sensors

MDPI

Article

A Bayesian Scene-Prior-Based Deep Network Model for Face Verification

Huafeng Wang [1,2,*,†], Wenfeng Song [2,†], Wanquan Liu [3,*,†], Ning Song [2,*,†], Yuehai Wang [1,†] and Haixia Pan [2,*]

1 Department of Electronics and Information Engineering, North China University of Technology, Beijing 100144, China; wangyuehai@ncut.edu.cn
2 Department of Software, Beihang University, Beijing 100191, China; swfbuaa@163.com
3 Department of Computing, Curtin University, Perth, WA 6102, Australia
* Correspondence: wanghuafeng@ncut.edu.cn (H.W.); W.Liu@curtin.edu.au (W.L.); zy1621125@buaa.edu.cn (N.S.); haixiapan@buaa.edu.cn (H.P.); Tel.: +86-189-1192-4121 (H.W.)
† These authors contributed equally to this work.

Received: 12 May 2018; Accepted: 8 June 2018 ; Published: 11 June 2018

Abstract: Face recognition/verification has received great attention in both theory and application for the past two decades. Deep learning has been considered as a very powerful tool for improving the performance of face recognition/verification recently. With large labeled training datasets, the features obtained from deep learning networks can achieve higher accuracy in comparison with shallow networks. However, many reported face recognition/verification approaches rely heavily on the large size and complete representative of the training set, and most of them tend to suffer serious performance drop or even fail to work if fewer training samples per person are available. Hence, the small number of training samples may cause the deep features to vary greatly. We aim to solve this critical problem in this paper. Inspired by recent research in scene domain transfer, for a given face image, a new series of possible scenarios about this face can be deduced from the scene semantics extracted from other face individuals in a face dataset. We believe that the "scene" or background in an image, that is, samples with more different scenes for a given person, may determine the intrinsic features among the faces of the same individual. In order to validate this belief, we propose a Bayesian scene-prior-based deep learning model in this paper with the aim to extract important features from background scenes. By learning a scene model on the basis of a labeled face dataset via the Bayesian idea, the proposed method transforms a face image into new face images by referring to the given face with the learnt scene dictionary. Because the new derived faces may have similar scenes to the input face, the face-verification performance can be improved without having background variance, while the number of training samples is significantly reduced. Experiments conducted on the Labeled Faces in the Wild (LFW) dataset view #2 subset illustrated that this model can increase the verification accuracy to 99.2% by means of scenes' transfer learning (99.12% in literature with an unsupervised protocol). Meanwhile, our model can achieve 94.3% accuracy for the YouTube Faces database (DB) (93.2% in literature with an unsupervised protocol).

Keywords: Bayesian network; deep learning network; scene transfer; deep features; face verification

1. Introduction

Face verification or recognition has attracted much attention with rapid progress in the past two decades, particularly equipped with recent deep learning techniques for significant performance improvement. However, its high performance usually depends on features extracted from a large number of labeled training samples as requested by the developed deep learning techniques. In these

large samples, one main challenge is to recognize a number of faces captured in various scenes that do not appear in such training scenes. Several previous works [1–4] have already suggested that an acquired face should be regarded as a mixture of two components, one being information in the the face region, such as pose [5], expression [6], and age [7], and the other being the backgrounds associated with the face region, illumination [8], and so on, with an example shown in Figure 1. We note that we do not deal with pose or expression separately as other researchers have done before; we integrate them into background scenes and tackle them in one framework.

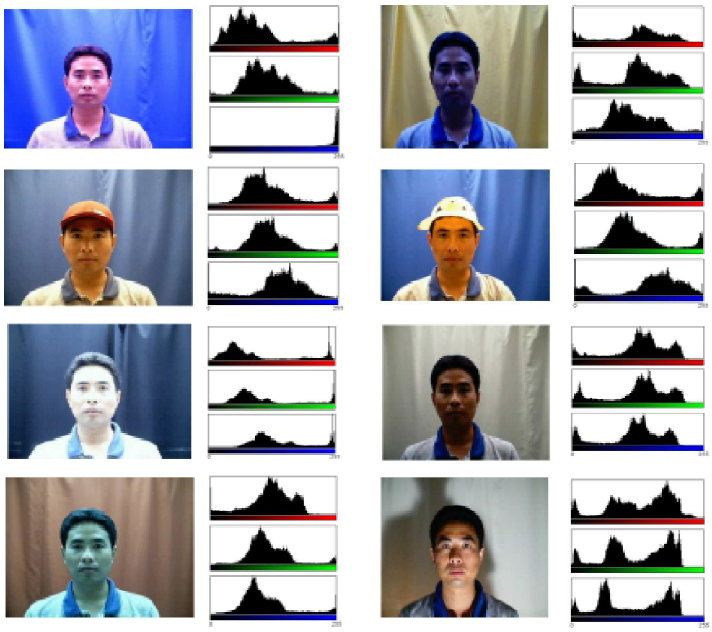

Figure 1. Faces in various illuminated scenes: the histograms indicate the large varieties for the same individual in different scenes.

Although most of the previous works in literature perform very well for popular face databases such as the LFW and YTF datasets [9,10], they still to some extent rely on the background variance of the dataset, which is referred to as the "scene" in this paper. That is to say, they require the training dataset to be large enough, that is, including many samples, to represent sufficient scenes as requested. Currently, many researchers focus on proposing approaches for face recognition/verification on the basis of idealistic scenes and validate their methods on specific datasets while ignoring the fact that the face samples in training and testing sets are frequently present in different scenarios. Therefore, many previous approaches are limited or fail for some applications if the size of the training samples is quite limited in terms of scenes. In order to build a robust face recognition or verification system, effective feature extraction as well as semantic scenes extracted from training samples are highly recommended [11]. The extraction of scene semantics in natural scenes has been well studied in [12] and [13]; we borrow the scene concept for face semantic segmentation tasks. In brief, the motivation of this paper is attributed to the fact that the same person with various backgrounds can improve the face-verification performance. This rationality can be justified by the following two aspects: on one hand, the dataset is effectively augmented via domain transfer learning; on the other hand, the final features learned from deep neural networks (NNs) are facilitated because of the enrichment of individual face scenes used for training. In summary, the main contributions of this paper can be summarized as follows:

- We propose a scene model based on the Bayesian deep network technique, which can infer several complicated scenes for the face-verification task.

- A new unsupervised face-verification model is developed on the basis of the scene transfer learning technique.
- Experiments on two challenging datasets validated the proposed model in the case of a lack of sufficient training samples.

The organization of the rest of the paper is as follows: In Section 2, we review the literature in this area; Section 3.1 develops the Bayesian prior scene model, and Section 3.2 focuses on the scene inference. Finally, in Section 4, we propose the deep learning model and present some quantitative results of our Bayesian scene based network model for face verification. The conclusions are given in Section 5.

2. Previous Work

As deep learning NNs are our main concern in this paper, we only review some results related to NNs. In literature, there are three main categories of networks related to the proposed network model: highly deep network based (HDNB), large-dataset-based (LDB), and multimodal-based (MMB) networks. As for the HDNB network, this category needs to use labeled data with a very deep network for training in order to achieve higher accuracy. In practice, the HDNB approaches rely on a deep structure, and they comprise a long sequence of convolutional layers. For example, the deep residual net [14] has more than 1000 layers and achieved 5.71% top-1 error on ImageNet validation. In 2014, Christian Szegedy [15] proposed a 22 layer net, the "Google Net", which is a network with a carefully crafted design that allows for increasing its depth and width while keeping the computational budget constant. However, this type of network model cannot be too deep because of vanishing gradients. Fortunately, Ropes Kumar Srivastava et al. [16] developed an approach to enable the depth of the net to increase without the vanishing-gradients constraint. They use the "transform gate" to transmit the information derived from the input data to keep the gradient descent from vanishing. Even with hundreds of layers, such networks still can be trained directly through simple gradient descent. Their research enlightened the study on extremely deep architectures with efficiency. LDB methods aim to train a classifier with a large dataset. Yana Taiga et al. [14] used 4 million data samples with a bootstrapping process and improved the transferring capability of the network with the aim of discovering the connection between the representation and the capability of discrimination. In 2015, Google [17] proposed a convolutional neural network (CNN) model (FaceNet) that could directly learn a compact Euclidean mapping from face images. As reported, it could achieve a highaccuracy of 99.6% on the LFW benchmark [10]. However, this approach needs to be trained on a large dataset with a data size of about 200 million. Yana Taigman [18] (Deepface) employed a 3D face model to align using locally connected layers without weight sharing. Later they declared [3] that the CNN method has a bottleneck with increasing data. They proposed a solution for alleviating this by replacing the naive random subsampling in the training set with a bootstrapping process. Moreover, a link between the representation norm and the capability of discrimination in a target domain was discovered, and this research sheds lights on how such networks can represent faces. Although it is suggested that the larger the data size is, the higher accuracy one can achieve, it is also clear that the last 0.4% point is hardly to be achieved by only increasing the size of the training dataset. Particularly, 0.1% improvement needs an increase in the data size of 199.99 million; the tendency is shown in Figure 2. MMB approaches are in essence associated with ensemble methods, which involve multi-models to work together. Jintao Liu et al. [16] (the Baidu Face) exploited a two-stage method based on multi-patch features and metric learning with triplet loss. Their method achieved 99.77% for pairwise verification. However, it tends to deplete too many computing sources because it needs too many overlapped computing jobs. Sun Jain et al. [19] tried different structures and different patches to extract the aligned face features. After that, they proposed a classifier to differentiate between different faces (verification or identification) and achieved high accuracy on both the LFW benchmark and Casia [2] dataset.

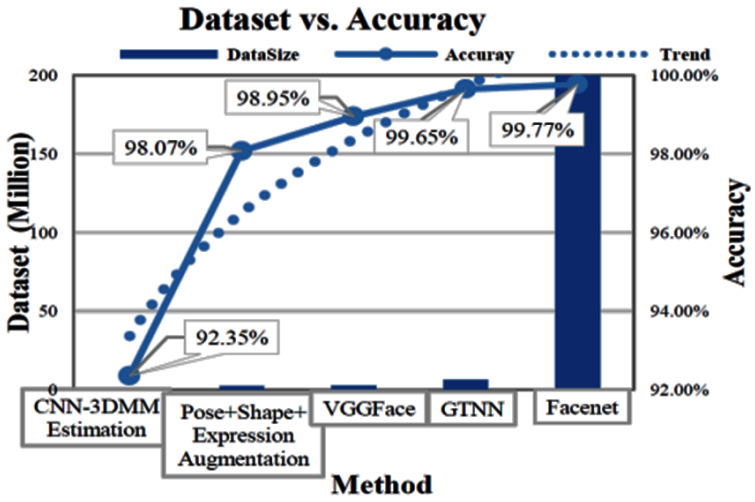

Figure 2. An illustration of accuracy with increasing dataset size.

As far as above three categories to be concerned, there is a consensus that the key issues to improve the face recognition/verification performance are to design a reasonable deep network with appropriate data size and then to work with a hybrid process model. However, the reality is that we can often have different methods while lacking an appropriate size of labeled data at hand. This is in fact the well-known problem of a small number of training samples, and it has been investigated extensively by the computer vision community [20]. This problem has not been sufficiently tackled for deep learning NNs, and currently researchers are trying hard to develop new network structures or promote broad applications for the currently existing network structures. This is mainly due to the fact that deep learning NNs are very complicated, as they aim to mimic the human brain functions and are still in a developing stage. To our surprise, the networks for the small data size problem do not perform as well as human beings. That is to say, human beings can distinguish between a large number of faces by using very few datasets in learning. For example, the work by Salakhutdinov, Ruslan et al. [21] could extract new features from very few training examples by learning both low- and high-level generic features, and these features could fully represent correlations between low- and high-level features. Motivated by their approach, we define the high-level features as a scene in this paper and propagate these scene factors to a deep learning network that is exploited to learn a practical model from input face images. Details are presented in following sections. For convenience, the related symbols and concepts are first listed in Table 1.

Table 1. Symbols and notation.

Symbol	Notation
s = (s_i, s_j)	Scene; number is unknown in advance
ϕ_{mix}, ϕ_{pure}	Two feature spaces, the pure and the mixed space
C	Category
p(,)	PDF (probability density function)
I_{mix}	The overlapping image space
θ	The statistics (μ; Σ)
$I(u, v)$	The pixel value at position u, v
π_k	The PDF of the kth scene
i,j,k	The image, category, and scene orders
ϕ_x	Features extracted from the image

3. The Proposed Methodology

To better understand the method presented in this paper, we need to clarify the concepts of the high-level features or scene first. Previously, S. Zheng et al. introduced a new form of CNN that combines the strengths of CNNs and conditional random field (CRF)-based probabilistic graphical modeling for segmentation [13,22,23]. Motivated by the proposed methods in [13,24], we embed the scene concept into the above semantic segmentation task. By "plugging" CRFs into the CNN, we can obtain a new deep network that has desirable properties derived from CNNs and CRFs. Although the network was originally designed for semantic segmentation, it provides great benefits when we integrate this idea into our approach for scene extraction and scene backwards propagation. In Figure 3, the first column is the original scene, and after the semantic segmentation shown in the second column, we can deduce more new scenes from columns 3 to 6, as we explain in the following sections.

Figure 3. The scene illustration by process of conditional random field (CRF) transferred to our proposed method.

3.1. The Bayesian Scene-Prior-Based Deep Network Model

First, the proposed pipeline for face verification in this paper is outlined in Figure 4. As shown in Figure 4, the whole process consists of two steps: the training and the verification. Next, we explain each part in detail. We first propose a combination of a Bayesian network and a CNN, which concerns both the global and the local feature distributions. Similarly to a human being's cognitive process, it is believed that a good object classifier should first have a good capability to understand the scenes, and then its capacity for knowing about objects' existence in these scenes can be much improved. In view of this, the proposed method consists of the following steps: (1) to learn the scenes; (2) to express the scenes; (3) to feed scene factors into the CNN training process and optimize the parameters for face verification; and (4) to finally feedback the learnt knowledge into a new learning iteration step if a new scene is given. Inspired by the contributions of a previous study that utilized the latent Dirichlet allocation (LDA) model to learn the natural scene categories [12], we propose a similar new method to learn the scene distributions between the face pairs' overlapping spaces. In this context, we consider the scene variance as continuous and use the mixture Gaussian model instead of the multi-nomial model to describe such distributions. The main idea is to transform a face pair into a very close scene in order to lessen the possible scene variation effect in face verification. As shown in Figure 5, after roughly detecting and aligning the face images from a dataset, the detected faces are segmented by the CRF to build a series of scene candidates. These preliminary candidates are then dealt with by the succeeding CNN feature extraction in order to learn a scene expression. Ultimately, a scene dictionary is output according to the distance measurement among any given face pair for the same person.

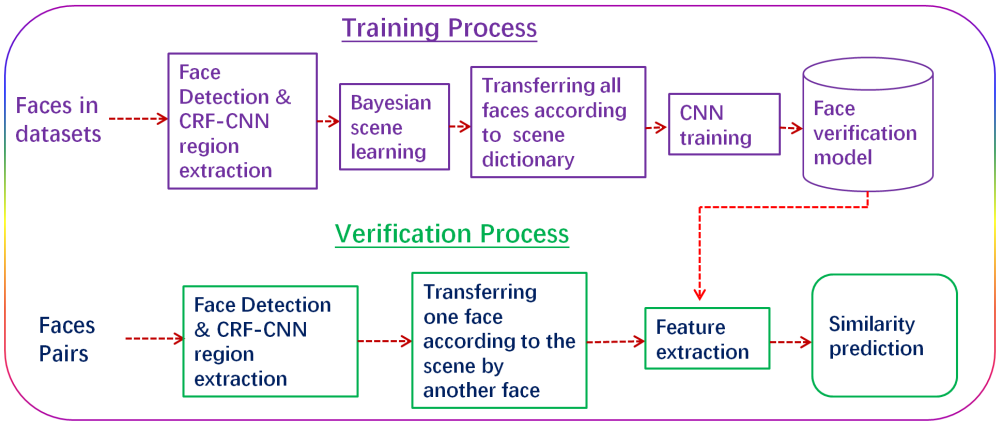

Figure 4. A pipeline for the proposed method.

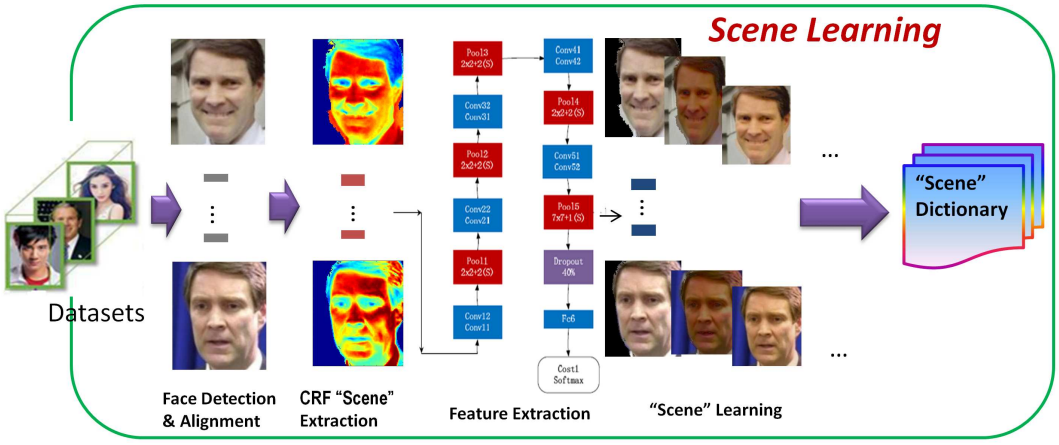

Figure 5. A brief view of the scene learning model.

For clarity, we further define the scene learning by a rigid mathematical description:

$$s = \{s_1, s_2, \ldots, s_M\}, \tag{1}$$

$$d(face_i, face_j) = \sqrt{(face_i - face_j)^T \Sigma^{-1} (face_i - face_j)}, \tag{2}$$

where Σ is the covariance matrix, and M is the number of scenes for a given person. The initial dictionary entry is determined on the basis of the distance d between the given face pairs.As noted, the covariance matrix is estimated by the face feature vector for the same person.

For the purpose of generating an augmented new dataset according to the learnt scenes, we need to know the relationship between the scenes and features first. Once a scene entry in the dictionary is given, the feature distribution for the same person can be expressed as a mixture Gaussian distribution, as below:

$$p(x) = N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{N/2}} |\Sigma| exp\{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\}. \tag{3}$$

Given the parameter θ, and where μ and Σ are the distribution parameters' mean and the covariance to be learnt in the training process, then given the scene S, the feature f, and the scene S, the model described in Equation (4) is followed:

$$p(f, s) = p(f|s)p(s) \rightarrow N(\theta|s). \tag{4}$$

When any two faces of a pair have a different θ value, the distance between them will be relatively large; conversely, the distance tends to be small. At the beginning of the procedure for our model, the face categories or scene entries C are randomly initialized. We suppose that there is a dataset containing M images denoted by $X = (x_1, x_2, \ldots, x_m)$ for a given person i. For convenience, we use the symbol x instead. One image may have K scenes with all the probabilities correlative to the change in the faces. In order to decide which scene a given image belongs to, the nearest-neighbor rule is applied to the set of the probabilities calculated from Equation (4) above. The scenes are not in discrete space, but they are continuous in $s \in [0, 1]$. The closer s is to 1, the more likely the image belongs to the given scene. Therefore,

$$p(x) = \sum_{s} p(s)p(f\,|s) = \sum_{k=1} \pi_k N(x_k|\mu_k, \Sigma_k), \tag{5}$$

where π_k is the distribution of scene s_k. Eventually, more faces can be generated for the imbalance training dataset according to Equation (5) on the basis of the learnt scene dictionary, as shown in Figure 6.

Figure 6. Generated faces according to the scenes.

As for the face pairs shown in Figure 7, Equation (4) can be expressed in another way, as follows:

$$p(x, y) = p(x|s)p(s|y)p(y). \tag{6}$$

Figure 7. The face pair with different scenes (each image has a different scene).

Furthermore, on the basis of the conditional distribution in the graph model of $p(x|s)$, we denote this latent variable by $\gamma(s_k)$; thus,

$$\begin{aligned} \gamma(s_k) = p(s_k|x) &= \frac{p(s_k)p(x|s_k{=}\gamma_k)}{\sum_{l=1}^{k}(p(s_k{=}\gamma_k)p(x|s_k{=}\gamma_k)} \\ &= \frac{\pi_k N(x|s_k, \Sigma_k)}{\sum_{l=1}^{K}(\pi_l N(x|\mu_l, \Sigma_l)}, \end{aligned} \tag{7}$$

where π_k is the prior probability of a scene. The above Bayesian model describes the prior relationships among the images, features, and image categories. An image needs a scene to describe the condition that a face is shown in it; therefore, one person's face may have several scenes to be assigned to. In this context, the scene model is used as a prior to the image, and then we exploit the semantic pixels to determine the face region. This priori information is backward-propagated to the next iteration for generating more reasonable faces. We usually refer to the scenes as the learnt higher-level features, each of which will have a dramatic effect on the distribution of the face images. During this iteration, the face in different scenes is determined semantically [25] by calculating the low-level (pixel-level) distribution, which can be expressed as a Dirichlet distribution. The scene of a given face is one of K random variables subject to

$$0 \leq \gamma(s_k) \leq 1, \sum_{k=1} \gamma(s_k) = 1. \tag{8}$$

Up to here, we have addressed how to create K scenes from training samples; next we explain how to infer a scene for a given face.

3.2. Scene Inference

Each image is a mixture of scenes, where the corresponding latent variables $\gamma(s_k)$ are denoted by an $N \times K$ matrix x with s_n^k rows (for a given person with n face images and K scenes). We let s be the kth scene, such that

$$p(s|x, \mu, \sum) \propto p(x|s, \mu)p(s|\sum) \propto p(x|s, \mu, \Sigma), \tag{9}$$

where μ and Σ are parameters learnt from the ground truth. The PDF of a scene is given by Equation (9). The scene can be considered as a main background factor for the identity of a given object, which is learnt from mixed image spaces. When a new scene is to be learnt, the original imbalanced data will even be augmented by generating new scene images. Eventually, this enlarges the original image space in the direction of the missing data characteristics by using iterative backward propagations. The number of scenes is a latent variable, which is a constant and is determined by the images from the training set. Then, the scene s is subject to

$$\mathrm{s} = \underset{s}{argmax}(p(s|\theta)). \tag{10}$$

Now the problem is how to propagate the scene information to the data features and make the data richer with the new scene feature. The aim is to obtain $p(image|s_{mix})$, where s_{mix} is the hybrid face scene in a verification task. Up to here, we have already been able to calculate $p(x)$, $p(c)$, $p(s)$, $p(s|x, c)$, and $p(f)$. Hence, the goal can be achieved by the following steps:

(1) Perform convolution and pooling on $I(u, v)$.
(2) Determine a mixed feature space φ_{mix} or a mixed image space, as shown in Equation (11).

By using the $L2$-norm, the mixed image space can be derived by the following equation:

$$\phi_{mix} = \{(i_1, i_2)|Malanobis(f(i_1), f(i_2)) \leq \varepsilon\}, \tag{11}$$

where i_1 and i_2 are two different images; f is the feature extracted from the images; and ε is an empirical value initialized within $[0.3, 0.5]$, as we need 90% error rate hits when the similarity is in this interval.

(3) Decide $p(x|s, \theta)$. The term $p(x|s, \theta)$ is in general obtained by integrating over the hidden variables π and s.

$$p(x|s, \theta) = \int p(\pi|\theta)(\prod_{n=1}^{N} \sum_{s_n} p(s_n|\pi)p(x_n|s_n, \theta))d\pi \tag{12}$$

As defined previously, θ represents the parameters to be already learnt with $\theta = (\mu,\Sigma)$ and s_n is the nth scene. When a scene is given or the distribution of the scene is fixed, we solve Equation (12) by maximizing the log of likelihood function; thus we have

$$\begin{aligned} ln\, p(x|\pi,\mu,\sum) &= \sum_{n=1}^{N} ln\{\sum_{k=1}^{K} \pi_k N(x_k|\mu_k,\sum\_k)\} \\ &= -\frac{ND}{2}(ln\, 2\pi) - \frac{N}{2} ln\,|\sum| - \frac{1}{2}\sum_{n=1}^{N}(x_n-\mu)^T\Sigma^{-1}(x_n-\mu), \end{aligned} \tag{13}$$

where μ and Σ are determined by the derivative of the log likelihood with respect to μ. In fact, μ and Σ are estimated by $\hat{\mu}$, $\hat{\Sigma}$, as follows:

$$\hat{\mu} = \frac{1}{N}\sum_{n=1}^{N} x_n, \hat{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(x_n-\hat{\mu})(x_n-\hat{\mu})^T. \tag{14}$$

Because the parameters s and θ are coupled, the variational approximation technique is used to maximize the log likelihood of the data and to minimize the Kullback–Leibler divergence between the approximation and the true posteriors. Here, we use the distribution $q(.)$ to approximate the true distribution. Then we optimize Equation (12) by maximizing the lower bound of the likelihood. The variational lower bound on the marginal likelihood for a single labeled face image can be computed as follows:

$$\begin{aligned} &\log p(x|s,\theta) \\ &\geq \sum_s q(\pi|s)\log p(\pi,s,x|\theta) - \sum_s q(\pi,s)\log q(\pi,s) \\ &= E_q[\log p(\pi,s,x|\theta)] - E_p[\log xq(\pi,s)]. \end{aligned} \tag{15}$$

By defining $L(\gamma,x,\theta)$ for the right-hand side (R.H.S.) of the above equation, we have

$$\log p(x|\theta) = L(\gamma,x,\theta) + KL(q(\pi,s|\gamma)||p(\pi,s|x,\theta)), \tag{16}$$

where $KL()$ is the Kullback–Leibler divergence between any two distributions, $q(\pi,s|\gamma)$ is an arbitrary variational distribution, and γ is the above-mentioned latent variable. The second term on the R.H.S. of the above formula stands for the KL distance of the two probability densities. As shown in Figure 8, the original size of the training datasets is finally enlarged according to the scene inference procedure, and then the enriched datasets are fed into a model training process in order to obtain a model for the next verification task.

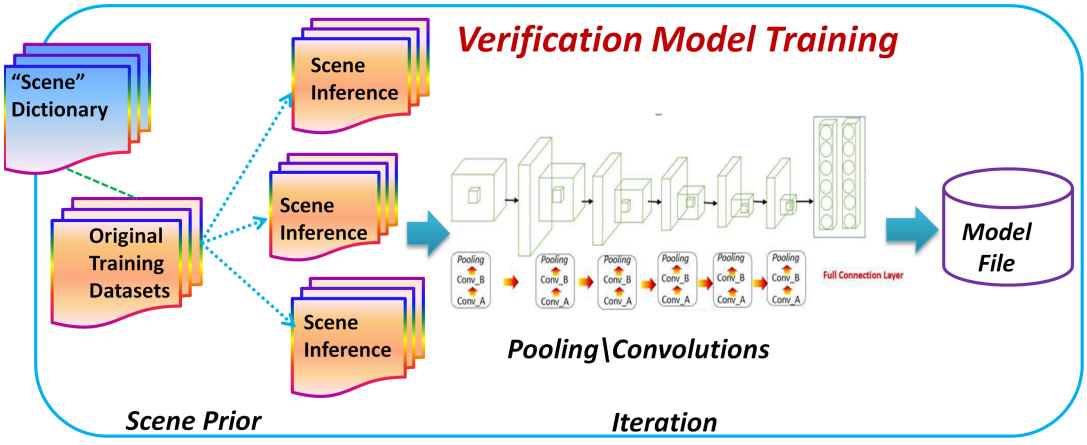

Figure 8. The illustration of the procedure for training a verification model on the basis of scene inference.

3.3. Hyperparameter Optimization

By simply considering the $Softmax$ loss or the Euclidean loss, the loss function is able to be affected by noisy scenes. Thus, we propose a ground truth distribution model based on the mixture Gaussian model. We denote the NN energy by E_{in} and the distribution energy by E_{scene}. Then we form the following distribution:

$$\begin{aligned} E &= E_{in} + E_{scene} + \zeta \\ &= \tfrac{1}{2}||f\left(I(x)\right) - y||_2^2 + \alpha KL(p(x|\theta), q(x|\theta)), \end{aligned} \tag{17}$$

where $I(x) = (I_1, I_2, \ldots, I_N)$ are the input images; ζ is a penalty item, which is a constant here; and α is the coefficient factor. The energy function has two main components. The first component is a plain NN error, which aims to make the predicted result much closer to the ground truth. However, if we only have this term, a large dataset is required in order to achieve a relatively higher accuracy. The training process also tends to be out of control after learning a batch size of the dataset and then results in an overfitting. As we know that the CNN was originally designed to learn the general features in a class, it cannot fit the variance of the training images. Thus we add the second term to generate the variance for the existing images, and we enlarge the variance by generating different scenes for a given image via the scenes' transformation. The bound becomes tight if and only if $p(x) = q(x)$. In addition to maximizing the log likelihood of the dataset, the conditional constraint (as shown in Equation (17)) could also select parameters that minimize the Kullback–Leibler divergence between the approximation and true posteriors. For implementation, the energy is approximated by an expected maximum (EM)-based variational learning method.

3.4. Overlapping Distributions' Transform

The validation dataset falls into two parts: one is data to be recognized easily with relatively discriminative feature spaces; the other contains some of the less familiar images that lie in a crossing space and that cannot be directly classified. We name all such image datasets in the crossing space as ϕ_{mix}; thus,

$$\begin{aligned} \phi(p_k, q_k) &= \{(x_i, x_j)|KL(p(s_k|x_{i)}, 1(s_k|x_j) \geq \eta)\} \\ &= \{(x_i, x_j)|Malanobis(s_k|x_i, s_k|x_j) \geq \varepsilon)\} \end{aligned} \tag{18}$$

$$\phi_{mix} = \bigcup_{k=1}^{K} \phi(p_k, q_k), \tag{19}$$

where η or ε is the threshold to determine the marginal of the overlapping space. Before we have the samples (refer to Section 3.1) to be generated for the next iteration and take the scene as a prior for the next time, we treat the difference in the pair as the scene variance. Then we can enlarge the distance in the overlapping classes' space with the dataset s and the scenes in s. Finally, we obtain the easily wrongly labeled faces and scenes for the overlapping space. However the difficulty is that the selected features are not sufficient to discriminate between the current varying scenes. Luckily, we are motivated by the human visual system in which, when people cannot recognize one person by the given low-level features, they would try to find more discriminative high-level features. Usually they extract the high-level features or semantic features. Similarly, we use the face region generated by the CRF to extract high-level semantic features for the recognition task.

The semantic feature extraction procedure is as shown in Figure 9. In this procedure, a semantic feature is determined by the CNN extracted feature on the basis of the face region produced by the CRF. As shown in Figure 9, the extracted features will be used as input for subsequent face-verification tasks. We note that the main purpose for the scene dictionary exploited in this context is to help the transformation of a given face into a specified scene.

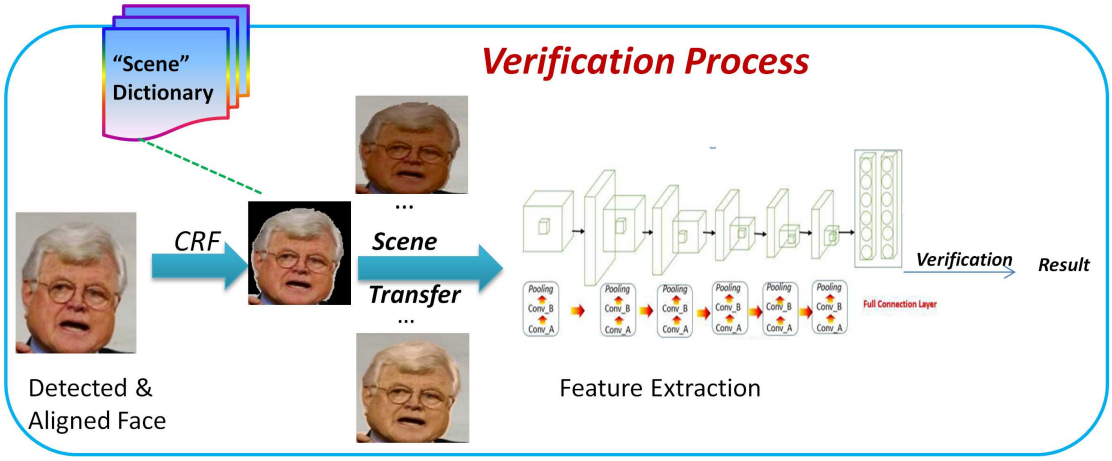

Figure 9. The semantic features' extraction procedure.

3.5. Scene Backward Propagation

Now we describe how to obtain the scene distribution among the images in the training dataset. As for Equation (17), the gradient of E_{scene} can be expressed as follows:

$$\begin{aligned} \frac{\partial E_{scene}}{\partial s_k} &= \frac{\partial E_{scene}}{\partial f_l}\frac{\partial f_l}{\partial s_k} = \delta, \\ \frac{\partial E_{scene}}{\partial \theta} &= \frac{\partial E_{scene}}{\partial f_l}\frac{\partial f_l}{\partial s_k}\frac{\partial s_k}{\partial \theta}, \end{aligned} \tag{20}$$

$$= \delta \Delta p(x|s_k) = \delta \prod_{k=1}^{K}, N(\Delta\mu_k, \Delta\Sigma_k), \tag{21}$$

where l represents the layer in the NN, θ is a parameter of the scene distribution, and f_l is the output activation function. The scene is propagated by the overlapping space. Then we can extract the scene distribution in Equation (22) and transform it back to the first layer of the NN.

$$p(x^{t+1}) = p(x^t) + \delta \prod_{k=1}^{K} N(\Delta\mu_k, \Delta\Sigma_k), \tag{22}$$

where μ_k is the learning rate for scene propagation, and δ is the gradient of a scene in pairs. The whole process is listed in the algorithms in the appendix.

4. Experiments and Results

For face detection, a face detector is used on each image, and a tight bounding box around each face is generated. These face thumbnails are resized and aligned to a size of 141 × 165 pixels. At the beginning, we use a CNN model to train the deep features. The training- and validation-related topics are given in the following sub-sections.

4.1. Datasets and Evaluation

The new method was evaluated for a face-verification task; that is, given a pair of two face images, a squared Cosine metric $\tau(x_i, x_j)$ was used to determine whether the two images were the same or a different person. We used CASIAWebFace [25], which contains 10,575 subjects and 494,414 face images used to train our model. As for the evaluation, LFW [10], which contains 13,233 images with 5749 identities collected from the Web with large variations in pose, age, expression, illumination, and so forth, and YTF [3], a video dataset containing 3425 videos of 1595 different subjects downloaded

from YouTube, were used. We considered the unsupervised protocol and followed the standard setting as described in [26]; in addition to the verification accuracy (Acc.), we used the ROC Receiver Operating Characteristic Curve to evaluatethe performance. We conducted the evaluation under the following setting: a cross-dataset validation, in which external data (CASIAWebFace) exclusive to LFW/YTF was used for training in order to show the generalization ability across different datasets. The datasets we used for validation were the LFW dataset in view #2, which has 6000 pairs, and the YTF face dataset, which has 5000 face pairs. As for the scene dictionary learning, we also exploited CASIAWebFace as the training dataset.

4.2. Training Process for the New Model

To find the differences with enough training images (ETI: training datasets augmented using scene transformation) and without enough training images (WETI: using the raw data as the training data), 250,000 iterations were run on both the ETI and WETI datasets. The observed loss (error) is illustrated in Figure 10. The green curve indicates the loss convergence speed for WETI and the red indicates that for ETI. One can observe that the convergence speed for ETI was quite fast. As discussed in the above section, Table 2 illustrates the feasible hyperparameters for our proposed model.

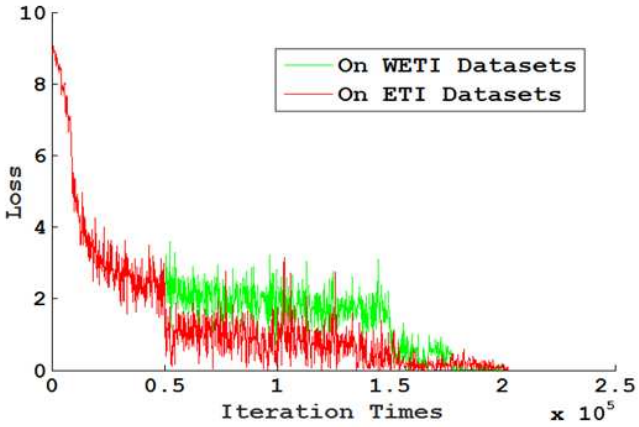

Figure 10. The loss on ETI datasets and WETI datasets with the same network model

Table 2. Convolutional neural network (CNN) models and the parameters.

Name	Type	Stride	Output	#P
Conv11	Conv	(3, 3, 1)	(100, 100, 32)	280
Conv12	Conv	(3, 3, 1)	(100, 100, 64)	18,000
Pool1	Maxpooling	(2, 2, 2)	(50, 50, 64)	
Conv21	Conv	(3, 3, 1)	(50, 50, 64)	36,000
Conv22	Conv	(3, 3, 1)	(50, 50, 128)	72,000
Pool2	Maxpooling	(2, 2, 2)	(25, 25, 128)	
Conv31	Conv	(3, 3, 1)	(25, 25, 96)	108,000
Conv32	Conv	(3, 3, 1)	(25, 25, 192)	162,000
Pool3	Maxpooling	(2, 2, 2)	(13, 13, 192)	
Conv41	Conv	(3, 3, 1)	(13, 13, 128)	216,000
Conv42	Conv	(3, 3, 1)	(13, 13, 256)	288,000
Pool4	Maxpooling	(2, 2, 2)	(7, 7, 256)	
Conv51	Conv	(3, 3, 1)	(7, 7, 160)	360,000
Conv52	Conv	(3, 3, 1)	(7, 7, 320)	450,000
Pool5	AVGpooling	(7, 7, 1)	(1, 1, 320)	
Dropout	Dropout		(1, 1, 320)	3,305,000
Fc6	Fullyconnect		10,575	
Cost1	Softmax		10,575	
KL	Generate		10,575	2000
Total				5,017,000

4.3. The Semantic Features' Extraction

As mentioned in the model description section, the semantic features are extracted after the CRF–RNN process. Essentially speaking, CRF–RNN is exploited to build a relationship between a scene and face by extracting the face regions from the scene backgrounds in which they co-exist (refer to Figure 6). Then the proposed CNN model can express the face feature with scene information in a semantic way. With the benefit of the representations from intermediate layers, we can turn any face image into a vector containing important scene attributes of the face. As shown in Figure 11, it is indicated that the semantic feature distribution for the same person tends to be very close (middle); on the contrary, the semantic feature distribution for different people varies considerably, as shown in Figure 11 (right), although the faces are transferred to the same scene level.

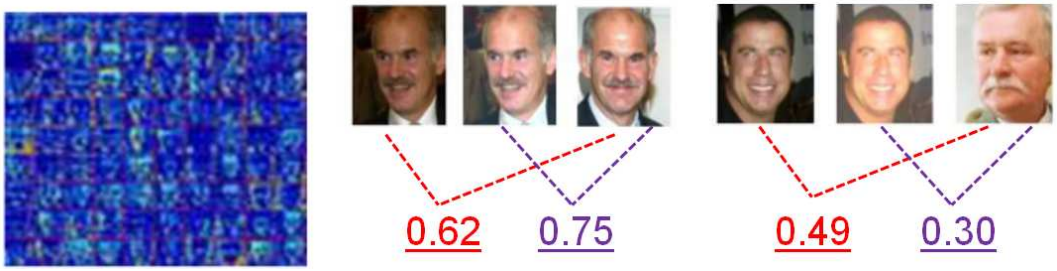

Figure 11. The extracted semantic feature and their responses to individuals, the value stands for cosine similarity

4.4. Distribution of Semantic Features

We randomly selected 10,000 images from the CASIAWebFace dataset and then extracted their features using both the VGGFace model [27] and our proposed model. Because it is difficult to visualize the distributions of extracted features directly, *tSNE* [28] was usedfor reducing the extracted features' dimension from a high dimension (10,575) to a low dimension (only 2). In this way, the differences in the distribution could be projected onto a two-dimensional plane. As shown in Figure 12, the same colors represent the same individual, and different colors represent different people. According to Figure 12 (left), the VGGFace model had a nearly circular feature distribution; that is, the extracted features of each category tended to gather together in a relatively small radius.

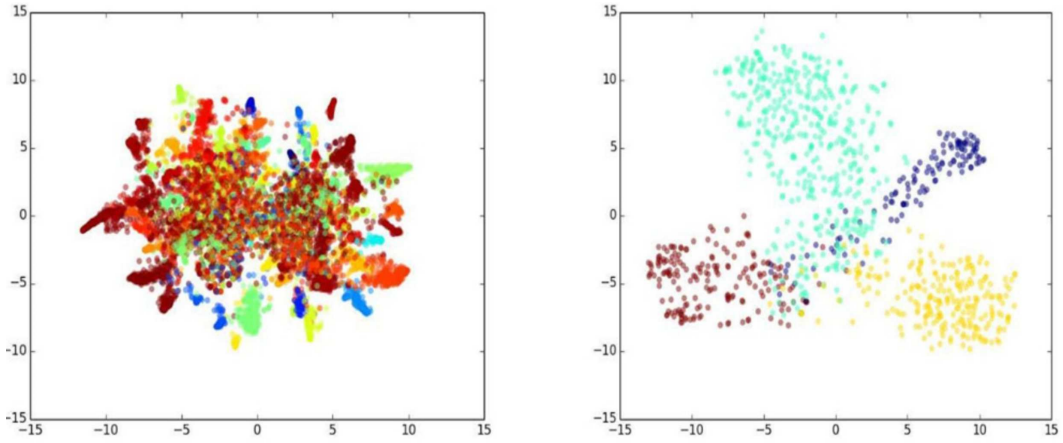

Figure 12. The features extracted by VGGFace.

As we can see, about 50% of the categories gathered in a compact form, while the rest were scattered (see Figure 12 (left)). Zooming in on the details, there was extensive overlapping among the features from the selected four individuals, as observed in Figure 12 (right).

Next, we look at those features extracted by our proposed model. As we see in Figure 13 (left), the distribution of respective individuals looked more like a strip. It is also observed that about 80% of the categories were dispersed with a relatively larger spacing. Compared to the zoomed-in distribution property shown in Figure 13 (right), Figure 13 (right) illustrates less overlapping, and the distribution for inner classes was much more uniform along certain directions than that the VGGFace model produced.

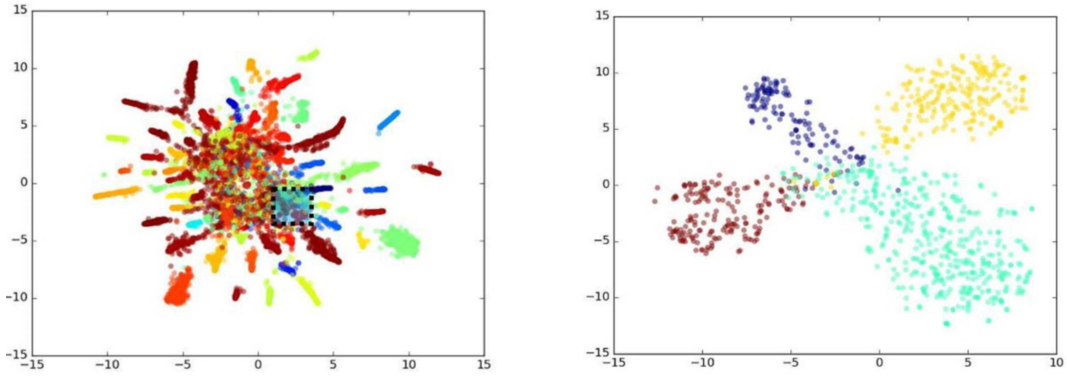

Figure 13. The features extracted by our method.

4.5. Comparison with Other Models

In order to validate the performance of the proposed model, an unsupervised protocol [10] was used on both the LFW and YTF datasets. The reason we chose this protocol for comparison was that a strict generalization ability is necessary for the face-verification task in an open scene. Firstly, we compare the Casia [25] and VGGFace [27] models with our model. For the training step, all the models were trained on the CASIAWebFace dataset, and then these three models were validated on the YTF dataset. For the verification step, the proposed model needs the face pair to be transformed into the same scene (see Figure 9). The ROC curves for the three models on the unsupervised protocol were plotted, as shown in Figure 14. One can see that the performance of our proposed model on the LFW dataset was better than that of the Casia [25] model (Area Under Curve (AUC) gap of 0.0427), but the AUC of the VGGFace model was 0.0301, which was slightly better than for our model. However, the VGGFace model has an unrestricted protocol, and a much larger dataset for training is inevitable.

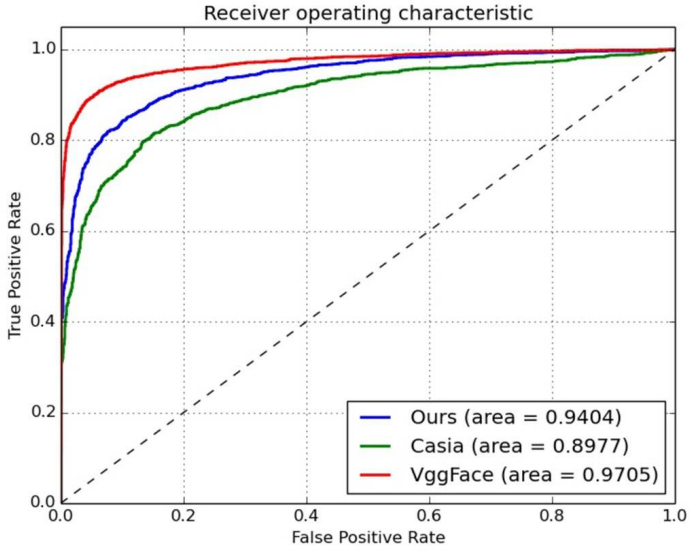

Figure 14. ROCs on LFW dataset with the unsupervised protocol.

Secondly, we compared the AUC with the top 10 of the leading board [9,25,29,30] for the face-verification task on the LFW dataset (see Figure 15). We also compared several up-to-date leading models, such as Casia [25], DeepFace [18], OpenFace [31], and VGGFace [27], in an unsupervised protocol for the YTF dataset (see Figure 16). As observed in Figures 15 and 16, our proposed model performed the best, and its AUC was 0.0075-fold higher than that of the VGGFace model.

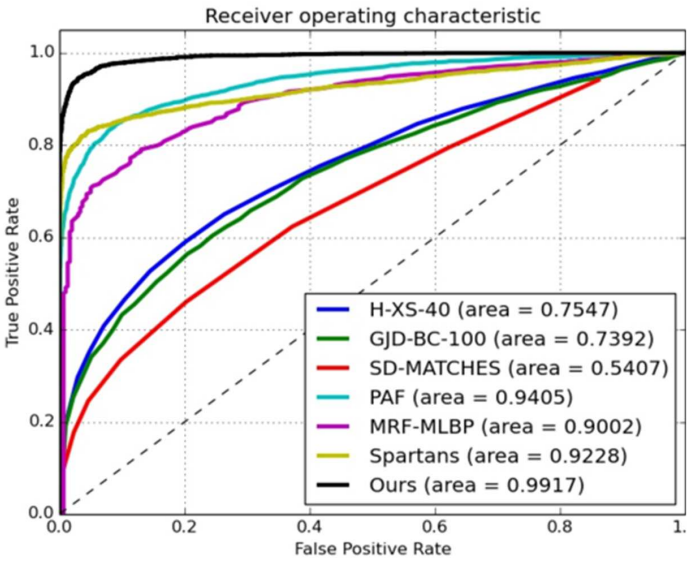

Figure 15. ROCs with different unsupervised models on LFW dataset.

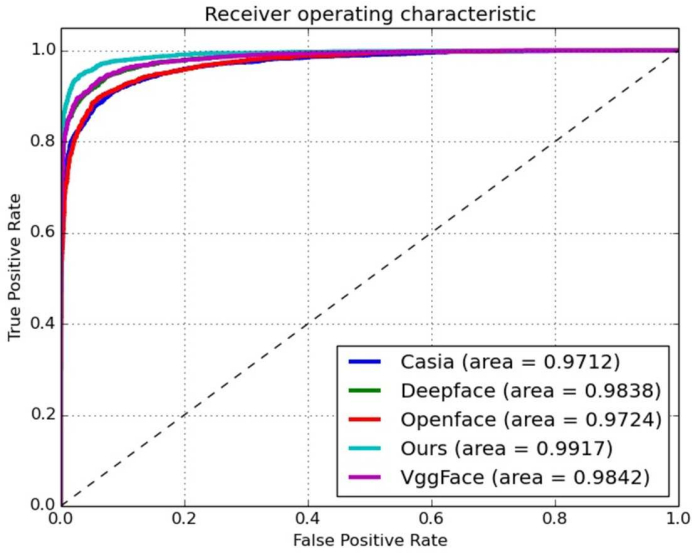

Figure 16. ROCs with unsupervised verification protocol on YTF dataset for some up-to-date models.

Thirdly, we summarize various protocols, training datasets, and networks in Table 3. Some performances listed in Table 3 are from the related publications. Here, it is shown that under the unsupervised protocol, our model achieved the best result on both the LFW and YTF datasets. In order to see the benefits much more clearly, we also compared the performances between ETI and WETI, which illustrated that ETI could gain an advantage of about 0.0062 over WETI on the LFW dataset, as shown in Figure 17.

Table 3. Performance on LFW and YTF databases.

Method	LFW	YTF	Protocol	Images	Networks
CNN-3DMM estimation [32]	92.35%	88.80%	Unrestricted	0.5 M	1
Casia [25]	97.73%	92.24%	Unrestricted	1.0 M	1
Pose/shape/expression augmentation [33]	98.07%	N/A	Unrestricted	2.5 M	1
VGGFace [27]	98.95%	97.30%	Unrestricted	2.6 M	1
Discriminative [34]	99.28%	94.90%	Unrestricted	0.7 M	1
SphereFace [35]	99.42%	95.00%	Unrestricted	0.5 M	1
DeepID [1:3] [19]	99.53%	93.20%	Unrestricted	0.3 M	200
CCL with AAM [36]	99.58%	95.28%	Unrestricted	0.5 M	1
Facenet [17]	99.63%	99.63%	Unrestricted	200 M	1
GTNN [37]	99.65%	N/A	Unrestricted	6.2 M	2
Baidu [16]	99.77%	N/A	Unrestricted	1.3 M	10
LBPNet [38]	94.04%	N/A	Unsupervised	0.5 M	1
Deepface [18]	95.20%	91.40%	Unsupervised	4 M	1
Casia [25]	97.30%	90.60%	Unsupervised	0.5 M	1
MRF-FUSION-CGKDA [29]	98.94%	93.20%	Unsupervised	0.5 M	5
AM-Softmax w/o FN [39]	99.12%	N/A	Unsupervised	0.5 M	1
Ours	99.2%	94.30%	Unsupervised	0.5 M	1

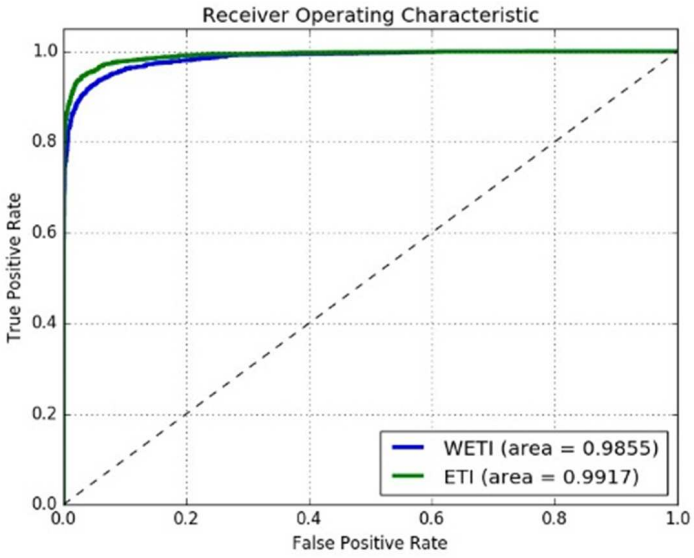

Figure 17. ROCs with unsupervised verification protocol on LFW dataset for without enough training images (WETI) and enough training images (ETI).

5. Conclusions and Discussions

In this paper, a new deep learning model is proposed for face verification and is essentially used to solve the small number of training samples requested by current deep learning networks. The main idea is to use scene transfer learning to generate more images for validation. The proposed model was evaluated from multiple perspectives. With the unsupervised protocol, our model performed better than the existing leading algorithms on the LFW dataset. According to Figure 16, its performance was at least 0.7% higher than that of other models under an unsupervised verification protocol. As illustrated by the ROCs, the VGGFace model slightly outperformed our proposed model on the YTF dataset; however, our model performed much better than the VGGFace model on the LFW dataset. The key point is that our model has much greater generalization capability, as it can deduce many more scenes for the training dataset. As observed in Table 3, it not only lists the performance and protocol, but also the training data size and network used by each model. As we can see, although our model was superior to DeepFace, with AUCs greater than those of LFW and YTF by 3.9% and 2.9%,

respectively, the size of the training dataset was just 1/8 that of DeepFace. That is to say, we required much less data for our proposed model to train a better CNN model for the face-verification task. Even with the similar training data size (0.5 Million), in comparison with the CASIAWebFace model on the supervised protocol, our model achieved better results than LFW and YTF by 1.44% and 2.06%, respectively. The proposed model has only a relatively simple network structure. In contrast, the Baidu model uses 10 networks and had only about a 0.6% improvement with the **unrestricted protocol**. Although our model has only a simple network, it still achieved a steady performance for both LFW and YTF with an **unsupervised protocol**. The DeepID3 model has a much more complicated network (consisting of 200 models), but it achieved only 0.3% higher than the proposed model for LFW, even with a supervised protocol. The proposed model also outperformed the DeepID3 model by 0.9% for YTF with an unsupervised protocol. Hence, we can draw a conclusion that the proposed model can achieve a better performance than the state-of-the-art models with a relatively small amount of training data. As for the dataset LFW, there were several face pairs judged with great difficulty by the other models that could be distinguished between by the newly proposed model. As shown in Figure 18, the numbers stand for the cosine distance between a given face pair. When we set the threshold τ equal to 0.50, these face pairs could then be identified with less difficulty.

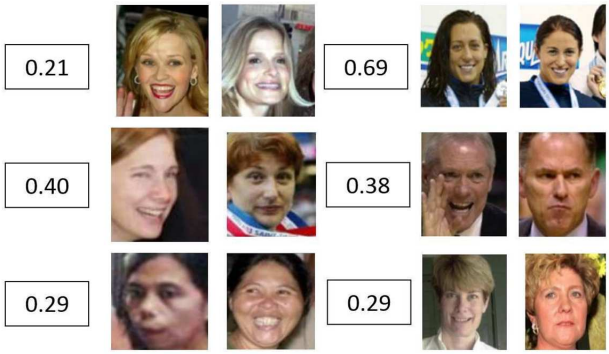

Figure 18. The face pairs in LFW were identified with great difficulty by other methods but could be distinguished between easily by our method.

However, there were still some face pairs that our model failed to verify properly. Figure 19 shows those pairs that were falsely accepted by our proposed model, and Figure 20 illustrates those pairs that were falsely rejected by the proposed model.

Figure 19. The pairs falsely accepted by the proposed model.

As we can see, the key reason for the failure of our model was likely that the face images were subject to facial expressions, shelter, and other factors. Our next work will aim for profound research on the facial expression scene and make our model capable of transforming all facial expression scenes into uniform scenes.

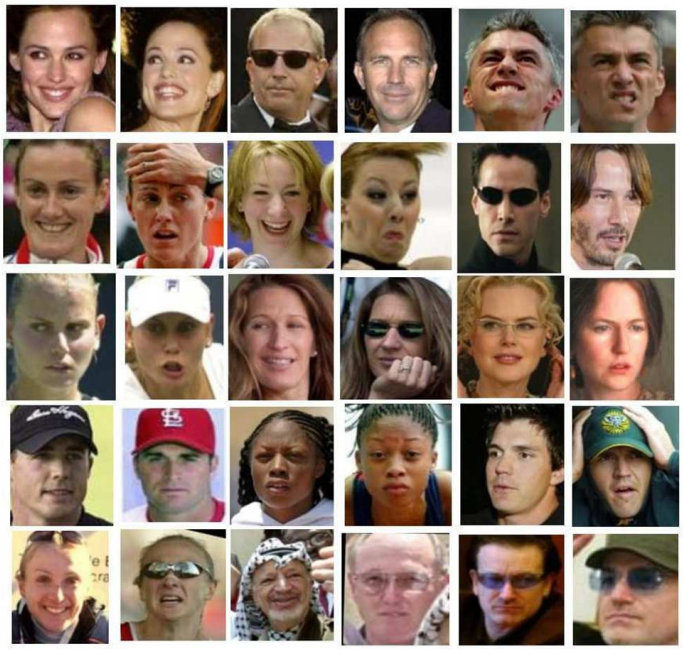

Figure 20. The pairs falsely rejected by the proposed model.

Author Contributions: Conceptualization: H.W. and W.L.; methodology: H.W. and W.S.; software: N.S. and Y.W.; validation: H.W., N.S., and H.P.; formal analysis: W.L.; investigation: H.W.; resources: Y.W.; data curation: H.P.; writing of original draft preparation: H.W.; writing review and editing: W.L.; visualization: H.P.; supervision: W.L.; project administration: H.W.; funding acquisition: W.L.

Funding: This research was funded by the North China University of Technology under Grant No. 2018-09-001.

Acknowledgments: The authors would like to thank Jiang Huang for his help with the face detection and Yehe Cai and Junyi Du for their help with the scene models.This work was partially supported by a grant from the National Natural Science Foundation of China (No.61573019, No.61703006, and No.61602321).

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Scene Dictionary Learning

Algorithm A1 Scene dictionary learning

Input X datasets of detected and aligned faces
Output S: scene dictionary for all individuals
Begin
Pretrain a CNN model on the basis of X by exploiting the default hyperparameters.
Randomly initialize a scene entry matrix S whose size is $M \times N$ (M is the number of people, and N is the maximum length of faces for a given person).
for all $face_i$ in X **do**
    $X_{new} \longleftarrow X$ by CRF
    Extract CNN feature ϕ_i for $face_i$
end for
for each $person_i$ in X **do**
    Initialize the value s_p with first scene
    for each $i \in [1, N]$ **do**
        if Malanobis(ϕ_i,ϕ_j) $\leq \epsilon$ **then**
            $face_j \in s_i$
        else
            $face_j \in s_{i+1}$
            $s_p \longleftarrow s_p \cup s_{i+1}$
        end if
    end for
    $S \longleftarrow s_p$
end for
return S
End

Appendix B. Scene Inference and Model Training

Algorithm A2 Scene inference and model training

Input : X datasets of detected and aligned faces; S: scene dictionary for all individuals
Output : Enlarged datasets X_{new} and a new trained CNN model

Begin
Sort the elements in S
for each $j \in [1, S.length]$ **do**
    Express the extracted faces' features with a mixture Gaussian distribution $p(f, s)$
    for each $k \in [1, K]$ **do**
        Determine π_k of S_k
        Estimate $\hat{\mu}$ and $\hat{\sum}$
        for each person $n \in [1, N]$ **do**
            $\ln p(x_n|\pi, \mu, \sum)$
            $= L(\gamma, x_n, \theta) + KL(q(\pi, s|\eta)||p(\pi, s|x_n, \theta))$
            $x_{s_{new}} \leftarrow \ln p(x_n|\theta)$
            update X_{new}
        end for
    end for
end for
//EM iteration
for all $t = 1$ to $iter < T$ (time of total iterations) **do**
    First, fix E_{scene} to learn the CNN network
    Extract features:
    $(f_x, f_y) = CNN(x_i, x_j)$

    "Conv":
    $x_j^l = f_{cnn}(x_i^{l-1})$

    "Pooling": $x_j^l = f_{pooling} x(x_j^{l-1})$

    Backward Propagation:
    //Calculate the gradient of E_{scene}
    $\delta = \frac{\partial E_{scene}}{\partial s_k}$

    //Transform back to the first layer of NN
    $p(x_{t+1}) = p(x_t) + \delta \prod_{i=1}^{n} N(\Delta\mu_k, \Delta\Sigma_k)$

end for
End

References

1. Sun, Y.; Wang, X.; Tang, X. Deep learning face representation from predicting 10,000 classes. In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), Columbus, OH, USA, 23–28 June 2014; pp. 1891–1898.
2. Sun, Y.; Wang, X.; Tang, X. Sparsifying neural network connections for face recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas Valley, NV, USA, 26 June–1 July 2016.
3. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. Web scale training for face identification. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.

4. Zhu, Z.; Luo, P.; Wang, X.; Tang, X. Deep Learning Identity Preserving Face Space. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 113–120.
5. Tran, L.; Yin, X.; Liu, X. Disentangled Representation Learning GAN for Pose Invariant Face Recognition. In Proceedings of the 2017 IEEE Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
6. Chen, W.; Liu, C.H. Transfer between pose and expression training in face recognition. *Vis. Res.* **2009**, *49*, 368–373. [CrossRef] [PubMed]
7. Chen, B.C.; Chen, C.S.; Hsu, W.H. Cross age reference coding for age invariant face recognition and retrieval. In *Computer Vision ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; pp. 768–783.
8. Cheng, Y.; Jiao, L.; Cao, X.; Li, Z. Illumination insensitive features for face recognition. *Vis. Comput.* **2017**, *33*, 1483–1493. [CrossRef]
9. Ruiz del Solar, J.; Verschae, R.; Correa, M. Recognition of Faces in Unconstrained Environments: A Comparative Study. *EURASIP J. Adv. Signal Process.* **2009**, *2009*, 184617. [CrossRef]
10. Huang, G.B.; Learned-Miller, E. *Labeled Faces in the Wild: Updates and New Reporting Procedures*; (UM-CS-2014-003), Technical Report; University of Massachusetts Amherst: Amherst, MA, USA, 2014.
11. Deng, W.; Zheng, L.; Ye, Q.; Murphy, K.; Kang, G.; Yang, Y.; Jiao, J. Image-Image Domain Adaptation with Preserved Self-Similarity and Domain-Dissimilarity for Person Re-identification. *arXiv* **2017**, arXiv:1711.07027.
12. Fei, L.; Perona, P. A Bayesian hierarchical model for learning natural scene categories. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 2, pp. 524–531.
13. Chen, L.C.; Barron, J.T.; Papandreou, G.; Murphy, K.; Yuille, A.L. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas Valley, NV, USA, 26 June–1 July 2016; pp. 4545–4554.
14. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas Valley, NV, USA, 26 June–1 July 2016; pp. 770–778.
15. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. *arXiv* **2014**, arXiv:1409.4842.
16. Liu, J.; Deng, Y.; Bai, T.; Huang, C. Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv* **2015**, arXiv:1506.07310.
17. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. *arXiv* **2015**, arXiv:1503.03832.
18. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. Deepface: Closing the gap to human level performance in face verification. In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), Columbus, OH, USA, 23–28 June 2014; pp. 1701–1708.
19. Sun, Y.; Liang, D.; Wang, X.; Tang, X. DeepID3: Face Recognition with Very Deep Neural Networks. *arXiv* **2015**, arXiv:1502.00873.
20. Raudys, S.; Pikelis, V. On dimensionality, sample size, classification error, and complexity of classification algorithm in pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **1980**, *3*, 242–252. [CrossRef]
21. Salakhutdinov, R.; Tenenbaum, J.B.; Torralba, A. Learning with Hierarchical Deep Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1958–1971. [CrossRef] [PubMed]
22. Zheng, S.; Jayasumana, S.; Romera Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; Torr, P.H.S. Conditional random fields as recurrent neural networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1529–1537.
23. Zhang, B.; Perina, A.; Li, Z.; Murino, V.; Liu, J.; Ji, R. Bounding multiple gaussians uncertainty with application to object tracking. *Int. J. Comput. Vis.* **2016**, *118*, 364–379. [CrossRef]
24. Wolf, L.; Hassner, T.; Maoz, I. Face recognition in unconstrained videos with matched background similarity. In Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), Colorado Springs, CO, USA, 20–25 June 2011; pp. 529–534.
25. Yi, D.; Lei, Z.; Liao, S.; Li, S.Z. Learning Face Representation from Scratch. *arXiv* **2014**, arXiv:1411.7923.
26. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Training very deep networks. *arXiv* **2015**, arXiv:1507.06228.

27. Parkhi, O.M.; Vedaldi, A.; Zisserman, A. Deep face recognition. In Proceedings of the 2015 British Machine Vision Conference, Swansea, UK, 7–10 September 2015.
28. Van Der Maaten, L. Accelerating tSNE Using Tree based Algorithms. *J. Mach. Learn. Res.* **2014**, *15*, 3221–3245.
29. Arashloo, S.R.; Kittler, J. Class Specific Kernel Fusion of Multiple Descriptors for Face Verification Using Multiscale Binarised Statistical Image Features. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 2100–2109. [CrossRef]
30. Xu, J.F.; Luu, K.; Savvides, M. Spartans: Single Sample Periocular Based Alignment Robust Recognition Technique Applied to Non Frontal Scenarios. *IEEE Trans. Image Process.* **2015**, *24*, 4780–4795. [CrossRef] [PubMed]
31. Amos, B.; Ludwiczuk, B.; Satyanarayanan, M. OpenFace: A General Purpose Face Recognition Library with Mobile Applications; Technical report, CMU CS 16 118; CMU School of Computer Science: Pittsburgh, PA, USA, 2016.
32. Tran, A.; Hassner, T.; Masi, I.; Medioni, G. Regressing Robust and Discriminative 3D Morphable Models with a very Deep Neural Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
33. Masi, I.; Tran, A.T.; Leksut, J.T.; Hassner, T.; Medioni, G.G. Do We Really Need to Collect Millions of Faces for Effective Face Recognition? *arXiv* **2016**, arXiv:1603.07057.
34. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In Proceedings of the European Conference on Computer Vision 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 499–515.
35. Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. Sphereface: Deep hypersphere embedding for face recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
36. Qi, X.; Zhang, L. Face Recognition via Centralized Coordinate Learning. *arXiv* **2018**, arXiv:1801.05678.
37. Hu, G.; Yang, H.; Yuan, Y.; Zhang, Z.; Lu, Z.; Mukherjee, S.S.; Hospedales, T.; Robertson, N.M.; Yang, Y. Attribute enhanced face recognition with neural tensor fusion networks. In Proceedings of the International Conference on Computer Vision (ICCV 2017), Venice, Italy, 22–29 October 2017.
38. Xi, M.; Chen, L.; Polajnar, D.; Tong, W. Local binary pattern network: A deep learning approach for face recognition. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3224–3228.
39. Wang, F.; Liu, W.; Liu, H.; Cheng, J. Additive Margin Softmax for Face Verification. *arXiv* **2018**, arXiv:1801.05599.