditions. Considering the operations being compared as: (a) TM unit (for DA) vs. (b) pre-processing of face image samples, we observe the performance enhancements due to the scenarios as: (i) *(a) with (b)*; (ii) *(a) without (b)*; (iii) *(b) with (a)* and finally (iv) *(b) without (a)*. This helps to explicitly compare their relative significance (*(a) vs. (b)*) for enhancement of performance. Given that the values in 4 columns of Table G.14 reflect performance enhancements of these proposed system under the four different conditions, it helps to compare the relative importance (significance) of pre-processing vs. TM operations for DA. A global overview of the table shows that values in the left two columns (2 and 3) are much larger than those on the right two (4 and 5). This indicates the fact that TM provides larger enhancement in performance than the pre-processing task (although the latter is useful too). A local view of the columns of Table G.14 highlights the same explicitly. Comparing values in column 3 (enhancement solely due to TM) with column 5 (enhancement solely due to pre-processing), reveal that the performance enhancements are of several orders in magnitude larger when TM operation is solely used than that due to pre-processing alone. Similar is the case between columns 2 and 4. Observe this in a complimentary manner – *i.e.* column 2 can also be visualized as that reflecting the fall (degradation) in performance when TM operation is suppressed with pre-processing being used, while that of column 4 exhibit the same when pre-processing is switched off but TM is active. Values (as fall in performance) in column 2 are higher than those in column 4, indicating the fact that TM operations are much more dearer to the success of deep-DA for FR than pre-processing of face images. Overall, the values in Table G.14 clearly indicate that the performance enhancement is larger due to TM than pre-processing of face probes. This helps to conclude that DA due to TM is mainly responsible for the enhancement of performance.

References

[1] M. Grgic, K. Delac, S. Grgic, Scface–surveillance cameras face database, Multim. Tools Appl. 51 (3) (2011) 863–879.

[2] Y. Wong, S. Chen, S. Mau, C. Sanderson, B.C. Lovell, Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) on Biometrics, 2011, pp. 74–81.

[3] R. Gopalan, R. Li, R. Chellappa, Domain adaptation for object recognition: an unsupervised approach, in: Proceedings of the International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 999–1006.

[4] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, in: Proceedings of the European Conference on Computer Vision (ECCV), 2010, pp. 213–226.

[5] S. Banerjee, S. Samanta, S. Das, Face recognition in surveillance conditions with bag-of-words, using unsupervised domain adaptation, in: Proceedings of the Indian Conference on Computer Vision Graphics and Image Processing (ICVGIP), 2014.

[6] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the International Conference on Machine learning (ICML), 2008.

[7] Y. Jin, C.-S. Bouganis, Robust multi-image based blind face hallucination, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5252–5260.

[8] A. Punnappurath, A.N. Rajagopalan, S. Taheri, R. Chellappa, G. Seetharaman, Face recognition across non-uniform motion blur, illumination, and pose, IEEE Trans. Image Process. 24 (7) (2015) 2067–2082.

[9] P.J. Phillips, H. Wechsler, J. Huang, P.J. Rauss, The FERET database and evaluation procedure for face-recognition algorithms, Image Vis. Comput. 16 (5) (1998) 295–306.

[10] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression (PIE) database, in: Proceedings of the International Conference on Automatic Face and Gesture Recognition (FG), IEEE, 2002, pp. 53–58.

[11] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report 07–49,, University of Massachusetts, Amherst, 2007.

[12] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: Proceedings of the British Machine Vision Conference, 1, 2015, p. 6.

[13] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815–823.

[14] Y. Sun, X. Wang, X. Tang, Deeply learned face representations are sparse, selective, and robust, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2892–2900.

[15] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1701–1708.

[16] H. Wang, Z. Li, X. Ji, Y. Wang, Face r-CNN, CoRR, 1706.01061. (2017).

[17] D. Wang, C. Otto, A.K. Jain, Face search at scale, IEEE Trans. Pattern Anal. Mach. Intel. 39 (6) (2017) 1122–1136.

[18] A.S. Georghiades, P.N. Belhumeur, D.J. Kriegman, From few to many: Illumination cone models for face recognition under variable lighting and pose, IEEE Trans. Pattern Anal. Mach. Intel. 23 (6) (2001) 643–660.

[19] R. Gross, I. Matthews, J. Cohn, T. Kanade, S. Baker, Multi-pie, Image Vis. Comput. 28 (5) (2010) 807–813.

[20] Z. Zhu, P. Luo, X. Wang, X. Tang, Recover canonical-view faces in the wild with deep neural networks, CoRR, 1404.3543. (2014).

[21] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1891–1898.

[22] Y. Sun, D. Liang, X. Wang, X. Tang, Deepid3: Face recognition with very deep neural networks, CoRR, 1502.00873. (2015).

[23] R. Ranjan, S. Sankaranarayanan, C.D. Castillo, R. Chellappa, An all-in-one convolutional neural network for face analysis, in: Proceedings of the International Conference on Automatic Face & Gesture Recognition (FG), IEEE, 2017, pp. 17–24.

[24] L. Wolf, T. Hassner, I. Maoz, Face recognition in unconstrained videos with matched background similarity, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 529–534.

[25] W. Zhao, R. Chellappa, Face Processing: Advanced Modeling and Methods, Academic Press, 2011.

[26] W. Zhao, R. Chellappa, P.J. Phillips, A. Rosenfeld, Face recognition: a literature survey, ACM Comput. Surv. 35 (4) (2003) 399–458.

[27] S.K. Zhou, R. Chellappa, W. Zhao, Unconstrained Face Recognition, 5, Springer Science & Business Media, 2006.

[28] X. Hu, S. Peng, L. Wang, Z. Yang, Z. Li, Surveillance video face recognition with single sample per person based on 3d modeling and blurring, Neurocomputing 235 (2017) 46–58.

[29] L. Qiang, W. Zhang, L.I. Hongliang, K.N. Ngan, Hybrid human detection and recognition in surveillance, Neurocomputing 194 (2016) 10–23.

[30] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: A deep convolutional activation feature for generic visual recognition., in: Pro-

ceedings of the International Conference on Machine Learning (ICML), 2014, pp. 647–655.

[31] M. Chen, K.Q. Weinberger, F. Sha, Y. Bengio, Marginalized denoising auto-encoders for nonlinear representations., in: Proceedings of the International Conference on Machine Learning (ICML), 2014, pp. 1476–1484.

[32] M. Long, J. Wang, Learning transferable features with deep adaptation networks, CoRR, 1502.02791. (2015).

[33] M. Ghifary, W.B. Kleijn, M. Zhang, D. Balduzzi, W. Li, Deep reconstruction-classification networks for unsupervised domain adaptation, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2016, pp. 597–613.

[34] G. Alain, Y. Bengio, What regularized auto-encoders learn from the data-generating distribution., J. Mach. Learn. Res. 15 (1) (2014) 3563–3593.

[35] Y. Bengio, E. Laufer, G. Alain, J. Yosinski, Deep generative stochastic networks trainable by backprop, in: Proceedings of the International Conference on Machine Learning (ICML), 2014, pp. 226–234.

[36] V.M. Patel, R. Gopalan, R. Li, R. Chellappa, Visual domain adaptation: A survey of recent advances, IEEE Signal Process. Mag. 32 (3) (2015) 53–69.

[37] S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis, IEEE Trans. Neural Netw. 22 (2) (2011) 199–210.

[38] B. Gong, K. Grauman, F. Sha, Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation., in: Proceedings of the International Conference on Machine Learning (ICML), 2013, pp. 222–230.

[39] K. Zhang, B. Schölkopf, K. Muandet, Z. Wang, Domain adaptation under target and conditional shift., in: Proceedings of the International Conference on Machine Learning (ICML), 2013, pp. 819–827.

[40] X. Wang, J. Schneider, Flexible transfer learning under support and model shift, in: Advances in Neural Information Processing Systems), 2014, pp. 1898–1906.

[41] B. Gong, Y. Shi, F. Sha, K. Grauman, Geodesic flow kernel for unsupervised domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 2066–2073.

[42] M. Baktashmotlagh, M.T. Harandi, B.C. Lovell, M. Salzmann, Domain adaptation on the statistical manifold, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 2481–2488.

[43] M. Baktashmotlagh, M. Harandi, B. Lovell, M. Salzmann, Unsupervised domain adaptation by domain invariant projection, in: Proceedings of the International Conference on Computer Vision (ICCV), 2013, pp. 769–776.

[44] L. Duan, D. Xu, I. Tsang, Learning with augmented features for heterogeneous domain adaptation, in: Proceedings of the International Conference on Machine Learning (ICML), 2012, pp. 711–718.

[45] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (59) (2016) 1–35.

[46] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, CoRR, 1702.05464. (2017).

[47] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, D. Krishnan, Unsupervised pixel-level domain adaptation with generative adversarial networks, CoRR, 1612.05424. (2016).

[48] S. Banerjee, S. Das, Soft-margin learning for multiple feature-kernel combinations with domain adaptation, for recognition in surveillance face dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) on Biometrics, 2016, pp. 169–174.

[49] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? in: Advances in Neural Information Processing Systems (NIPS), 2014, pp. 3320–3328.

[50] L. Duan, D. Xu, I. Tsang, Learning with augmented features for heterogeneous domain adaptation, CoRR, 1206.4660. (2012).

[51] M.D. Zeiler, R. Fergus, Stochastic pooling for regularization of deep convolutional neural networks, CoRR, 1301.3557. (2013).

[52] D. Scherer, A. Müller, S. Behnke, Evaluation of pooling operations in convolutional architectures for object recognition, in: Proceedings of the International Conference on Artificial Neural Networks (ICANN), Springer, 2010, pp. 92–101.

[53] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, CoRR, 1502.03167. (2015).

[54] S. Arora, A. Bhaskara, R. Ge, T. Ma, Provable bounds for learning some deep representations., in: Proceedings of the International Conference on Machine Learning (ICML), 2014, pp. 584–592.

[55] Y. Bengio, et al., Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.

[56] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR, 1409.1556. (2014).

[57] M.D. Zeiler, G.W. Taylor, R. Fergus, Adaptive deconvolutional networks for mid and high level feature learning, in: Proceedings of the International Conference on Computer Vision (ICCV), 2011, pp. 2018–2025.

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

[59] D. Williams, G.E. Hinton, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.

[60] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems (NIPS), 2012, pp. 1097–1105.

[61] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2921–2929.

[62] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Proceedings of the European Conference on Computer Vision (ECCV), Springer, 2014, pp. 818–833.

[63] S. Sharma, R. Kiros, R. Salakhutdinov, Action recognition using visual attention, CoRR, 1511.04119. (2015).

[64] X. Glorot, A. Bordes, Y. Bengio, Domain adaptation for large-scale sentiment classification: a deep learning approach, in: Proceedings of the International Conference on Machine Learning (ICML), 2011, pp. 513–520.

[65] F. Zhuang, X. Cheng, P. Luo, S.J. Pan, Q. He, Supervised representation learning: transfer learning with deep autoencoders., in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2015, pp. 4119–4125.

[66] S. Kullback, R.A. Leibler, On information and sufficiency, Annals Math Stat 22 (1) (1951) 79–86.

[67] N.X. Vinh, J. Epps, J. Bailey, Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance, J. Mach. Learn. Res. 11 (Oct) (2010) 2837–2854.

[68] S. Rudrani, S. Das, Face recognition on low quality surveillance images, by compensating degradation, in: Proceedings of the International Conference Image Analysis and Recognition (ICIAR), 2011, pp. 212–221.

[69] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.

[70] D. Ciregan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3642–3649.

[71] P.Y. Simard, D. Steinkraus, J.C. Platt, Best practices for convolutional neural networks applied to visual document analysis., in: Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), volume 3, 2003, pp. 958–962.

[72] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.

[73] A. Asthana, S. Zafeiriou, S. Cheng, M. Pantic, Incremental face alignment in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1859–1866.

[74] H. Farid, Blind inverse gamma correction, IEEE Trans. Image Process. 10 (10) (2001) 1428–1433, doi:10.1109/83.951529.

[75] F. Juefei-Xu, M. Savvides, Encoding and decoding local binary patterns for harsh face illumination normalization, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2015, pp. 3220–3224.

[76] R. Gopalan, S. Taheri, P. Turaga, R. Chellappa, A blur-robust descriptor with applications to face recognition, IEEE Trans. Pattern Anal. Mach. Intel. 34 (6) (2012) 1220–1226.

[77] J.C. Chen, J. Zheng, V.M. Patel, R. Chellappa, Fisher vector encoded deep convolutional features for unconstrained face verification, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2016, pp. 2981–2985, doi:10.1109/ICIP.2016.7532906.

[78] J. Hoffman, S. Guadarrama, E.S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, K. Saenko, Lsda: Large scale detection through adaptation, in: Advances in Neural Information Processing Systems (NIPS), 2014, pp. 3536–3544.

[79] P. Vageeswaran, K. Mitra, R. Chellappa, Blur and illumination robust face recognition via set-theoretic characterization, IEEE Trans. Image Process. 22 (4) (2013) 1362–1372.

[80] S. Arya, N. Pratap, K. Bhatia, Future of face recognition: A review, Procedia Computer Science 58 (2015) 578–585. Second International Symposium on Computer Vision and the Internet (VisionNet15).

[81] H. Wechsler, J.P. Phillips, V. Bruce, F.F. Soulie, T.S. Huang, Face Recognition: From Theory to Applications, 163, Springer Science & Business Media, 2012.

[82] Y. Guo, L. Zhang, Y. Hu, X. He, J. Gao, Ms-celeb-1m: A dataset and benchmark for large-scale face recognition, in: Proceedings of the European Conference on Computer Vision, Springer, 2016, pp. 87–102.

[83] Q. Chen, J. Huang, R. Feris, L.M. Brown, J. Dong, S. Yan, Deep domain adaptation for describing people based on fine-grained clothing attributes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 5315–5324.

[84] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587.

Samik Banerjee received his Bachelors of Technology in Computer Science and Engineering from St. Thomas' College of Engineering and Technology, Kolkata, India in 2008, and Masters of Engineering in Computer Science and Engineering from Bengal Engineering and Science University, Shibpur, India in 2010. He has been a Ph.D. scholar in the Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India, since 2012. His research interests include computer vision, face recognition, deep learning, neural networks, machine learning and transfer learning.

Dr. Sukhendu Das is currently employed as a Professor in the Department of Computer Science and Engg., IIT Madras, Chennai, India. He completed his B.Tech degree from IIT Kharagpur in the Department of Electrical Engg. in 1985 and M. Tech Degree in the area of Computer Technology from IIT Delhi in 1987. He then obtained his Ph.D degree from IIT Kharagpur in 1993. His current areas of research interests are: Visual Perception, Computer Vision: Digital Image Processing and Pattern Recognition, Computer Graphics, Artificial Neural Networks, Computational Science and engineering, Soft Computing, Deep Learning and Computational brain modeling. Dr. Sukhendu Das has been a faculty of the Department of CS&E, IIT Madras, INDIA since 1989. He has worked as a visiting scientist in the University of Applied Sciences, Pforzheim, Germany, for post-doctoral research work, from Dec. 2001 till May 2003; and as a visiting fellow/scientist in the University of UWA, Perth, Australia, during June–Aug. 2006, and July–Sept. 2008. He has guided Six (currently guiding 2) Ph. D students, 26 (currently guiding 7) M.S., 42 M. Tech. (+ Dual) and 9 B. Tech students. He had completed several international and national sponsored projects and consultancies, both as principle and co-investigators. He has published more than 150 technical papers in international and national journals and conferences. He has reviewed several papers in international journals (IEEE, IET, Elsevier, Springer etc.) and chaired several sessions in conferences. He has received three (3) best papers and a best design contest award. Significant and novel technical contributions are: MSGAN for Video prediction; MST-CSS representation for CBVR tasks; SUBBAND face, Deep-DA or transfer-CNN, Eigen-domain transformation (EDT) and Eigen-scale space (ESS) representations for face-based biometry applications; Creation of an Outdoor Surveillance Face Database (support from MCIT, GOI) for biometry; and Manifold based alignment for optimization using Domain Adaptation, for applications in face, object and video categorization tasks.