Unsupervised Domain Adaptation using Maximum Mean Covariance Discrepancy and Variational Autoencoder

Fabian Barreto[1]
Department of
Electronics and Telecommunication
Xavier Institute of Engineering
Mumbai, India

Dr Jignesh Sarvaiya[2]
Department of Electronics,
Sardar Vallabhbhai
National Institute of Technology
Surat, India

Dr Suprava Patnaik[3]
School of Electronics,
Kalinga Institute of Industrial Technology
Bhubaneswar, India

Sushilkumar Yadav[4]
Jio Platforms Limited
Navi Mumbai, India

Abstract—**Face Recognition has progressed tremendously from its initial use of holistic learning models to using hand-crafted, shallow, and deep learning models. DeepFace, a nine-layer Deep Convolutional Neural Network (DCNN), reached near-human performance on unconstrained face recognition for the Labeled Faces in the Wild (LFW) dataset. These models performed very well on the benchmark datasets, but their performance sometimes deteriorated for real-world applications. The problem arose when there was a domain shift due to different distribution spaces of the training and testing models. Few researchers looked at Unsupervised Domain Adaptation (UDA) to find the domain-invariant feature spaces. They tried to minimize the domain discrepancy using a static loss of maximum mean discrepancy (MMD). From MMD, the researchers delved into the higher-order statistics of maximum covariance discrepancy (MCD). MMD and MCD were combined to get maximum mean and covariance discrepancy (MMCD), which captured more information than MMD alone. We use a Variational Autoencoder (VAE) with joint mean and covariance discrepancy to offer a solution for domain adaptation. The proposed MMCD-VAE model uses VAE to measure the discrepancy in the spread of variance around the mean value and uses MMCD to measure the directional discrepancy in the variance. Analysis was done using the TinyFace benchmark dataset and the Bollywood Celebrities dataset. Three objective image quality parameters, namely SSIM, pieAPP, and SIFT feature matching, demonstrate the superiority of MMCD-VAE over the conventional KL-VAE model. MMCD-VAE shows an 18 % improvement in SSIM and a remarkable improvement in the perceptual quality of the image over the conventional KL-VAE model.**

Keywords—Deep learning; domain adaptation; face recognition; maximum mean covariance discrepancy; transfer learning; variational autoencoders

I. Introduction

In the past decade, Face Recognition (FR) research has achieved high accuracy using Deep Learning (DL) approaches. It has matched that of the humans and even transcended it. Advances in DL have facilitated the growth of large training datasets required to implement DL algorithms effectively. Presently we have datasets that use large amounts of labeled data from the internet, consisting of face images in an unconstrained environment, with a marked diversity of ethnicity, gender, and age.

At times, in real-world applications, one notices a certain discrepancy. The target face image dataset is acquired in different settings compared to the source. There is a difference in the performance of a learned model on a source dataset and a target dataset. Also, in some applications, it is not possible to have large datasets from a particular domain to train a deep learning model. So can one borrow pre-trained models from similar domains? This can help to improve the learning process. However, the caveat is that the performance is boosted only for trained and tested datasets with identical data distributions.

It is interesting to understand the learning process between the deep networks and the human person in this context. The way that learning happens in deep networks and human persons is different. Humans learn from a limited set of labeled data. The other advantage humans possess is that they can generalize their learning and apply it to new conditions or situations.

The authors in [1] have shown the theoretical limitations on the performance by studying the error bounds for different source and target data distributions. The term “data shift”, as first used in 2009, in [2], is the change of distribution of features [3]. The change in the distributions is referred to as covariate shift in [4]. Even a Deep CNN can experience domain shift [5]. Domain Adaptation (DA) algorithms attempt to understand these different shifts in statistical distributions for adaptation in domains.

The paper is organized as follows. Section II presents a review of domain adaptation techniques. Section III describes the metrics for measuring distribution discrepancy. Section IV focuses on the deep domain adaptation for face recognition. Section V presents the proposed MMCD-VAE latent feature extraction model. Section VI elaborates the experimental results, and finally Section VII provides the conclusion and future work of this study.

II. A Review of Domain Adaptation

A. Domain Adaptation and Transfer Learning

The authors in their landmark paper [6] gave an overview of the Transfer Learning (TL) process, where they situated the DA task in the context of TL. Tasks were Inductive Transfer Learning, Transductive Transfer Learning or Unsupervised Transfer Learning based on label availability for the source and target domain. A summary is shown in Fig. 1 as in [6], which shows how DA is a subset of TL.

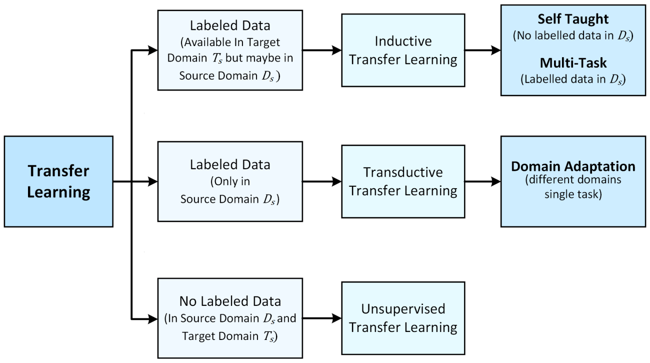
Transfer Learning
Labeled Data (Available In Target Domain T_s but maybe in Source Domain D_s)
Inductive Transfer Learning
Self Taught (No labelled data in D_s)
Multi-Task (Labelled data in D_s)
Labeled Data (Only in Source Domain D_s)
Transductive Transfer Learning
Domain Adaptation (different domains single task)
No Labeled Data (In Source Domain D_s and Target Domain T_s)
Unsupervised Transfer Learning

Fig. 1. The Relationship of Domain Adaptation to Transfer Learning [10].

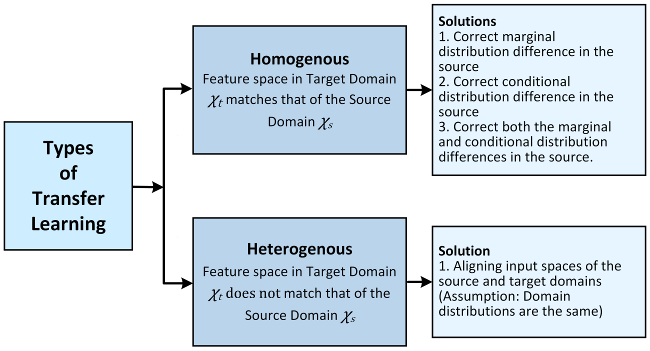
Types of Transfer Learning
Homogenous
Feature space in Target Domain $\mathcal{X}_t$ matches that of the Source Domain $\mathcal{X}_s$
Solutions
1. Correct marginal distribution difference in the source
2. Correct conditional distribution difference in the source
3. Correct both the marginal and conditional distribution differences in the source.
Heterogenous
Feature space in Target Domain $\mathcal{X}_t$ does not match that of the Source Domain $\mathcal{X}_s$
Solution
1. Aligning input spaces of the source and target domains (Assumption: Domain distributions are the same)

Fig. 2. Types of Transfer Learning.

The authors in [7] define TL in terms of the domains and the given tasks. They classify TL as being homogenous when the feature space is the same and heterogeneous when the feature spaces are different, as shown in Fig. 2. They also clarify that the domain adaptation process seeks to change a source domain to match more closely with the target domain. The terms supervised or unsupervised refer to the source domain availability of labeled data. And for the target domain, as informed or uninformed. A word of caution is also given on Negative transfer when the learned information detrimentally effects the target domain.

The authors in [8] elaborate on the transfer learning categories and present about forty representative approaches to transfer learning along with experimental verification. The broad categories are shown in Fig. 3.

The notations given in [6] and [7] are used to explain the concepts of DA.

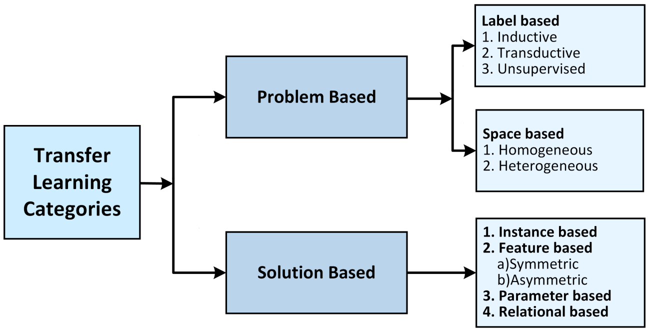

Fig. 3. Transfer Learning Categories.

Let the source domain labeled data be given by $\mathcal{D}_s = \{x_i^s, y_i^s\}_{i=1}^M$, with ith sample x_i^s, and label y_i^s. The number of source images is given by M.

Let target domain unlabeled data be given by $\mathcal{D}_t = \{x_i^t\}_{i-1}^N$, with ith sample x_j^t. The number of target images is given by N. The difference in data distributions as shown in Fig. 4, is given by $P(X_s, Y_s) \neq P(X_t, Y_t)$.

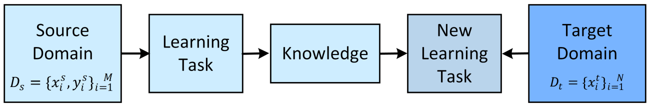

Fig. 4. A Simplified Transfer Learning Model for Domain Adaptation.

Many researchers have done surveys on TL [6], [7], [9], [10] [8] and DL [11], [12], [13], [14] and [15]. Beginning from Machine learning to Deep learning, the authors have methodically explained the nuanced terminology and clarified any inconsistencies in the terms that are used to explain the concepts of TL and DL.

III. Metrics for Measuring Distribution Discrepancy

A. Maximum Mean Discrepancy (MMD)

The distribution variations are found using metrics that measure distribution discrepancy. The ones often used are Kullback–Leibler divergence [16], the maximum mean discrepancy (MMD) [17], [18], the Bregman divergence [19], and the Wasserstein distance [20].

Among the most commonly used is MMD. It finds the measure between the mean of the two distributions into a reproducing kernel Hilbert space (RKHS). Maximum Mean Discrepancy (MMD) [17], [21], [22] is thus a distribution distance metric. The MMD [23] between two distributions s and t, is given by

$$L_M(s,t) = \sup_{\|\phi\|_{\mathcal{H}} \leq 1} \left\| E_{\mathrm{x}^s \sim s}\left[\phi\left(\mathrm{x}^s\right)\right] - E_{\mathrm{x}^t \sim \mathrm{t}}\left[\phi\left(\mathrm{x}^{\mathrm{t}}\right)\right] \right\|_{\mathcal{H}}^2 \quad (1)$$

Sup (“supremum”) is the largest, least upper bound (generalizations of “max”), E is the expectation of the distribution. ϕ maps original data to RKHS. The detailed proofs are given in [24].

B. Maximum Covariance Discrepancy (MCD)

The notations for MCD are taken from [24] where one can find the detailed proofs.

$$\begin{aligned} &\mathrm{MCD}[p, q, \mathcal{H}] \\ &\quad = \sup_{\|a\| \leq 1} \sum_{i,j \in I} a_{ij} \left(\operatorname{cov}\left[e_i(x), e_j(x)\right] - \operatorname{cov}\left[e_i(y), e_j(y)\right] \right) \end{aligned} \tag{2}$$

where $\mathcal{H}$ is RKHS over X and $\{e_i \mid i \in I\}$ is an orthogonal basis of $\mathcal{H}, \|a\|= \left(\sum_{i,j \in I} a_{ij}^2\right)^{1/2}$, with cov given by: $\operatorname{cov}\left[e_i(x), e_j(x)\right] = E_x\left[e_i(x) e_j(x)\right] - E_x\left[e_i(x)\right] E_x\left[e_j(x)\right]$.

C. Maximum Mean and Covariance Discrepancy (MMCD)

The authors in [24] have shown that the MMCD-based domain adaptation achieves better results for image classification. MMCD has both the first- and second-order statistical information in the RKHS. The notations for MMCD are taken from [24], where one can find detailed proofs.

$$\mathrm{MMCD}[p, q, \mathcal{H}] = \left(\|\mu[p] - \mu[q]\|_{\mathcal{H}}^2 + \beta \|C[p] - C[q]\|_{\mathrm{HS}}^2 \right)^{1/2} \tag{3}$$

where $\mu[p] = E_x[\phi(x)]$ and β, used to balance the MCD term, is a non-negative parameter, and C is a centered covariance operator. They show that MMD and MCD of MMCD measures the difference between means and covariances of the distributions with the degree d = 1 of the polynomial kernel.

IV. Deep Domain Adaptation for Face Recognition

The authors in [25], [26], [23] discuss the approaches and challenges to deep domain adaptation in the context of face recognition which indeed is a challenging task. In real-life face recognition applications, there are domain shifts due to changing conditions, like background, location, change of pose, occlusion, illumination, and other factors.

In [25], the authors have used the TaoMM dataset created using face images of Chinese fashion models. They combined the CASIA-WebFace [27] and VGGFace-Good [28] datasets and used about 1.3 million images to train their model. They also trained the model on their TaoMM dataset. These trained models were then tested on the LFW dataset [29] which has a different distribution than the TaoMM dataset. The learned weights of labeled data are transferred to initialize the training model. They also refine all weights using face verification loss in an end-to-end framework.

Their system architecture consisted of a modified inception-v2 [30] model that enhanced training using Stochastic Gradient Descent. They used an NVIDIA GTX TITAN X GPU and pre-trained for 25 epochs that lasted 89.4 hours with a learning rate of 0.2 and decay half for every five epochs. A learning rate of 0.04 and decay half for every ten epochs was used and performed on two similar GPUs for 20 epochs that lasted 18.6 hours. The two GPUs were needed as the model was complex, and the mini-batch size was 360. Their results are comparable to the state-of-the-art single models like DeepFace [31], DeepID [32] and BaiduFace [33].

The authors in [23] use clustering-based domain adaptation (CDA). They elaborate on how the unsupervised domain adaptation methods for object classification are not applicable to face recognition tasks. The reasons are that a larger discriminating power for the classification of faces is required, and the classes in both domains are non-overlapping. CDA generates pseudo-labels and uses cosine-similarity to form a cluster. They also use deep domain confusion network (DDC) [34] and deep adaptation networks (DAN) [35]. Here MMD estimator is integrated into the CNN error to minimize domain divergence. Thus the end classification is done based on features invariant to domain changes.

They trained the CNN with labeled source data and fine-tuned it with clustered target pseudo-labeled data, which helps determine the target data's discriminative representation. They evaluated their method on GBU [36], IJB-A/B/C [37], [38], [39] and RFW [40] datasets. The architectures that they used were VGGNet [41] and ResNet-34 [42]. Both architectures are trained on CASIA-WebFace, the former tuned using Softmax loss and later with Arcface loss [43]. They preprocessed the images of datasets by resizing, aligning and augmenting them. A Gaussian kernel is used in the MMD.

Their results outperform LRPCA-face [36], Fusion [44], VGG [44], Arcface [43] DDC [34] and DAN [35] for the GBU dataset. They remark that a uniform face-aligned algorithm can achieve good FR performance. Also, incorporating MMD helps in minimizing domain discrepancy. Similarly, better performance is obtained for IJB-A/B/C and RFW datasets. They also showed the visual representations of the learned features using t-distributed stochastic neighbor embedding (t-SNE) [45].

V. Proposed MMCD-VAE Latent Feature Extraction Model

A. Architectures

1) Deep Autoencoder (DAE): In a Deep Autoencoder (DAE) feature selection function is carried out by an encoder. Later a decoder reconstructs the best image corresponding to the selected features. Deep CNN models are very powerful in feature extraction of the images generated from deep CNN AE. These decoders are noise-free and have competent low-dimensional feature space representation. However, only CNN-based generation requires uniform samples from all the categories.

2) Variational Autoencoder (VAE): Variational Autoencoder (VAE), as shown in Fig. 5 is an unsupervised probabilistic deep-neural network model consisting of an encoder-decoder pair. The encoder carries out dimension reduction and domain adaptation by having a progressively lesser number of neuronal units in a feed-forward architecture. The decoder does the reverse and brings back the compressed domain representations to their original shape by gradually increasing the number of neuronal processors. Variational autoencoders are the fabrication of a CNN Autoencoder with regularized training to avoid over-fitting. It results in a latent space favorable for the generative process. VAE is unique in the way

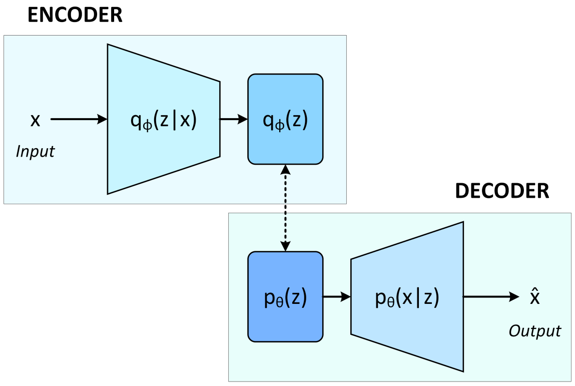

Fig. 5. Mapping Distribution of a Variational Autoencoder (VAE).

it uses the selected features for latent representation shared between encoder-decoder pairs.

Latent representation is nothing but the distribution of collected traits used as the communication protocol between the encoder-decoder pair. In practice, encoding and decoding distributions are parametric models. Joint optimization leading to reliable reconstruction ensures latent features contain the most salient statistical features and capture variations over main features.

The Face Recognition task falls under categorical marginal distribution. Assuming that ϕ and θ are parameter sets for encoder and decoder, optimized for minimum reconstruction loss, then VAE objective function can be written as:

$$\begin{aligned} L_{VAR\_ELBO} = \\ & -\gamma D\left(q_{\varnothing}(z) \| p_\theta(z)\right) \\ & + E(x) E_{q_{\emptyset}(z|x)}\left[\log p_\theta(x \\ & \mid z)\right] \end{aligned} \tag{4}$$

where D is any strict divergence and $\gamma > 0$ is a scaling coefficient, E is the expectation operator, q_ϕ and p_θ are the distribution functions of encoder and decoder, respectively. The selection of divergence can play a crucial role. Traditionally evidence lower bound (ELBO) criterion is used in VAEs. The goal of the encoder is to obtain a simplified approximate distribution q and optimize the variational parameter ϕ such that q_ϕ be as similar as possible to the true distribution of inputs. One of the approaches is to minimize Kullback-Leibler (KL) divergence. It is defined as:

$$\text{KL}\left[q_\phi(w \mid D), p(w \mid D)\right] = \int q_\phi(w \mid D) \frac{q_\phi(w \mid D)}{p(w \mid D)} dw \tag{5}$$

where p(w | D) is the actual distribution of input samples w. Intractability due to the integration term present in equation 5, is resolved by substituting an approximation for p in terms of q_ϕ. This substitution results in the popular Bayes by Backprop [46], a tractable objective function. ELBO suffers from uninformative latent code and variance overestimations in the feature space. Also, ELBO-VAE tends to over-fit data, and as a result of the over-fitting, it learns a $q_\phi(z)$ whose variance tends to infinity.

3) Proposed MMCD-VAE Model for Domain Adaptation: The proposed MMCD-VAE Model for Domain Adaptation is shown in Fig. 6. The encoder generates the same distribution for all possible variations in a sample's inputs, which works for learning good features. Regularization is possible as the input is encoded to a distribution with some variance instead of a point. Regularization aims to have continuity and completeness in the generative process. Distributions are forced to be as close as to a standard normal distribution.

MMD evaluates the distribution as identical if and only if all their first moments are the same. Therefore, MMD divergence is a metric of differential moments of p(z) and q(z) distributions and is accomplished using the kernel embedding trick [47]. MMD prefers to maximize the mutual information between an input x and the latent representation z. Training ELBO on a dataset with complimentary samples will still try to obtain encoder q_ϕ and decoder p_θ as Gaussian distributions with non-zero variance. For ELBO regularization term $\gamma D(q_\phi(z) \mid\mid p_\theta(z))$ is not strong enough as against the loss function term $E(x)E_{q\phi(z|x)}[logp_\theta(x \mid z)]$. Complimentary samples will have class means way apart, and accordingly, MMD optimization will end up by having two modes of q_ϕ, pushed to stay far from each other. This will reduce ambiguity in reconstruction. In practice, this matters for datasets with fewer samples.

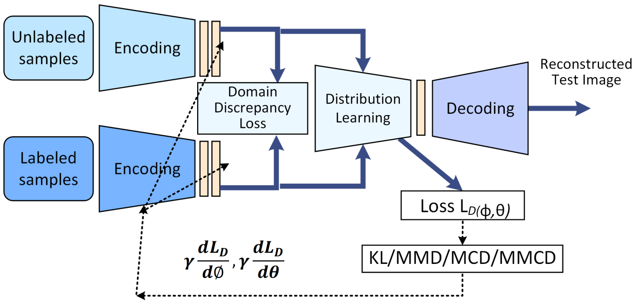

Fig. 6. Proposed MMCD-VAE Model for Domain Adaptation

The Loss function (objective function) indicates the degree to which the test image has been reconstructed and is given by:

$$\begin{aligned} l_i(\phi, \theta) = & -E_{z \sim \theta_\phi(z|x_i)}\left[\log P_\theta\left(x_i|z\right)\right] \\ & + \text{MMCD}\left[Q_\phi\left(z|x_i\right)||P_\theta(z)\right] \end{aligned} \tag{6}$$

Given two distributions p, q in RHKS

$$MMCD[p, q, H] = \left(\|\mu[p] - \mu[q]\|_H^2 + \beta\|C[p] - C[q]\|_H^2\right)^{1/2} \tag{7}$$

where $\mu[p] = E_x[\phi(x)]$ and β is a non-negative parameter. But $C[p] = E\left[w_p w_p^\top\right] - E\left[w_p\right] E\left[w_p\right]^\top$ and
$C[q] = E\left[\omega_q \omega_q^\top\right] - E\left[\omega_q\right] E\left[\omega_q\right]^\top$

Algorithm 1: The Proposed MMCD-VAE Algorithm

```
Data:
Training Dataset $= \{X_i^n\}_{i=1}^N$
Testing Dataset $= \{Y_i^n\}_{i=1}^N$
Encoder Network $= q_\phi$
Decoder Network $= p_\theta$
Batch Size $= B$
Epochs $= S$
Learning Rate $= \alpha$
Result:
Reconstructed Test Image
Initialize parameters of the Encoder and Decoder
for epochs = $i \leftarrow 1$ to $S$ Randomly select batches of
  Input images from the Training dataset do
    for $i \leftarrow 1$ to $N$ $\mu_z(i), \sigma_z(i) = q_\phi(z \mid x_i)$ Draw $L$
      samples from $z \sim N(\mu_z(i), \sigma_z(i))$ do
        for $j \leftarrow 1$ to $L$ $\mu_{\hat{x}}(i), \sigma_{\hat{x}}(i) = p_\theta(x_i \mid z)$ do
        end
    end
    Define Objective function (L) using Log
      likelihood and MMCD distance
    Update
    $\phi = \phi_{old} + \alpha * \nabla_\phi \ _{ADAM}(\frac{\partial L}{\partial \phi})$;
    $\theta = \theta_{old} + \alpha * \nabla_\theta \ _{ADAM}(\frac{\partial L}{\partial \theta})$
end
Return trained encoder $= q_\phi$ and trained decoder $= p_\theta$
```

$$\begin{aligned}\mathrm{MMCD}[p,q] = \Big[&\|E[x]-E[y]\|_2^2 + \\ &\beta\left[\Big\|E\left[xx^\top\right]-E[x]E[x]^\top\right. \\ &\left.-\left(E\left[yy^\top\right]-E[y]E[y]^\top\right)\Big\|\right]^2\Big]^{1/2}\end{aligned} \tag{8}$$

where $x \sim p, y \sim q$. Given limited X and Y sampled from p and q respectively there is $MMCD[p,q] \ : \ \left(\|\mu_p - u_q\|_2^2 + \beta\left\|\Sigma_p - \Sigma_q\right\|_F^2\right)^{1/2}$ where $\mu_p = \frac{1}{n}X$ is the mean vector and $\Sigma_p = \frac{1}{n}X\mu_n X^\top$ is the covariance matrix of X.

Substituting Equation (8) in (6) we get:

$$\begin{aligned}l_i(\phi,\theta) = -E_{z\sim\theta_\phi(z|x_i)}\left[\log P_\theta\left(x_i|z\right)\right] + \Big[&\|E[x]-E[y]\|_2^2 + \\ &\beta\left[\Big\|E\left[xx^\top\right]-E[x]E[x]^\top\right. \\ &\left.-\left(E\left[yy^\top\right]-E[y]E[y]^\top\right)\Big\|\right]^2\Big]^{1/2}\end{aligned} \tag{9}$$

The authors in [24] have experimented with different kernel and non-kernel based cases. The kernels used were linear, polynomial, Gaussian, and Exponential. When a linear kernel is adopted, MMD, MCD, and MMCD measure the difference between the mean and covariance of the distributions, respectively.

B. Datasets

1) Bollywood Celebrities Dataset: The Bollywood Celebrities dataset [48] contains the localized face of 100 Bollywood Celebrities. A class has 80 to 150 samples of size 64 × 64 pixels. These are in wild conditions with different orientations, illuminations, age transitions. The sample images are shown in Fig. 7. Experimentation is carried out on 64 × 64 size RGB images.

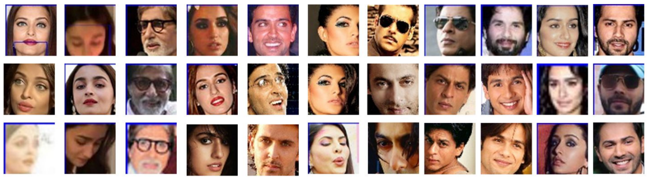

Fig. 7. Sample Images from Bollywood Celebrities Dataset.

2) TinyFace Dataset: The TinyFace dataset contains 5,139 labeled facial identities given by 169,403 native Low Resolution face images (average 20 × 16 pixels) designed for the 1:N recognition test. The sample images are shown in Fig. 8. These are from public web data across a large spectrum and unconstrained environment.

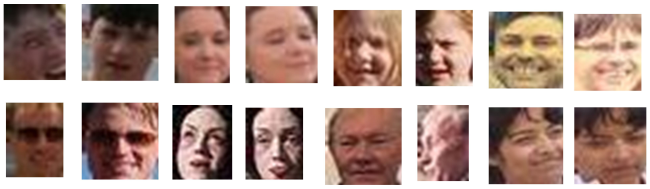

Fig. 8. Sample Images from TinyFace Dataset.

C. Objective Image Quality Comparison Metrics

The quality of the images needs to be evaluated using either a subjective or objective method. The former is based on human judgment, and the latter is by explicit numerical statistical parameters.

1) SSIM: Traditionally the most popular metric for image quality assessment was Peak Signal to Noise Ratio (PSNR). A standard metric is Structural Similarity Index (SSIM) which measures the similarity between two images. It was developed by Wang [49], and looked at structural information changes in the images. SSIM considers three factors, loss of correlation, luminance distortion, and contrast distortion [50]. For the SSIM index, a value of 0 means no correlation between images, and 1 means the two images are the same.

2) PieAPP: PieAPP [51] is a perceptual image-error metric that robustly predicts visual differences like humans. It uses pairwise preference as a robust way to create large Image quality assessment (IQA) datasets and uses a new pairwise-learning framework to train an error-estimation function. A reference image and a distorted image are given as input resulting in a PieAPP value as an output. Lower the value of the PieAPP error metric better the image perceptual quality.

3) SIFT Features: Face recognition is challenging compared to many other object recognition tasks as face features in the two domains are often non-overlapping. Global alignment of the source and target samples is not feasible for unconstrained face images. The goal of the proposed unsupervised domain adaptation model is to discover novel domain-invariant representations using scale-invariant features transform (SIFT) [52], as a parametric evaluation entity for the domain adaptation. Some authors [53], [54] have worked using SIFT for face recognition but have not used VAE. The challenge is to maximize scale-invariant features and thus get the corresponding match.

Many domain adaptation algorithms match the distribution without understanding the goodness in preserving key spatial features. This work analyses domain adaptation by optimizing encoder and decoder parameters. We use training samples and utilize unlabeled testing samples.

VI. Results and Discussion

A. Experimental Setup

The domain adaptation experiments were conducted on NVIDIA GeForce RTX 2070 SUPER GPU. The PC configuration consists of a Multi-core (8 total) and Hyper-threaded (16 total) 3.80 gigahertz Intel Core i7-10700K. The memory is 32 GB, and the SSD hard drive has a 1TB capacity. The software used was Python version 3.8.5 (64-bit), libraries NumPy and Matplotlib, TensorFlow, and Keras.

The Bollywood Celebrities dataset was used for training. As the images for this dataset are 64 × 64, the target images were resized to 64 × 64. 300 epochs were used to train the model.

B. Experimental Results and Discussion

1) Training and Testing on Bollywood Celebrities Dataset: In [24], the authors used MMCD and compared the classification performance using two benchmark datasets PIE and Office-Caltech. Their performance was better than nearest neighbor, principal component analysis, correlation alignment transfer component analysis, geodesic flow kernel, and joint domain adaptation. We have combined MMCD with VAE and the training and testing details are mentioned below.

The MMCD-VAE model was first trained with the Bollywood Celebrities dataset for 300 epochs and then tested on different images from that dataset. The generated images for KL-VAE and MMCD-VAE models with the Training and Testing on Bollywood Celebrities dataset are shown in Fig. 9. SSIM and PieAPP error metric comparison is shown in Table I.

MMCD-VAE performs better than KL-VAE. MMCD-VAE shows an average of 20 % improvement in SSIM and a remarkable improvement in perceptual quality of the image, as seen from the PieAPP error metric, over the conventional KL-VAE model.

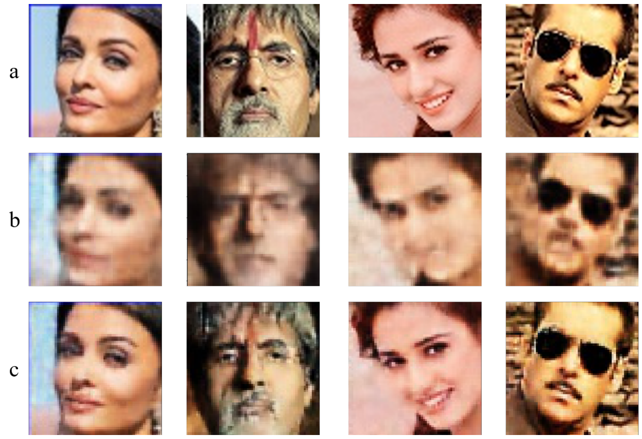

Fig. 9. Results for Bollywood Celebrities Dataset Images (a) Original (b) KL-VAE Generated Image (c) MMCD-VAE Generated Image.

TABLE I. SSIM and PieApp Comparison for KL-VAE and MMCD-VAE with Training and Testing on Bollywood Celebrities Dataset

Face Images	SSIM		PieAPP error metric	
(Bollywood	Original	Original	Original	Original
Dataset)	vs	vs	vs	vs
	KL-VAE	MMCD-VAE	KL-VAE	MMCD-VAE
Actor 1	0.719202	0.915040	3.537072	0.377244
Actor 2	0.646112	0.859116	2.928503	0.365942
Actor 3	0.567586	0.811546	3.581390	0.940700
Actor 4	0.571617	0.790679	3.197515	1.132178
Actor 5	0.660647	0.883705	4.058572	1.056081
Actor 6	0.575935	0.808702	3.890267	1.467767
Actor 7	0.596703	0.881254	2.642451	0.065619
Actor 8	0.677114	0.916528	2.971156	0.904211
Average	0.626865	0.858321	3.350866	0.788718

Fig. 10 demonstrates the SIFT features for the Bollywood Celebrities generated images. The proposed MMCD-VAE method is also applied to face images of the same class, but varying domains and generated face images are tested for inter-class similarity, as shown in Fig. 11. It can be seen that MMCD-VAE generated images have comparatively more SIFT key points than conventional KL-VAE generated images. More scale-invariant features assure that the proposed MMCD-VAE can capture more information.

The reconstruction loss gives the measure of how well the test image has been reconstructed and is shown in Fig. 12. We

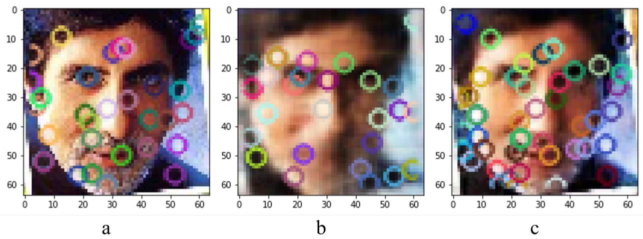

Fig. 10. SIFT Features for Bollywood Celebrities Dataset Images for Same Class (a) Original Image 36 SIFT Features (b)KL-VAE Image 27 SIFT Features (c) MMCD-VAE 49 SIFT Feature.

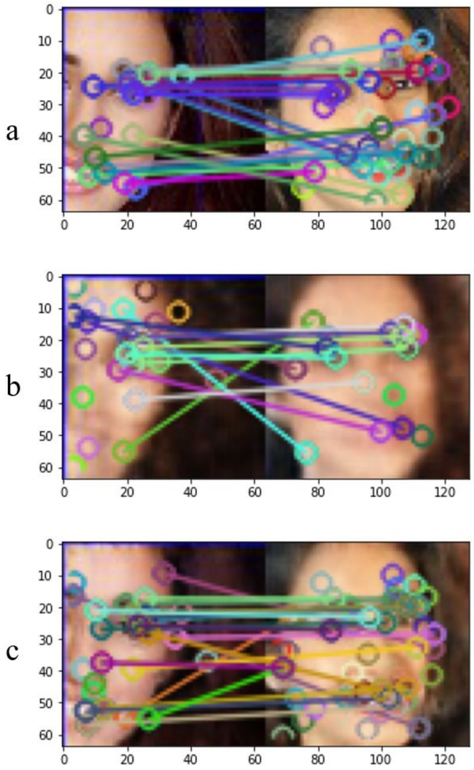

Fig. 11. Results for Bollywood Celebrities Dataset Images for Same Class Different Domain(a) Original Image 18 SIFT Matching Features (b) KL-VAE Image 12 SIFT Matching Features (c) MMCD-VAE 25 SIFT Matching Features.

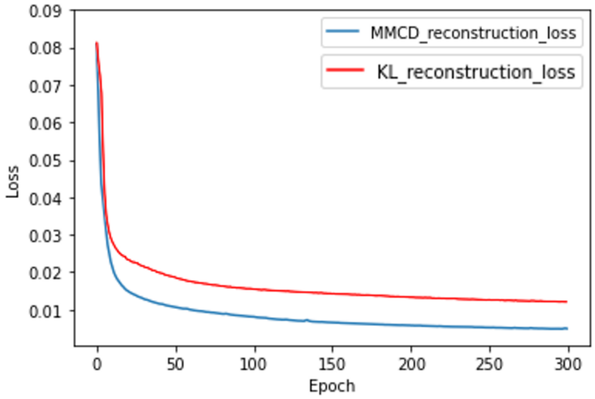

Fig. 12. Reconstruction Loss for the Conventional KL-VAE v/s Proposed MMCD-VAE.

TABLE II. SSIM AND PIEAPP COMPARISON FOR KL-VAE AND MMCD-VAE WITH TRAINING ON BOLLYWOOD CELEBRITIES AND TESTING ON TINYFACE DATASET

Face	SSIM		PieAPP error metric	
Images	Original	Original	Original	Original
(TinyFace	vs	vs	vs	vs
Dataset)	KL-VAE	MMCD-VAE	KL-VAE	MMCD-VAE
Face 1	0.472164	0.671485	1.819303	1.429740
Face 2	0.527809	0.668509	3.577199	2.522136
Face 3	0.559499	0.735055	1.111426	1.166033
Face 4	0.477758	0.650393	2.297195	1.644578
Face 5	0.529925	0.754939	1.154995	0.967473
Face 6	0.377013	0.618141	4.915103	1.919206
Face 7	0.508398	0.664005	1.727580	1.665994
Face 8	0.530098	0.667237	3.178573	1.667042
Average	0.497833	0.678721	2.472672	1.622775

observe that the MMCD-VAE model training is stable like the conventional KL-VAE, and demonstrates that the MMCD-VAE reconstruction loss is a meaningful metric of progress.

2) Training on Bollywood Celebrities Dataset and Testing on TinyFace Dataset: In VAE networks, the latent representations correspond to different levels of abstraction mapped to multifarious face attributes. Better the hidden representations, the greater is the adaptation quality. The MMCD-VAE model trained with Bollywood Celebrities dataset for 300 epochs was tested on TinyFace data. The total dataset was not tested but only a sample was used to check the results. The MMCD-VAE model performs better than the KL-VAE model, as seen from the subjective quality of the generated face images given in Fig. 13. The TinyFace dataset images are low resolution images. Even in the case of an original blurry image, the generated image has clearer features of eyes, nose, and mouth. As seen in Fig. 14, there are more SIFT key points in MMCD-VAE than KL-VAE generated images.

SSIM and PieAPP error metric comparison is shown in Table II. MMCD-VAE performs better than KL-VAE. MMCD-VAE shows an average of 18 % improvement in SSIM and an improvement in perceptual quality of the image over the conventional KL-VAE model. In this case, the PieAPP error metric difference between KL-VAE and MMCD-VAE is smaller than the one observed with the Bollywood Celebrities dataset images as the TinyFace are low-resolution images.

VII. CONCLUSION AND FUTURE WORK

This study reviewed the literature on domain adaptation, especially in Face Recognition. It began by looking into the challenging problem of how models trained on benchmark datasets, at times, fail in real-world scenarios. One example is test images collected from the online web. The benchmark dataset on which a model is trained is often high resolution and performs poorly for low-resolution target images. This happens because the source and target domain experience shifts due to changing conditions. Hence the need for domain adaptation and the various metrics for determining the distribution discrepancy.

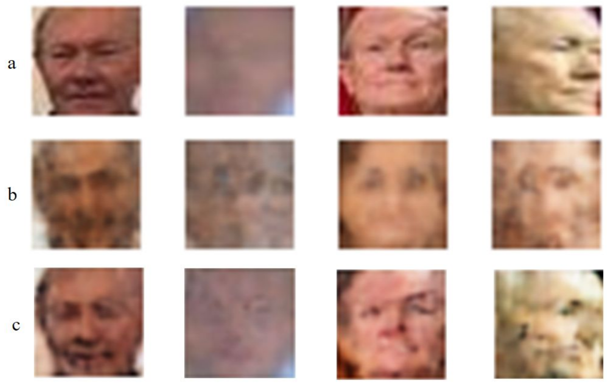

Fig. 13. Results for TinyFace Dataset (a) Original Image (b) KL-VAE Generated Image (c) MMCD-VAE Generated Image.

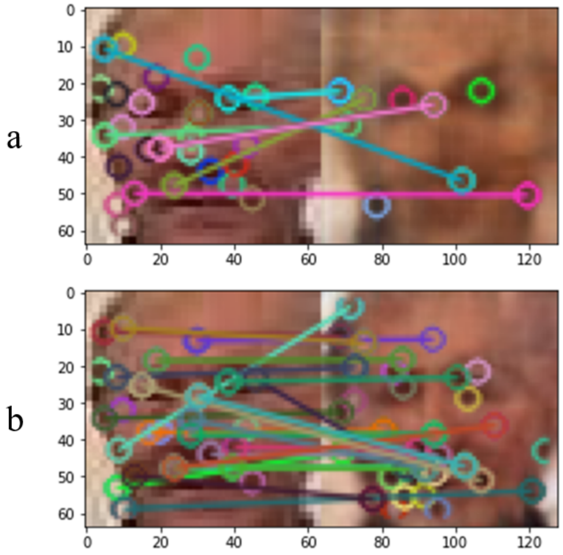

Fig. 14. SIFT Matching Features (a)Original and KL-VAE (b)Original and MMCD-VAE.

In the experimental part, we compared the performance of the proposed MMCD-VAE model. Results are compared for sample images taken from the Bollywood Celebrities dataset and TinyFace dataset. TinyFace is a challenging dataset, because it is low-resolution and recognition performance drops with the decrease in resolution. Quantitative comparisons are shown for matching SIFT key points and SSIM. The MMCD-VAE domain adaptation method rendered images with better Objective Image Quality, as seen in the SSIM, pieApp, and SIFT key-points metrics.

The future scope is to look at detailed testing of RFW datasets to better understand how to improve face recognition across diverse races. The low-resolution surveillance face images of the QMUL-SurvFace dataset is another area to pursue further research. An emerging area of research is adversarial discriminative domain adaptation, which reduces the difference between the source and target domain distributions using adversarial learning methods.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira *et al.*, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, p. 137, 2007.

[2] J. Quiñonero-Candela, M. Sugiyama, N. D. Lawrence, and A. Schwaighofer, *Dataset shift in machine learning*. Mit Press, 2009.

[3] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern recognition*, vol. 45, no. 1, pp. 521–530, 2012.

[4] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of statistical planning and inference*, vol. 90, no. 2, pp. 227–244, 2000.

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*. PMLR, 2014, pp. 647–655.

[6] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[7] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[8] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[9] J. Zhang, W. Li, and P. Ogunbona, "Transfer learning for cross-dataset recognition: a survey," *arXiv preprint arXiv:1705.04396*, vol. 5, 2017.

[10] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks*. Springer, 2018, pp. 270–279.

[11] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE signal processing magazine*, vol. 32, no. 3, pp. 53–69, 2015.

[12] S. Sun, H. Shi, and Y. Wu, "A survey of multi-source domain adaptation," *Information Fusion*, vol. 24, pp. 84–92, 2015.

[13] G. Csurka, "A comprehensive survey on domain adaptation for visual applications," *Domain adaptation in computer vision applications*, pp. 1–35, 2017.

[14] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[15] G. Wilson and D. J. Cook, "A survey of unsupervised deep domain adaptation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 5, pp. 1–46, 2020.

[16] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, "A practical transfer learning algorithm for face verification," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3208–3215.

[17] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel method for the two-sample-problem," *Advances in neural information processing systems*, vol. 19, pp. 513–520, 2006.

[18] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[19] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 929–942, 2009.

[20] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[21] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1718–1727.

[22] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, "Training generative neural networks via maximum mean discrepancy optimization," *arXiv preprint arXiv:1505.03906*, 2015.

[23] M. Wang and W. Deng, "Deep face recognition with clustering based domain adaptation," *Neurocomputing*, vol. 393, pp. 1–14, 2020.

[24] W. Zhang, X. Zhang, L. Lan, and Z. Luo, "Maximum mean and covariance discrepancy for unsupervised domain adaptation," *Neural Processing Letters*, vol. 51, no. 1, pp. 347–366, 2020.

[25] G. Wen, H. Chen, D. Cai, and X. He, "Improving face recognition with domain adaptation," *Neurocomputing*, vol. 287, pp. 45–51, 2018.

[26] Z. Luo, J. Hu, W. Deng, and H. Shen, "Deep unsupervised domain adaptation for face recognition," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 453–457.

[27] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[28] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*, 2015.

[29] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[31] Y. Taigman, M. Yang, M. Ranzato, and 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*, 2014, pp. 1701–1708.

[32] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1891–1898.

[33] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, "Targeting ultimate accuracy: Face recognition via deep embedding," *arXiv preprint arXiv:1506.07310*, 2015.

[34] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *arXiv preprint arXiv:1412.3474*, 2014.

[35] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*. PMLR, 2015, pp. 97–105.

[36] P. J. Phillips, J. R. Beveridge, B. A. Draper, G. Givens, A. J. O'Toole, D. Bolme, J. Dunlop, Y. M. Lui, H. Sahibzada, and S. Weimer, "The good, the bad, and the ugly face challenge problem," *Image and Vision Computing*, vol. 30, no. 3, pp. 177–185, 2012.

[37] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1931–1939.

[38] C. Whitelam, E. Taborsky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen *et al.*, "Iarpa janus benchmark-b face dataset," in *proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 90–98.

[39] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney *et al.*, "Iarpa janus benchmark-c: Face dataset and protocol," in *2018 International Conference on Biometrics (ICB)*. IEEE, 2018, pp. 158–165.

[40] M. Wang, W. Deng, J. Hu, X. Tao, and Y. Huang, "Racial faces in the wild: Reducing racial bias by information maximization adaptation network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 692–702.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

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

[43] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[44] P. J. Phillips, "A cross benchmark assessment of a deep convolutional neural network for face recognition," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 705–710.

[45] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[46] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*. PMLR, 2015, pp. 1613–1622.

[47] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 1999.

[48] Y. Sushilkumar, "Bollywood celebrities localized face dataset," https://www.kaggle.com/sushilyadav1998/bollywood-celeb-localized-face-dataset, Nov. 2017, [Online; accessed 1-July-2021].

[49] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[50] A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 2366–2369.

[51] E. Prashnani, H. Cai, Y. Mostofi, and P. Sen, "Pieapp: Perceptual image-error assessment through pairwise preference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1808–1817.

[52] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[53] C. Geng and X. Jiang, "Face recognition using sift features," in *2009 16th IEEE international conference on image processing (ICIP)*. IEEE, 2009, pp. 3313–3316.

[54] L. Lenc and P. Král, "Automatic face recognition system based on the sift features," *Computers & Electrical Engineering*, vol. 46, pp. 256–272, 2015.