Retraction Notice

The Editor-in-Chief and the publisher have retracted this article, which was submitted as part of a guest-edited special section. An investigation uncovered evidence of systematic manipulation of the publication process, including compromised peer review. The Editor and publisher no longer have confidence in the results and conclusions of the article.

AMS, SDC, SCB, and SP either did not respond directly or could not be reached.

Robust face recognition using multimodal data and transfer learning

**Akhilesh Mohan Srivastava,* Sai Dinesh Chintaginjala,
Samhit Chowdary Bhogavalli, and Surya Prakash**
Indian Institute of Technology Indore, Department of Computer Science and Engineering,
Indore, Madhya Pradesh, India

Abstract. In recent years, technological advancements in face recognition have sparked numerous research efforts and have opened up a variety of applications in fields such as security, access control, and identity verification. The accuracy of two-dimensional (2D) face recognition is not up to the mark when used in highly illuminated or dark environments. Further, its vulnerability to spoofing makes it a poor choice for security applications. These problems can be easily resolved with the help of three-dimensional (3D) face recognition. However, 3D data comes with its own set of issues and challenges. The resources and computational power required to collect and process 3D data are found to be heavy. Most recent signs of progress in this area have been achieved by training deep neural networks on large datasets, which is computationally costly and time-consuming. To address these issues, instead of using 3D face data directly, we propose the use of a 2.5D representation of 3D face data along with registered 2D face images, which makes it relatively easy to work with in terms of computational power and time requirements. The paper proposes a robust face recognition approach using multi-modal data (2.5 face images along with 2D face images) and transfer learning. The proposed approach is built on ResNet-34 and Siamese network models. The ResNet-34 network is first trained on 2D face images. Further, by reusing the pretrained ResNet-34 network model on 2D images, we perform transfer learning to produce a network that can make accurate predictions on 2.5D images. The final outcome of the face recognition is achieved by fusing the results obtained on 2D and 2.5D data. The proposed approach has been validated on the University of Notre Dame 3D face dataset (ND-Collection D). The experimental analysis shows the effectiveness of the proposed technique. © *2022 SPIE and IS&T* [DOI: 10.1117/1.JEI.32.4.042105]

Keywords: biometrics; 3D face recognition; deep neural networks; Siamese network; residual network; data augmentation; depth images.

Paper 220930SS received Sep. 7, 2022; accepted for publication Nov. 29, 2022; published online Dec. 30, 2022.

1 Introduction

Face recognition is a form of personal identification. It is the technique of recognizing a person's face in an image and determining to whom it belongs. In the beginning, face recognition systems focused on retrieving facial landmarks from images, like the relative size and location of an individual's eyes, nose, cheekbone, and jaw. However, because these face quantifications are retrieved manually by computer specialists and researchers using face recognition software, these systems are highly subjective and prone to error.[1] Face recognition software generally uses computer algorithms to extract unique features from a person's face and uses them for recognition. Details such as eye distance or outline of the face are then transformed into a mathematical representation and are compared with data from other faces in a face recognition database. Many two-dimensional (2D) face recognition systems proposed over the last few decades have performed well in a controlled environment. Remarkably, the accuracy of 2D face recognition has enhanced dramatically since the advent of deep learning. However, the inherent limitations of 2D images, such as pose, expression, illumination variations, occlusion, and image quality-related

*Address all correspondence to Akhilesh Mohan Srivastava, phd1701101001@iiti.ac.in issues, continue to provide a challenge to these systems.[2] In most cases, 2D face recognition systems provide good results. Still, their performance decreases when the image used has poor contrast or illumination, change in orientation of the face, or presence of noise. Another major drawback of 2D face recognition systems is that they can be easily forged. These problems limit the usage of 2D face recognition systems in security-critical applications. Face recognition can be used to overcome these problems by utilizing three-dimensional (3D) data.

3D face recognition has become an active research topic in recent years as it is not affected by the limitations of 2D face recognition like pose, lightning conditions, and expressions.[3] 3D face images provide rich geometric information that gives more discriminative features.[4] 3D face models include more shape information than 2D images. Furthermore, in terms of scale, rotation, and lighting, 3D models are relatively unchanged.[5] Based on their feature extraction techniques, 3D face recognition systems can be divided into traditional and deep learning-based methods. Traditional ways of 3D face recognition include approaches that are based on iterative closest point (ICP)[6,7] matching and principal component analysis (PCA). By contrast, practically, most of the deep learning-based techniques used for 3D face recognition rely on pretrained networks that are subsequently fine-tuned using the converted data (for example, 2D images from 3D face images). Visual geometry group (VGGNet),[8] residual neural network (ResNet),[9] artificial neural networks (ANNs),[10] and recent lightweight convolutional neural networks (CNNs) like MobileNetV2[11] are popular deep learning-based facial recognition networks.

A 3D face image is an abstract representation of the face and can be represented as a depth image, point cloud, polygon mesh, and voxel. Figure 1 shows examples of these face representations. These representations have been used in the literature to extract the features and perform 3D face recognition. A depth image provides us with the object's "depth" or "z" information in the actual world in terms of intensity values. Surface modeling methods, like mesh, can obtain the points' topological information, such as connectivity between the points. In contrast, the data is unstructured in the case of point cloud representation, and the topological information is absent. The voxel image is a volumetric representation of each point where the change in volume size affects the resolution of the 3D image. Point clouds are the rawest form of 3D data and are the direct outcome of the object scanning process. In point clouds, a 3D object is represented by digitizing its surface in the form of an unordered set of data points.

Though 3D face recognition has been found superior to 2D face recognition, there are a few challenges with it as well. Two of the most challenging aspects of 3D face recognition are the acquisition of 3D images, which requires specialized hardware, and the time needed to process 3D data, which is found to be bulky. Due to the challenges of the scanning process, there is no availability of large datasets, whereas due to the bulky nature of 3D data, the training time of these systems is more. This limits the use of deep learning-based approaches in 3D face recognition.

Given the significance and vast implementations of 3D data in areas such as biometrics and object recognition in general, it becomes essential to address the issues faced during the training of the deep neural network, such as the availability of a large amount of 3D face data, complex

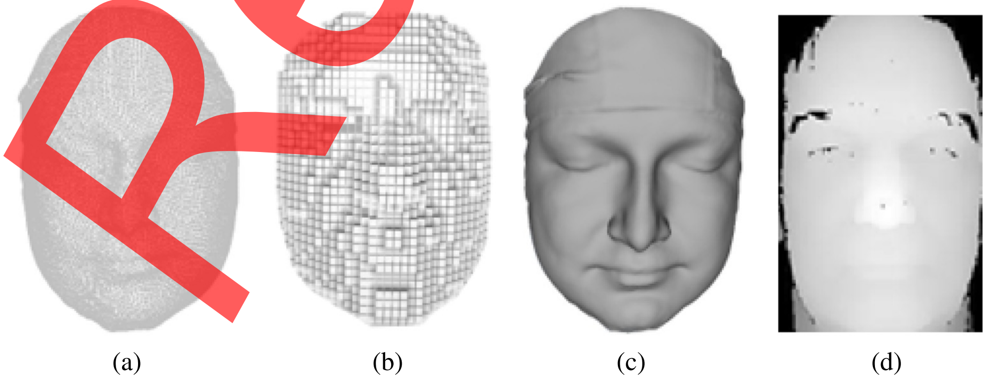


Fig. 1 Different representations of 3D face images used in recognition: (a) point cloud, (b) voxel, (c) polygon mesh, and (d) depth image.

preprocessing, and exhaustive training time. To mitigate the aforementioned issues associated with the 3D face data, this paper provides the following significant contributions.

- We propose a novel Siamese network-based deep learning architecture for face recognition, which utilizes ResNet-34 as a feature extractor and Siamese network architecture for recognition purposes.
- In this work, to mitigate the complexity involved with the processing of 3D face images and the training time, we utilize the 2.5D (depth image) representation of 3D face images along with 2D images.
- To handle the limited availability of 3D data and to avoid overfitting, we employ data augmentation because if the data is scarce, the model becomes so efficient at learning the features that it even learns noise (if present) of the training samples.
- We also utilize the transfer learning approach to reduce the training time and improve the overall testing accuracy as compared with without the transfer learning case.
- To further improve the overall testing accuracy, we propose the fusion of results obtained from proposed models that are trained on 2D and 2.5D face images, respectively.

The rest of the paper is organized as follows. Section 2 presents related work on face recognition carried out in the 2D, 2.5D, and 3D data domains. The proposed technique is described in the next section. Section 4 shows the experimental analysis and the results obtained from the experiments. Finally, the paper is concluded in the last section.

2 Related Work

This section discusses some of the significant existing works in face recognition. Although 2D face recognition has had a lot of success, changes in pose and lighting conditions still significantly impact accuracy.[12,13] The majority of researchers have moved to 3D face recognition due to its capacity to overcome similar restrictions and shortcomings of 2D face identification. Furthermore, when the position and illumination circumstances are the same, the geometric information offered by 3D face data results in better recognition performance than 2D.[3,14] Curvature-based algorithms have been tested on a small 3D face database by Wu et al.,[15] and they have achieved 100% identification accuracy. Gordon[16] has shown in a face recognition experiment that a combination of frontal and side views improves face recognition accuracy. Following that, more and more techniques in 3D face recognition were presented due to the emergence of 3D scanning equipment that was mainly based on laser and structured light technology. Blanz and Vetter[17] have introduced the 3D deformation model (3DMM) synthesis approach, and the model has been used for 3D face identification. At the time, because of the limitations of 3D scanning technology, their 3D deformation model had to be recreated from 2D images. The reconstruction of the 3D model necessitates a significant amount of computing. Many researchers agree that 3DMM is helpful for face recognition; however, the computational complexity of the reconstruction process limits their usability.[18–20] Pan et al.[21] proposed 3D face recognition using facial range data to extract multiple horizontal profiles. One disadvantage of this approach is that the recognition accuracy drops dramatically as the head posture changes. Zhang and Gao[22] have examined techniques and designed methods for 3D face recognition, including pose variations, and have experimented with the most significant angle, which can be recognized while the pose varies. In 3D facial recognition, Chua et al.[23] have employed point signatures. This method only uses the rigid portion of the face (under the eyebrow just above the nose) to deal with variations in facial emotions. The images utilized in the experimentation were taken from the expressions of six subjects, and the recognition rate was 100%. Hesher et al.[24] have tested the PCA approach, which employs a variety of feature vectors and different sizes of images. In this research, the data set of images consists of 37 subjects, each with six different facial appearances. The recognition accuracy is found to improve when multiple images are used in the gallery. Moreno et al.[25] have split the 3D face model utilizing the Gaussian curvature method, then have built a feature vector based on the segmented portion for face recognition. Their technique scored 78% recognition accuracy on samples of 420 faces from 60 individuals with a variety of facial expressions. Martinez[26] has partitioned the facial model into small portions and developed a probabilistic method to match each portion locally. Further, the matched results are integrated for facial recognition. By generating form signatures for 3D polygonal models, Osada et al.[27] have solved the basic challenge of assessing similarities between 3D objects. The suggested technique depicts the object signature as a sampled shape distribution derived from a new shape function that determines the item's general geometric features. The method is resistant to geometric modifications like rotations and translations and may be used as a preclassifier in 3D object recognition systems.

Formerly, 3D object identification relied on the ICP[7] matching technique, differential geometry method,[28] and spherical correlation approach[29] to calculate matching score from free-form curved surfaces. Prior to 2004, there have been a few freely available 3D face databases. Song et al.[30] have devised a 3D face recognition algorithm that can withstand significant head displacement. The technique leverages geometric information from feature points on the face to correct the head position in a 3D face scan. Samir et al.[31] suggested a method for analyzing facial shapes based on the curvature of the surface. The fundamental concept is to approximate a facial surface using a constrained level curve from the depth image. Using the combination of linear support vector machine (LSVM) and linear discriminant analysis (LDA), Wong et al.[32] presented a 3D face recognition system. By collecting local features from several regions, this approach obtains the sum of invariants. From the frontal face image, 10 subregions and subsequent feature vectors are retrieved. Another approach for retrieving comparable shapes from a vast 3D object collection is priority-driven search.[33] This approach uses local 3D feature sets to represent the objects. The algorithm produces a ranked list of the target objects derived from how closely any subset of k features qualifies for the probe and the predicted object. Many research organizations have recently set up various 3D face databases to test and assess their personal 3D face recognition systems. On diverse 3D face databases, different 3D face recognition algorithms perform differently. Several approaches are employed on a particular 3D face database, and their effectiveness with other databases may vary. Huang et al.[34] have presented a multiscale local binary model (MS-LBP) depth map as a novel 3D surface representation approach. This approach is used with the combination of shape index (SI) map and scale-invariant feature transform (SIFT). Using this approach on the Face Recognition Grand Challenge database (FRGC v2.0[35]), the Rank-1 accuracy is obtained as 96.1%. This approach has been demonstrated to be the potential for handling facial probes that are partially occluded. On the Bosphorus[36] database, Li et al.[37] have presented a mesh-based 3D face recognition method that makes use of a new local shape descriptor and a SIFT-type comparison procedure. Smeets et al.[38] have developed the meshSIFT algorithm and its application on 3D face recognition. The algorithm retrieves features on various scales from 3D surfaces, giving expression-stable 3D face identification. It is been tested against the FRGC and Bosphorus databases.

To obtain 3D geometric information, Soltanpour and Wu[39] have used SIFT keypoint detection on pyramidal shape maps and combined it with 2D keypoints. In this work, the FRGC v2.0 and Bosphorus databases are used for experimentation. On FRGC v2.0, the verification rate is obtained as 99% for all versus all comparisons, and on Bosphorus, it is found to be 95.8% for neutral versus all comparisons. The disadvantage of this SIFT-based method is that it is sensitive to changes in pose. To address the challenges like missing parts, occlusions, and data corruptions, Lei et al.[40] have proposed an efficient 3D face recognition approach where a 3D face scan represents significant facial expressions and variations in the pose with a set of local keypoint-based multiple triangle statistics (KMTS) that is robust to incomplete facial data. A two-phase weighted collaborative representation classification (TPWCRC) framework is taken to accomplish face recognition. Furthermore, performance is evaluated on six databases, namely, Bosphorus, GavabDB, UMB-DB, SHREC 2008, BU-3DFE, and FRGC v2.0 databases.

In 3D facial expressions recognition (FER), Hariri et al.[41] have explored the application of covariance matrices of descriptors, rather than the descriptors themselves. The performance is evaluated on the BU-3DFE and the Bosphorus databases and has been compared with the similar existing methods. Deng et al.[42] have proposed a new 3D face recognition approach based on the local covariance descriptor and Riemannian kernel sparse coding to assess the inherent correlation precisely of extracted features. FRGC v2.0 and Bosphorus databases are being used for experiments and the proposed approach significantly improves the identification accuracy as compared with other current existing methods. Yu et al.[43] have proposed a rigid registration approach based on surface resampling and denoising, which reduces the influence of sampling difference and noise on registration residuals. Bosphorus and FRGC v2.0 databases are used for experiment and the proposed algorithm outperforms the state-of-the-art algorithms. Shi et al.[44] have proposed a 3D face recognition approach integrating LBP and SVM to increase the accuracy and speed of 3D face identification. The feature information of the 3D facial depth image is extracted using the LBP technique, and then the feature information is classified using the SVM algorithm. The experiment shows that the algorithm gives a higher recognition rate and consumes less time by picking samples from the Texas 3DFRD 3D face depth database and the self-made 3D face depth library.

Volumetric CNNs are used in most deep learning-based techniques on 3D data. One type of input mode to 3D CNNs is voxelized forms.[45–47] These forms, however, are hampered by empty data spaces and require computationally costly convolution processes. To capture quality face shapes, it necessitates an excellent level of voxel resolution, which takes a lot of memory. Point cloud features, on the contrary, represent a set of 3D points in such a way that they remain constant to certain internal[48,49] and external[50] modifications. These features might be local or global, and they must be optimally blended to provide the fairest models. Feature-based deep neural networks (DNNs)[51,52] use 3D data in vectors to retrieve unique features from object's shapes and classify them with a deep neural network. Additionally, embedding patch method in CNNs[53] has been proposed as a way to improve face representation. Amores et al.[54] present a feature-dependent solution for 3D nonrigid object's shape extraction in extensive databases by employing a text search method called bag of features. The method can create meaningful and efficient shape descriptors using multiscale diffusion heat kernels, and the results obtained on a large-scale shape retrieval benchmark are state-of-the-art in their respective fields. Compared with CNNs, transformers are a more prevalent and effective solution for a variety of vision problems. Pan et al.[55] use Pointformer as the foundation for cutting-edge object detection models, demonstrating considerable improvements over baseline algorithms on indoor and outdoor datasets. Dosovitskiy et al.[56] investigated the direct use of transformers for image recognition using a conventional transformer encoder in natural language processing and evaluated performance versus cost for several CNN architectures. While these preliminary results are promising, numerous hurdles remain, including other computer vision tasks such as detection and segmentation.

The study of transfer learning is driven by the idea that humans may intelligently utilize previously acquired knowledge to solve new problems faster or more effectively. Neural information processing systems is a postconference workshop held on December 1–2, 1995. The topic of the workshop was "Learning to Learn: Knowledge Consolidation and Transfer in Inductive Systems" and introduced fundamental motivation for transfer learning, focusing on the requirement for lifetime machine learning algorithms that keep and reuse previously acquired knowledge.[57] It was emphasized that small data and personalization should be the emphasis of the future machine learning research. In similar lines, Luttrell et al.[58] combine a pretrained facial recognition model with transfer learning approach creating a network that can accurately predict on a considerably smaller dataset. In template adaption, VGG system is used for transfer learning where features obtained from pretrained VGGNet are clubbed with template-specific linear SVMs, and this approach outperforms against the other similar approaches by a wide margin.[59] Zhao et al.[60] proposed an instance-based transfer learning method called, a weighted ensemble transfer learning framework with multiple feature representations. Kute et al.[61] introduce a unique technique for face recognition and association based on components of faces via transfer learning, demonstrating that the gained knowledge from entire face images is used to classify the components of the face. Cengıl and Çinar[62] developed a multiple classification model of flower images and achieved highest performance with VGG16 model as a pretrained network. Li et al.[63] have proposed a technique for face recognition that does not depend on facial expressions. The technique is based on transfer learning and Siamese networks that can resolve the issue of small-sized sample. Vishnuvardhan and Ravi[64] have presented an effective method for training a facial recognition model, that has been used in banking and other fields. The method employs a transfer learning approach on the cutting-edge facial recognition model, FaceNet, to retrieve deep features of the face and a type of nearest neighbors (NN) algorithm for labeling the face in place of requiring big datasets or powerful GPU computing for training the model. The technique is evaluated on the Georgia techface-database (GTFD). It obtains an accuracy of 96.67%, which is quite near to human vision (97.53%) and represents a substantial advancement over previous techniques.

3 Proposed Technique

In the proposed technique, we convert 3D data to 2.5D data to reduce the resource consumption and time requirements. We use data augmentation to augment 2D and 2.5D face data to increase the size of the dataset to make the training robust. The 2D and 2.5D face images are taken as input to the ResNet-34[9] architecture for feature extraction. In Fig. 2, the block diagram shows the details of the proposed technique. After training the ResNet-34 model on all subject's training samples, we retrieve embeddings from the architecture's second-to-last dense layer. The embeddings retrieved are then utilized for training the Siamese Network,[65] which computes the similarity score between two feature vectors and forecasts if the two objects provided are from the same or distinct object classes. As stated, training of ResNet-34 model is first carried out on 2D data and subsequently, transfer learning is used to train it for 2.5D data where the pre-trained network ResNet-34 on 2D data is reused as the starting point for the training.

3.1 *Prepossessing*

For experimentation, 3D (and corresponding 2D) frontal facial images dataset (ND-Collection D[66]) from the University of Notre Dame (UND) has been used. The proposed technique requires 2D and 2.5D data. Hence the images from 3D database are converted to 2.5D and are then used. Depth images, depth maps, *xyz* maps, surface profiles, and range images are other names for the

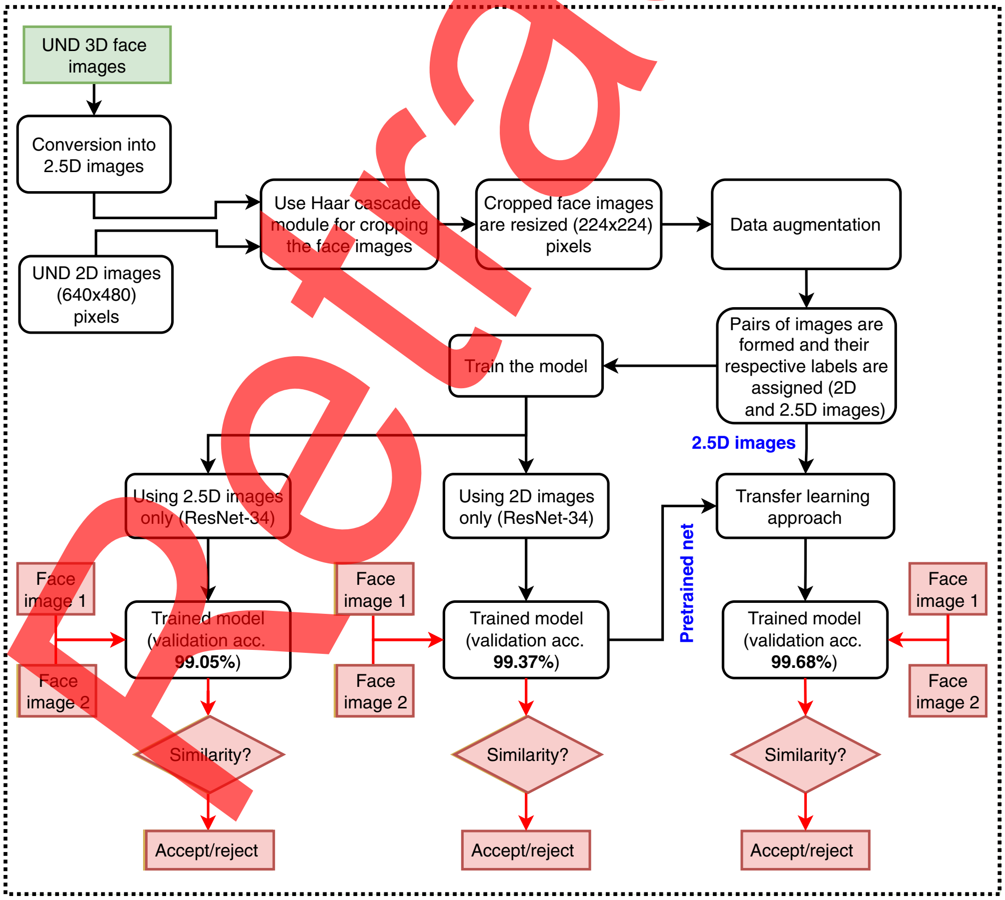
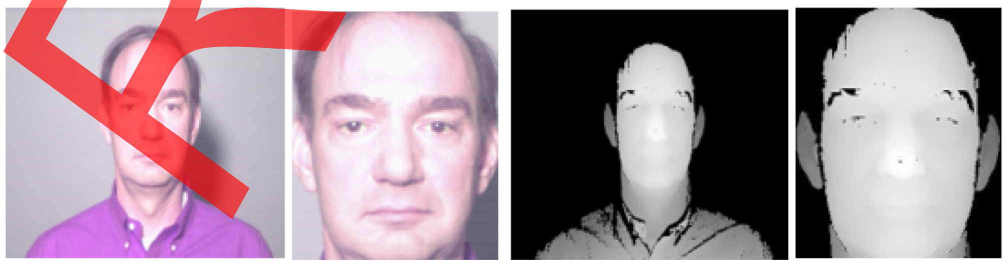

UND 3D face images
Conversion into 2.5D images
UND 2D images (640x480) pixels
Use Haar cascade module for cropping the face images
Cropped face images are resized (224x224) pixels
Data augmentation
Pairs of images are formed and their respective labels are assigned (2D and 2.5D images)
Train the model
2.5D images
Using 2.5D images only (ResNet-34)
Using 2D images only (ResNet-34)
Transfer learning approach
Pretrained net
Face image 1
Face image 2
Trained model (validation acc. 99.05%)
Face image 1
Face image 2
Trained model (validation acc. 99.37%)
Trained model (validation acc. 99.68%)
Face image 1
Face image 2
Similarity?
Similarity?
Similarity?
Accept/reject
Accept/reject
Accept/reject

Fig. 2 Block diagram of the proposed technique.

2.5D images. The 2.5D or range images directly encode surface position for 3D objects. As a result, the shape may be computed very easily. There are two main ways to depict 2.5D images, one is using a list of 3D coordinates in a given reference frame (cloud of points) with no requirement for order whereas the other is a matrix of depth values of points along the x, y image axes, which reveals spatial organization. A 2.5D or range image is a normalized matrix representation where the intensity of each pixel represents the depth of the same location in a 3D image. To generate a 2.5D image from a 3D image, the 3D point cloud is mapped onto a 2D grid, where the 2D grid values depict the depth of the points in the given 3D image. Before feeding these images to the network, it is necessary to preprocess both 2D and 2.5D images under the requirements of the network. In the UND 3D face dataset, most of the images are tainted with spikes because of sensor noise, and thus removal of these spikes is necessary before their use. To denoise the facial images, a fix sized sliding window is traversed across the object. The center of the window is translated to the mean offset if the computed value exceeds a threshold.[67] The 2D images are cropped to include only the faces, which is necessary to remove extraneous information such as shoulders and backgrounds from the images. Figure 3 shows original samples of 2D and 2.5D face images along with their cropped versions respectively from UND dataset. The Haar cascade model is used to crop both 2D and 2.5D face images from original samples. Images are further resized to 224 × 224 pixels to accommodate network requirements.

3.2 *Data Augmentation*

3D data collection often requires more time, so there is very little data available for 3D objects, which poses a challenge in training deep networks. The University of Notre Dame (UND) database (ND-Collection D) contains a small number of samples per subject. It has 277 subjects and the number of samples per subject varies from 3 to 4 in the database. Due to the lack of data, neural network models suffer from overfitting and/or underfitting during training, significantly impacting the model's efficiency. To extract meaningful features, training must be robust, and sufficient samples for training are required. To accomplish this, augmentation is required to overcome the data scarcity. Data augmentation is a technique for generating new data samples from existing ones. This is achieved by transforming samples from the database into new and unique samples using domain-specific knowledge. We augment the image samples by modifying the original image into a new image of the same class by performing multiple transform operations such as scaling, shifting, random rotation, brightness adjustment and Gaussian noise addition. In this work, data augmentation is mainly performed by using two techniques, i.e., rotation and zooming. The rotation data augmentation technique rotates the image by a specified angle. The image can be rotated clockwise or counterclockwise directions around the center of the image between 1 deg to 359 deg. Slight rotations such as 1 deg to 20 deg or −1 deg to −20 deg are preferable as it preserves the label of the data post-transformation. We have chosen 15 deg as a rotation angle and have augmented 2D and 2.5D face images. We have also used the zoom data augmentation method in the experiments where we utilize 0.2 as the zoom value which creates 20% zoom in face images. In the experiments before training and evaluation, 2D and 2.5D images are augmented using the rotation and zoom operations (rotation range as 15 deg and zoom as 0.2) to create new samples and thus augment the data. Table 1 shows original number



(a) A sample 2D image and its cropped version (b) A sample 2.5D image and its cropped version

Fig. 3 A sample image from UND 2D and converted 2.5D face database and its cropped version.

Table 1 Details of 2D and 2.5D images of UND database used in the experimental evaluation of the proposed model.

Dataset	# of subjects	# of image samples	# of image samples after augmentation
2D face images	277	953	5718
2.5D face images	277	953	5718

of samples and the final number of samples obtained after data augmentation where for each image sample in the dataset, we create five other samples by using combination of one or more of the augmentation techniques mentioned above.

3.3 *Proposed Model*

The proposed deep neural network model is an ensemble of ResNet-34 architecture and the Siamese Network as depicted in Fig. 4. We use ResNet-34 network for feature extraction whereas Siamese Network for the recognition based on the extracted features. Further, we propose the use of transfer learning to make the training of the proposed network faster. After preprocessing and augmentation of 2D and 2.5D face images, we prepare experimental set-up for three experiments that are described in Secs. 4.2–4.4, respectively. For example, in Experiment 1 (Sec. 4.2), three sets are prepared, i.e., training set, validation set, and testing set, respectively. The split percentage for each set is 70%, 15%, and 15%, respectively. Since our proposed model consists of ResNet-34 as feature extractor and Siamese network as classification module, the training set (the input set), the validation set, and the testing set need to be modified accordingly. For this, pairs of images are formed for the training, validation, and test sets, and respective labels to the pairs (genuine or imposter) are assigned. Now, the training set is made up of training pairs, which are fed to the network for the training. The training of the network is carried out until a satisfactory validation accuracy is not achieved. We save the model with weights and later use it in Experiment 3 (Sec. 4.4). Typically, the output of the last layer of ResNet-34 is the class of the sample that is given as input. Instead, we are employing ResNet-34 until the second-to-last dense layer of the network. This makes the network to generate feature vectors for the image samples that have been provided as input. We take these features from our train set and use them to train the Siamese network described in Sec. 3.3.2.

3.3.1 *Feature extraction with ResNet-34*

Over the last few years, there have been a series of breakthroughs in the area of computer vision. Especially with the introduction of deep CNN, we are getting state-of-the-art results on problems

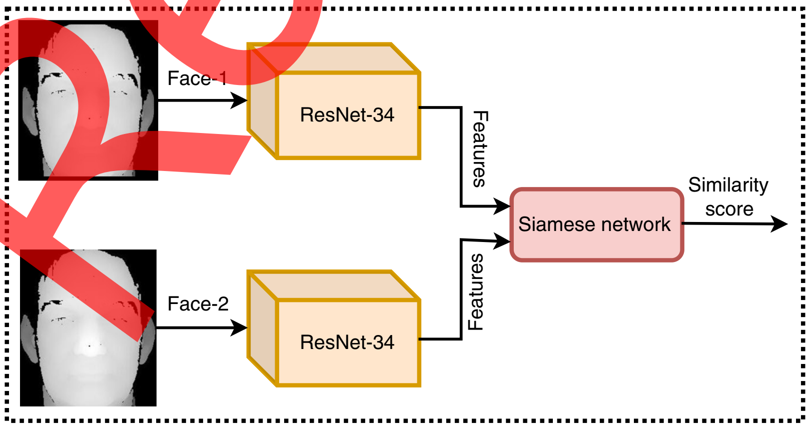


Fig. 4 Network architecture of the proposed model.

like image classification and recognition. This has encouraged researchers to make use of deeper neural networks (adding more layers) to solve complex tasks with improved classification and recognition accuracies. However, it has been seen that as we go on adding more and more layers to the neural networks, it becomes difficult to train them and the accuracy starts saturating and then degrades too. This is not due to overfitting or underfitting but due to the issue of vanishing gradient. If the network is dense, the gradients that calculate the loss function eventually reach zero after several chain rule executions. As a result, the weights never update their values and thus, no learning occurs. This issue is handled in ResNet-34 by using the concept of residual network where it uses the residuals from each layer in the succeeding connected layers. ResNet-34 network model consists of 34 convolutional layers. Its detailed architecture is shown in Fig. 5. The ResNet-34 starts with a convolution layer of 7×7 sized kernel (64) with a stride of two followed by a max-pooling operation. It consists of four residual blocks with size of 3, 4, 6, and 3, respectively. To display all blocks of the network, we have made conv_block-1, conv_block-2, conv_block-3, and conv_block-4 with different colors. The con_block-1 consists of two blocks each having filter size 3×3, and 64 channels (represented as [3×3, conv(), 64] in Fig. 5). Similarly, conv_block-2, conv_block-3, and conv_block-4 are represented as [3×3, conv(), 128], [3×3, conv(), 256], and [3×3, conv(), 512], respectively. In Fig. 5, except for the first block, each block starts with a 3×3 kernel of stride of 2. In Fig. 5, one residual conv_block-1 is being replaced by two conv_block-1, hence total of six conv_block-1 are required. In the same manner for residual conv_block-2, conv_block-3, and conv_block-4 eight, twelve, and six blocks are needed, respectively. The arrows are used for skip connections allowing an alternate shortcut path for the gradient and enabling the gradient to flow backward from later layers to the original filters. These connections also help in allowing the model to learn the identity functions, which ensures that a higher layer will perform at least as good as a lower layer, and not worse.

The ResNet-34 model is pretrained on the ImageNet dataset,[68–70] which has 100,000+ images divided into 200 classes. We make use of the pretrained ResNet-34 model to leverage the power of its robust training on large dataset and capability of handling the vanishing gradient problem.

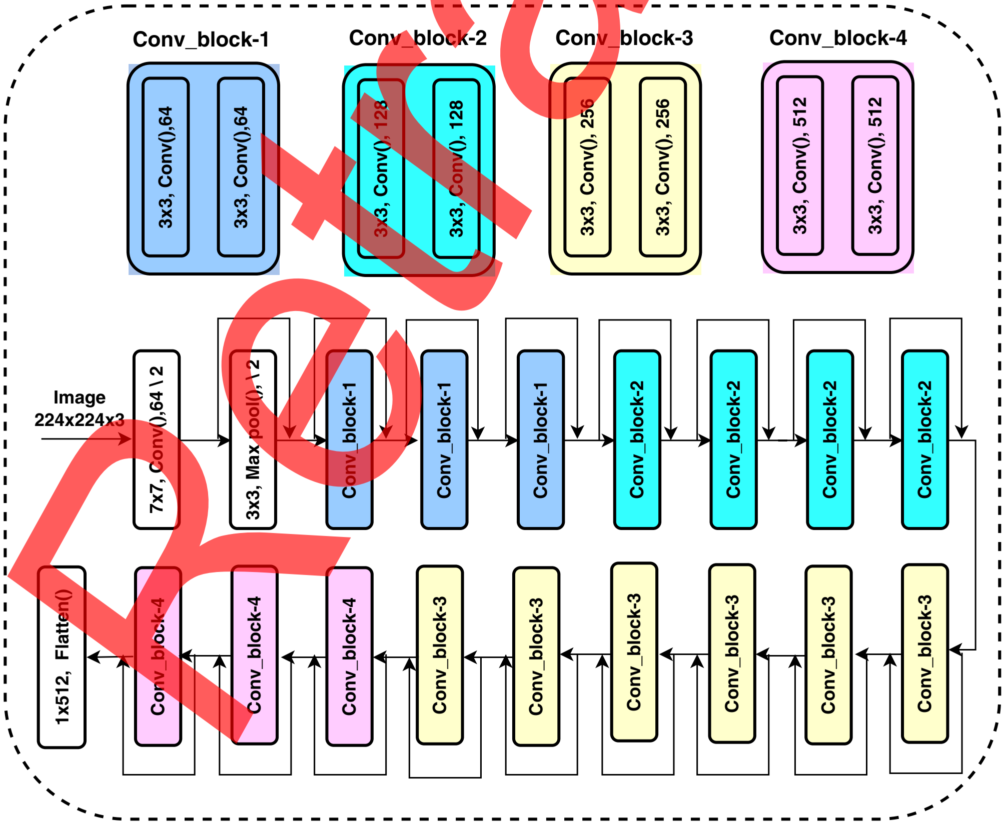


Fig. 5 The architecture of ResNet-34 model.

Before starting the training of ResNet-34 model in the proposed network, we divide the augmented database into the train set, the validation set, and the test set and train ResNet-34 on the train set.

The class of the sample that is provided as input is typically the output of ResNet-34's last layer. Here, the second to last dense layers of ResNet-34 has been used as the output layer to generate feature vectors. These feature vectors are then used to train the Siamese Network.

The termination criterion for the training for each experiment is based on "early stopping" which is based on the comparison of the outcomes of training and the validation processes. In Fig. 7, we can see that as the epochs pass, the error or loss graph in both the training and validation splits decrease. However, at some point, the validation error gets flatten out or grow, although the training error continues to decline. The objective of a validation set is to give us an idea of how our model behaves on data on which it has not been trained. As a result, the epoch at which the validation error begins to rise is precisely when the model overfits to the training set and fails to generalize new data appropriately. This is when we must halt or stop the training.

3.3.2 *Recognition using Siamese network*

A neural network is typically trained to predict multiple classes. When we need to add/remove new classes from the dataset, this causes an issue. In such a situation, we require to upgrade the neural network by retraining it on the whole dataset. Deep neural networks often require a vast amount of data to learn, which wastes time. In contrast, Siamese neural network learns a similarity function, and we can train it to recognize whether or not the two images are identical to one another. The network enables the identification of new types of data without retraining the neural network. A Siamese neural network is a type of neural network architecture with two or more similar subnetworks. The term "Siamese" refers to having the same setup in the two networks, including the same parameters and weights. The updation of parameters is repeated in all subnetworks in it. A Siamese network takes two input feature vectors and determines the similarity between them by matching these feature vectors. It learns a similarity function that compares two inputs expressing how similar they are to each other and generates a similarity score. Further, a threshold value is used on the score to determine whether or not the two feature vectors (or the corresponding test and reference images from which feature vectors have been obtained) are in the same or different classes. If the score is obtained as 0, it shows that there is no similarity, whereas if the score is 1, it shows the complete similarity between the two input feature vectors. Using two sets of pairs of extracted features, the Siamese network must be trained on two separate sample classes: genuine pairs with features from the same class and impostor pairs with features from different classes. It is shown in Fig. 6 that the Siamese network makes use of four different functions to compute the association between the features in pairs. These functions are addition, multiplication, total differences, and square of the total differences between the two features. The concatenation of the outputs of these four operations is then fed to the convolutional layers for training. These feature pairs are used to train the network, determining whether the inputs are genuine or imposters. As it is desired that the training should be as robust as possible, we produce all genuine and impostor pairs for each class by combining each class with every other class. Finally, we examine the model by running the test set on the trained ResNet-34 architecture, creating feature vectors. Subsequently, as previously mentioned, we match every one of these test features against train features from all classes to forecast if the pair is a genuine pair or an imposter. The outcome of the Siamese Network produces the

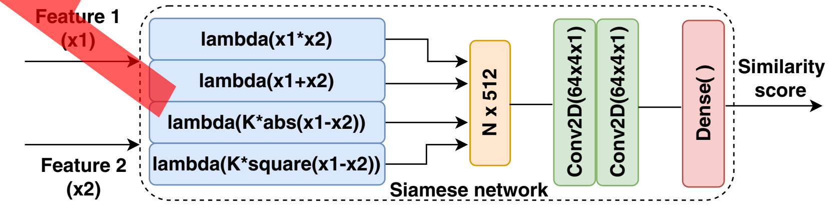


Fig. 6 Detailed architecture of the Siamese Network.

genuineness probability of the pairs and using an appropriate threshold, we determine if a pair is genuine or an impostor.

3.3.3 *Transfer learning*

Transfer learning is a machine learning technique in which a model is created and trained for a specific task, and its weights and architecture are used as the basis for a model working on a different task. It helps deep neural networks to achieve higher accuracy by employing reduced efforts. Usually, as the number of layers in a network is increased, the computation time and the resources required to train the network also increases drastically. Transfer learning helps fine-tune a large pretrained network model on new data in resource-limited settings. As demonstrated in Table 4 for Experiment-3, the time required for training the network reduces due to the fine-tuning of the weights.

We chose ResNet-34 for feature extraction and efficiently make use of transfer learning to train the network on our database. We load the weights of the pretrained ResNet-34 network and then fine-tune them with our database rather than training it from scratch. Further, we initially train the model on only UND 2D data and use this as a starting point and then retrain it on 2.5D data.

4 Experimental Analysis

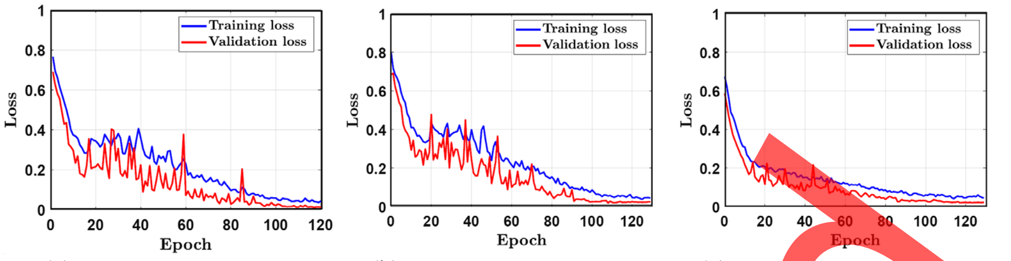
The proposed model relies on the pretrained ResNet-34 for feature extraction and the Siamese network for the computation of similarity scores between the two feature vectors. A pair of images is first preprocessed to the size 224 × 224 pixels and is passed into the network. The ResNet-34 extracts useful features from the images and produces them in the form of feature vectors. These feature vectors are passed through the layers of the Siamese Network to compute the similarity score between the two input images. In this section, three experiments have been performed: recognition using only 2D data, recognition using only 2.5D data, and recognition using the transfer learning approach. We have also combined the outputs of 2D and 2.5D models and performed "OR" and "AND" operations to evaluate the testing accuracy in fusion scenarios. All experiments have been conducted on a machine equipped with an Intel Xeon Gold processor, an NVIDIA GV100GL (Tesla V100 PCIe 32GB) graphics card, and 128GB of RAM.

4.1 *Database Used*

The UND database (ND-collection D)[71,74] contains 277 subjects with 953 aligned 3D face images along with coregistered 2D face images, which have been used for experimentation. This data has been acquired using Minolta Vivid 900 3D range scanner. The face scans in the database contain a considerable amount of noise in the form of spikes. The images are pre-processed as described in Sec. 3.1 before using them in the experimentation. Further, the data goes through the augmentation process as proposed in Sec. 3.2 to increase the size of the database.

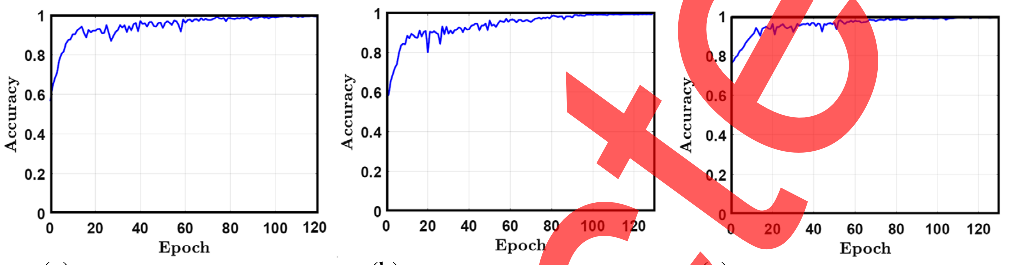
4.2 *Experiment 1 – Recognition Using Only 2D Data*

After augmentation of the 2D images, there are a total of 5718 images in the dataset. We split this dataset into three parts, namely: training, validation, and testing data, which is 70%, 15%, and 15% of the total dataset, respectively. Pairs of images are formed for the training, validation, and test data, and respective labels to the pairs (genuine or imposter) are assigned. The training pairs are sent to the network, and the training of the network is carried out until a satisfactory validation accuracy is achieved. After the training, the model is saved with its weights for future use. In this experiment, we obtain a validation accuracy of 99.05% whereas the testing accuracy for the same is obtained as 98.30%. The graphs of validation and training loss *vs.* epoch are shown in Fig. 7(a) and validation accuracy versus epoch is shown in Fig. 8(a) for this experiment.



(a) Recognition using only 2D data (b) Recognition using only 2.5D data (c) Recognition using transfer learning

Fig. 7 Plots of loss versus epoch for training/validation carried out in three experiments: (a) recognition using only 2D data, (b) recognition using only 2.5D data, and (c) recognition using transfer learning.



(a) Recognition using only 2D data (b) Recognition using only 2.5D data (c) Recognition using transfer learning

Fig. 8 Plots of validation accuracy versus epoch for three experiments: (a) 2D data only, (b) 2.5D data only, and (c) using transfer learning.

4.3 *Experiment 2 – Recognition Using Only 2.5D Data*

The dataset for this experiment contains 5718 images after the augmentation of 2.5D images. We divide the dataset into three parts, namely: training, validation, and testing data, which is 70%, 15%, and 15% of the total dataset, respectively. Pairs of images from these datasets are created and assigned their respective genuine or imposter labels. The training pairs are further passed into the network, and the network is trained until it achieves a satisfactory validation accuracy. Following the training, the model and its associated weights are saved for future use in performing the recognition task. In this experiment, we obtain a validation accuracy of 99.37%. The trained model is further used for testing, where a testing accuracy of 99.10% is obtained. Figures 7(b) and 8(b) show the graphs for validation and training loss versus epoch and the graphs for validation accuracy versus epoch, respectively.

4.4 *Experiment 3 – Recognition Using Transfer Learning Approach*

The dataset in this experiment too comprised of 5718 2.5D images, which were obtained after augmentation of the original 2.5D images. As done in other experiments, we split the dataset into three parts, namely, training, validation, and testing data, respectively, 70%, 15%, and 15% of the total dataset. We first load the model saved in Experiment 1, which is trained on 2D images in Sec. 4.2. Further, pairs of 2.5D images are formed for training, validation, and test datasets, and their respective labels (genuine or imposter) are assigned. The training pairs are then fed to the network which has already been trained on the 2D images, and the training is performed until a satisfactory validation accuracy is achieved. After training, the model and its weights are saved. In this experiment, we obtain a validation accuracy of 99.68%, whereas the network produces a testing accuracy of 99.24%. The graphs of validation and training loss *vs.* epoch for this experiment are shown in Fig. 7(c), whereas the same for validation accuracy *vs.* epoch is shown in Fig. 8(c). The figure shows that the validation accuracy is almost similar to the one obtained in experiment 2 of Sec. 4.3, where only 2.5D data is used. However, the training converges faster than when only 2.5D images are used due to the employment of transfer learning, thus saving time during training. From Fig. 9, we also observe the same where the validation accuracy starts

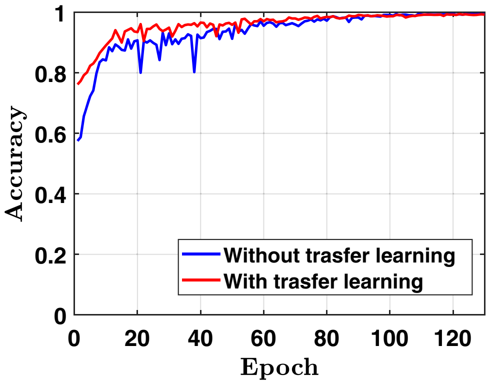


Fig. 9 Comparison of validation accuracy obtained without transfer learning and with transfer learning.

at a higher value due to the use of transfer learning. In the experiment, we used weights of 2D face images trained network and fine-tuned with 2.5D face images. At this stage, we have loaded the trained model on 2D face images that were already saved in Experiment 1 (Sec. 4.2). So instead of starting training from the beginning, only fine-tuning the weights of the 2D trained network for 2.5D face images is required. Here, the use of transfer learning reduces training time, and validation accuracy starts at a higher value shown in Fig. 9. The figure shows a comparison of validation accuracy obtained without transfer learning and with transfer learning. With the transfer learning approach, validation accuracy starts at 78%. In contrast, it starts at 59% without transfer learning, which shows a big difference and proves the advantage of employing transfer learning in the training process

4.5 *Combining the Results of 2D and 2.5D Models*

We observe that network performance has improved after using the transfer learning approach, achieving a testing accuracy of 99.24%. Furthermore, we also attempted a fusion of the outcomes of trained networks on 2D and 2.5D face images, respectively. We take the results from the 2D and the 2.5D models and perform "AND" and "OR" operations to get the final results in the fusion scenarios. This essentially makes a combined model that takes four images as input where two images are considered from the 2D dataset, whereas the other two are from the 2.5D dataset. The two images of the 2D dataset are passed to the 2D model, and the other two images of the 2.5D dataset are passed to the 2.5D model. We take outputs from these models and perform "AND" and "OR" operations to get a final output, as shown in Fig. 10. The experiment with the "AND" operator gives a test accuracy of 99.49%, whereas the experiment with the "OR" operator provides a test accuracy of 97.05%. The comparison between all the experiments is summarized in Table 2.

4.6 *Performance Evaluation in Terms of Some Additional Parameters*

The proposed model is further evaluated for the above-mentioned experiments on the basis of Rank-1 and Rank-2 accuracies, the area under receiver operative characteristics (ROC) curve (AUC), and an equal error rate (EER). Rank-k accuracy is used to analyze the identification performance of a biometric system. It shows the proportion of times the correct sample occurs within the top-k matches. To judge the ranking capabilities of an identification system, the cumulative matching characteristic curve (CMC) is used. AUC is a performance metric that quantifies the degree to which classes may be distinguished at different thresholds and can be calculated with the help of a ROC curve. AUC of a higher value indicates that the model is capable of predicting a class in a better way, whereas the AUC value for a perfect classifier is 1. The EER is the point on the ROC curve that corresponds to an equal probability of incorrectly identifying a positive or negative sample. It is calculated by crossing the ROC curve with the unit square's diagonal.

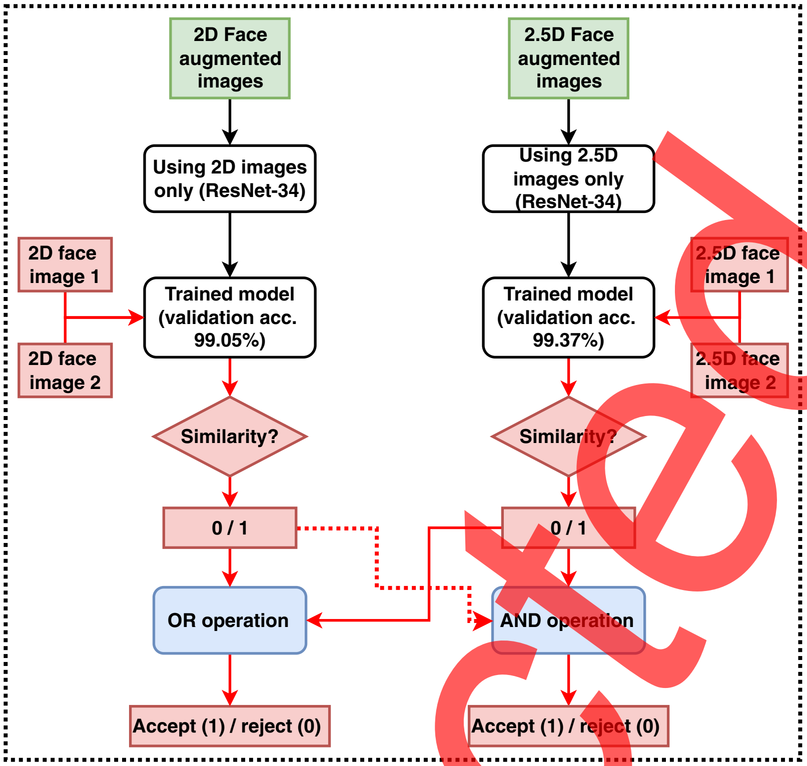


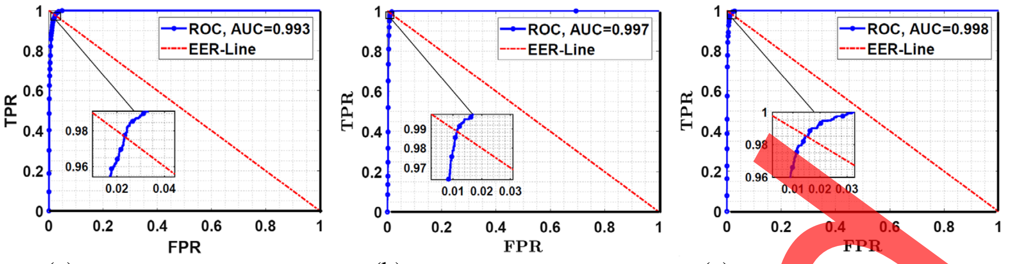
Fig. 10 Block diagram of the procedure used for combining the 2D and 2.5D models.

Table 2 Results of the proposed model in terms of validation and testing accuracies for different experiments.

Experiment name	Validation accuracy (%)	Testing accuracy (%)
Using only 2D data	99.05	98.30
Using only 2.5D data	99.37	99.10
Using transfer learning	99.68	99.24
Combining 2D and 2.5D models (OR operator)	—	97.05
Combining 2D and 2.5D models (AND operator)	—	99.49

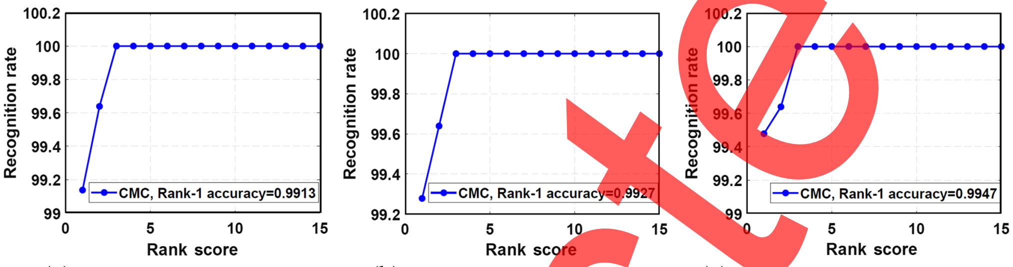
Figure 11 shows ROC curves for the three experiments. Further, the CMC curves where rank-1 accuracy for the three experiments has been shown are presented in Fig. 12. In training, the time taken per epoch is found to be different for all three experiments. Table 3 shows the Rank-1 accuracy, AUC, EER, and the average training time per epoch for all three experiments. From the table, it is evident that the best result is obtained when transfer learning is employed. The training time per epoch of the transfer learning approach is drastically reduced as compared with the training time per epoch in case of 2.5D data. This leads to the reduction in overall computational time and resources as when transfer learning is used, only tuning of weights is required instead of performing the training process from the beginning.

We have used ND-Collection D database of University of Notre Dame (UND) for experimentation. The reason to choose this database for experimentation purposes is that it consists of 3D images along the coregistered 2D images, as is required by our proposed technique. We have compared the performance of the proposed technique with the existing techniques relevant to our work in Table 4. The comparison has been performed in terms of EER and Rank-1 accuracy



(a) Recognition using only 2D data (b) Recognition using only 2.5D data (c) Recognition using transfer learning

Fig. 11 ROC curves for different experiments: (a) recognition using only 2D data, (b) recognition using only 2.5D data, and (c) recognition using transfer learning.



(a) Recognition using only 2D data (b) Recognition using only 2.5D data (c) Recognition using transfer learning

Fig. 12 CMC curves for different experiments: (a) recognition using only 2D data, (b) recognition using only 2.5D data, and (c) recognition using transfer learning.

Table 3 Performance of the proposed model in terms of different evaluation parameters.

Experiment name	Rank-1 accuracy (%)	Rank-2 accuracy (%)	AUC (%)	EER	Average training time per epoch (in seconds)
Using only 2D data	99.13	99.63	99.3	0.0228	110
Using only 2.5D data	99.27	99.63	99.7	0.0104	117
Using transfer learning	99.47	99.24	99.8	0.0138	70

Table 4 Performance comparison of the proposed network with the state-of-the-art techniques on UND Collection-D face database.

Techniques	EER	Identification rate (Rank-1 accuracy) (%)
Chang et al.[71]	—	98.5
Haar et al.[72]	—	98.0
Berretti et al.[73]	—	82.1
Srivastava et al.[67]	0.0080	97.05
Proposed model	0.0138	**99.47**

Note: The bold value represents the result of the proposed technique which is the best identification rate among others identification rates.

values. It is clearly evident from the table that the performance of the proposed technique is superior to that of the existing techniques. This concludes that the exploitation of the face features in the proposed network is capable of delivering results that are superior to those obtained by conventional techniques.

5 Conclusions

A pretrained ResNet-34 architecture for feature extraction and Siamese network for face verification has been used. We observe that ResNet-34 acts as a good feature extractor for both 2D images and 2.5D depth images. Further, we have used data augmentation to overcome the problem of limited face samples. We evaluate the proposed model on the University of Notre Dame (UND) face database (ND-Collection D) by performing three experiments, i.e., recognition using only 2D data, recognition using only 2.5D data, and recognition using a transfer learning approach and achieving Rank-1 accuracy of 99.13%, 99.27%, and 99.47% with EER as 0.0228%, 0.0104%, and 0.0138%, respectively. We also observe that the testing accuracy for 2.5D data is higher compared with 2D data, which proves that 2.5D data (essentially representing 3D data) carries more information than 2D data. We also combine the results of 2D and 2.5D models and achieve Rank-1 accuracy of 99.47%. Our experimental results show that the best performance is achieved when the transfer learning approach is used. The graphical analysis of these results also verifies that the proposed model achieves high accuracy, implying perfect segregation between genuine and imposter pairs. In the transfer learning approach, the average training time in each epoch is reduced in comparison to other approaches. Because of the high efficiency and high accuracy of the proposed model, it can be effectively used for biometric authentication in different applications. The salient contributions of this work lie in proposing the deep neural model for fusion-based face recognition, the use of 2D and 2.5D data for face recognition in the proposed model, and in devising a mechanism for faster training of the proposed network with the help of transfer learning.

Acknowledgments

The authors declare no conflict of interest.

References

1. A. Rosebrock, "What is face recognition?" 2021, https://www.pyimagesearch.com/2021/05/01/what-is-face-recognition (accessed 28 July 2022).
2. H. Zhou et al., "Recent advances on singlemodal and multimodal face recognition: a survey," *IEEE Trans. Hum.-Mach. Syst.* **44**(6), 701–716 (2014).
3. K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D+ 2D face recognition," *Comput. Vision Image Understanding* **101**(1), 1–15 (2006).
4. G. B. Huang et al., "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," in *Proc. Workshop Faces in 'Real-Life' Images: Detect., Align., and Recognit.* (2008).
5. Y. Cai et al., "A fast and robust 3D face recognition approach based on deeply learned face representation," *Neurocomputing* **363**, 375–397 (2019).
6. F. Wang and Z. Zhao, "A survey of iterative closest point algorithm," in *Proc. Chin. Autom. Congr. (CAC)*, pp. 4395–4399 (2017).
7. D. Chetverikov, D. Stepanov, and P. Krsek, "Robust euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm," *Image Vision Comput.* **23**(3), 299–309 (2005).
8. O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. Br. Mach. Vision Conf. (BMVC)*, pp. 41.1–41.12 (2015).
9. K. He et al., "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 770–778 (2016).
10. S. Lawrence et al., "Face recognition: a convolutional neural-network approach," *IEEE Trans. Neural Network* **8**(1), 98–113 (1997).
11. M. Sandler et al., "MobileNetV2: inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, pp. 4510–4520 (2018).
12. A. F. Abate et al., "2D and 3D face recognition: a survey," *Pattern Recognit. Lett.* **28**(14), 1885–1906 (2007).

13. W. Zhao et al., "Face recognition: a literature survey," *ACM Comput. Surv.* **35**(4), 399–458 (2003).
14. C. Xu et al., "Depth vs. intensity: which is more important for face recognition?" in *Proc. the 17th Int. Conf. Pattern Recognit. (ICPR)*, Vol. 1, pp. 342–345 (2004).
15. Y. Wu, G. Pan, and Z. Wu, "Face authentication based on multiple profiles extracted from range data," in *Proc. Audio- and Video-Based Biometric Person Authentication*, pp. 515–522 (2003).
16. G. G. Gordon, "Face recognition from frontal and profile views," in *Proc. Int. Workshop Automatic Face- and Gesture-Recognit., (IWAFGR)*, pp. 47–52 (1995).
17. V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proc. 26th Annu. Conf. Comput. Graphics and Interact. Tech.*, pp. 187–194 (1999).
18. O. Arandjelovic et al., "Face recognition with image sets using manifold density divergence," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit. (CVPR)*, Vol. 1, pp. 581–588 (2005).
19. Y. Hu et al., "Automatic 3D reconstruction for face recognition," in *Sixth IEEE Int. Conf. Autom. Face and Gesture Recognit.*, pp. 843–848 (2004).
20. K. Lee et al., "Video-based face recognition using probabilistic appearance manifolds," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, Vol. 1, pp. I/313–I/320 (2003).
21. G. Pan, Z. Wu, and Y. Pan, "Automatic 3D face verification from range data," in *Proc. 2003 Int. Conf. Multimedia and Expo. ICME'03. Proc. (Cat. No. 03TH8698)*, Vol. 3, pp. III–133 (2003).
22. X. Zhang and Y. Gao, "Face recognition across pose: a review," *Pattern Recognit.* **42**(11), 2876–2896 (2009).
23. C.-S. Chua, F. Han, and Y.-K. Ho, "3D human face recognition using point signature," in *Proc. 4th IEEE Int. Conf. Autom. Face and Gesture Recognit.*, pp. 233–238 (2000).
24. C. Hesher, A. Srivastava, and G. Erlebacher, "A novel technique for face recognition using range imaging," in *Proc. 2003 Seventh Int. Symp. Signal Process. and Its Appl.*, Vol. 2, pp. 201–204 (2003).
25. A. B. Moreno et al., "Face recognition using 3D surface-extracted descriptors," in *Proc. Irish Mach. Vision and Image Process. Conf.*, Vol. 2 (2003).
26. A. M. Martinez, "Recognition of partially occluded and/or imprecisely localized faces using a probabilistic approach," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, Vol. 1, pp. 712–717 (2000).
27. R. Osada et al., "Matching 3D models with shape distributions," in *Proc. Int. Conf. Shape Model. and Appl.*, pp. 154–166 (2001).
28. G. G. Gordon, "Face recognition based on depth maps and surface curvature," in *Proc. Geometric Methods in Comput. Vision*, pp. 234–247 (1991).
29. H. Tanaka, M. Ikeda, and H. Chiaki, "Curvature-based face surface recognition using spherical correlation. principal directions for curved object recognition," in *Proc. Third IEEE Int. Conf. Autom. Face and Gesture Recognit.*, pp. 372–377 (1998).
30. H. Song, U. Yang, and K. Sohn, "3D face recognition under pose varying environments," in *Proc. Int. Workshop Inf. Security Appl.*, pp. 333–347 (2003).
31. C. Samir, A. Srivastava, and M. Daoudi, "Three-dimensional face recognition using shapes of facial curves," *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(11), 1858–1863 (2006).
32. K.-C. Wong et al., "Optimal linear combination of facial regions for improving identification performance," *IEEE Trans. Syst. Man Cybernet. Part B (Cybernet.)* **37**(5), 1138–1148 (2007).
33. T. Funkhouser and P. Shilane, "Partial matching of 3D shapes with priority-driven search," in *Proc. Fourth Eurograph. Symp. Geometry Process.*, pp. 131–142 (2006).
34. D. Huang et al., "3D face recognition using distinctiveness enhanced facial representations and local feature hybrid matching," in *Proc. 2010 Fourth IEEE Int. Conf. Biometrics: Theory, Appl. and Syst. (BTAS)*, pp. 1–7 (2010).
35. P. Phillips et al., "Overview of the face recognition grand challenge," in *Proc. 2005 IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit. (CVPR'05)*, Vol. 1, pp. 947–954 (2005).

36. A. Savran et al., "Bosphorus database for 3D face analysis," *Lect. Notes Comput. Sci.* **5372**, 47–56 (2008).
37. H. Li et al., "Expression robust 3D face recognition via mesh-based histograms of multiple order surface differential quantities," in *Proc. 2011 18th IEEE Int. Conf. Image Process.*, pp. 3053–3056 (2011).
38. D. Smeets et al., "meshSIFT: local surface features for 3D face recognition under expression variations and partial data," *Comput. Vision Image Understand.* **117**(2), 158–169 (2013).
39. S. Soltanpour and Q. J. Wu, "Multimodal 2D-3D face recognition using local descriptors: pyramidal shape map and structural context," *IET Biometrics* **6**(1), 27–35 (2016).
40. Y. Lei et al., "A two-phase weighted collaborative representation for 3D partial face recognition with single sample," *Pattern Recognit.* **52**, 218–237 (2016).
41. W. Hariri et al., "3D facial expression recognition using kernel methods on riemannian manifold," *Eng. Appl. Artif. Intell.* **64**, 25–32 (2017).
42. X. Deng, F. Da, and H. Shao, "Efficient 3D face recognition using local covariance descriptor and riemannian kernel sparse coding," *Comput. Electr. Eng.* **62**, 81–91 (2017).
43. Y. Yu, F. Da, and Y. Guo, "Sparse ICP with resampling and denoising for 3D face verification," *IEEE Trans. Inf. Forensics Secur.* **14**(7), 1917–1927 (2019).
44. L. Shi, X. Wang, and Y. Shen, "Research on 3D face recognition method based on LBP and SVM," *Optik* **220**, 165157 (2020).
45. D. Maturana and S. A. Scherer, "Voxnet: a 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Rob. and Syst. (IROS)*, pp. 922–928 (2015).
46. C. R. Qi et al., "Volumetric and multi-view CNNs for object classification on 3D data," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 5648–5656 (2016).
47. Z. Wu et al., "3D shapenets: a deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 1912–1920 (2015).
48. M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: a quantum mechanical approach to shape analysis," in *Proc. IEEE Int. Conf. Comput. Vision Workshops (ICCV Workshops)*, pp. 1626–1633 (2011).
49. M. M. Bronstein and I. Kokkinos, "Scale-invariant heat kernel signatures for non-rigid shape recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, pp. 1704–1711 (2010).
50. R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Rob. and Autom.*, pp. 3212–3217 (2009).
51. K. Guo, D. Zou, and X. Chen, "3D mesh labeling via deep convolutional neural networks," *ACM Trans. Graphics* **35**(1), 3:1–12:12 (2015).
52. A. Ioannidou et al., "Deep learning advances in computer vision with 3D data: a survey," *ACM Comput. Surv.* **50**(2), 20:1–38:38 (2017).
53. Y. Zhang et al., "Patch strategy for deep face recognition," *IET Image Process.* **12**(5), 819–825 (2018).
54. A. M. Bronstein et al., "Shape google: geometric words and expressions for invariant shape retrieval," *ACM Trans. Graphics* **30**(1), 1–20 (2011).
55. X. Pan et al., "3D object detection with pointformer," in *Proc. 2021 IEEE/CVF Conf. Comput. Vision and Pattern Recognit. (CVPR)*, pp. 7459–7468 (2021).
56. A. Dosovitskiy et al., "An image is worth 16x16 words: transformers for image recognition at scale," in *Proc. 2021 Int. Conf. Learn. Represent. (ICLR)* (2021).
57. S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.* **22**(10), 1345–1359 (2010).
58. J. Luttrell et al., "Facial recognition via transfer learning: fine-tuning keras-vggface," in *Proc. Int. Conf. Comput. Sci. and Comput. Intell. (CSCI)*, pp. 576–579 (2017).
59. N. Crosswhite et al., "Template adaptation for face verification and identification," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG 2017)*, pp. 1–8 (2017).
60. H. Zhao, Q. Liu, and Y. Yang, "Transfer learning with ensemble of multiple feature representations," in *Proc. IEEE 16th Int. Conf. Software Eng. Res., Manag. and Appl. (SERA)*, pp. 54–61 (2018).

61. R. S. Kute, V. Vyas, and A. Anuse, "Component-based face recognition under transfer learning for forensic applications," *Inf. Sci.* **476**, 176–191 (2019).
62. E. Cengıl and A. Çinar, "Multiple classification of flower images using transfer learning," in *Proc. Int. Artif. Intell. and Data Process. Symp. (IDAP)*, pp. 1–6 (2019).
63. Z. Li et al., "3D expression-invariant face verification based on transfer learning and siamese network for small sample size," *Electronics* **10**(17), 2128 (2021).
64. G. Vishnuvardhan and V. Ravi, "*Face recognition using transfer learning on FaceNet: application to banking operations*," pp. 301–309, Springer (2021).
65. H. Wu et al., "Face recognition based on convolution siamese networks," in *Proc. 10th Int. Congr. Image and Signal Process., BioMed. Eng. and Inf. (CISP-BMEI)*, pp. 1–5 (2017).
66. R. Stefanik, "University of notre dame, computer vision research lab," 2018, https://cvrl.nd.edu/projects/data/#nd-collection-d (accessed 28 July 2022).
67. A. M. Srivastava et al., "A technique to match highly similar 3D objects with an application to biomedical security," *Multimedia Tools Appl.* **81**, 13159–13178 (2021).
68. J. Deng et al., "ImageNet: a large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit. (CVPR)* (2009).
69. O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vision (IJCV)* **115**(3), 211–252 (2015).
70. K. Yang et al., "Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy," in *Proc. Conf. Fairness, Account., and Transp.* (2020).
71. K. I. Chang, K. W. Bowyer, and P. J. Flynn, "Face recognition using 2D and 3D facial data," in *Proc. Workshop Multimodal User Authentication*, pp. 25–32 (2003).
72. F. B. Haar Ter and R. C. Veltkamp, "Expression modeling for expression-invariant face recognition," *Comput. Graphics* **34**, 231–241 (2010).
73. S. Berretti, A. Bimbo, and P. Pala, "Sparse matching of salient facial curves for recognition of 3-D faces with missing parts," *IEEE Trans. Inf. Forensics Secur.* **8**(2), 374–389 (2013).
74. P. Flynn, K. Bowyer, and P. J. Phillips, "Assessment of time dependency in face recognition: an initial study," in *Audio- and Video-Based Biometric Person Authentication*, J. Kittler and M. S. Nixon, Eds., pp. 44–51, Springer, Berlin, Heidelberg (2003).

Akhilesh Mohan Srivastava is a PhD research scholar in the Department of Computer Science and Engineering, Indian Institute of Technology Indore, India. He has completed his Master of Technology degree in computer science and engineering from Dr. A.P.J. Abdul Kalam Technical University, India. His areas of research interest are biometrics, pattern recognition, computer vision, image processing, and machine learning. Biometric recognition in 2D and 3D using deep learning for various modalities is one of his study interests.

Sai Dinesh Chintaginjala received his Bachelor of Technology in computer science and engineering from the Indian Institute of Technology Indore, India. His research interests include computer vision, image processing, and deep learning.

Samhit Chowdary Bhogavalli received his Bachelor of Technology in computer science and engineering from the Indian Institute of Technology Indore, India. His research interests include computer vision, deep learning, and machine learning.

Surya Prakash received his MS and PhD degrees in computer science and engineering from the Indian Institute of Technology Madras, India, and the Indian Institute of Technology Kanpur, India, respectively. He is currently an associate professor in the Department of Computer Science and Engineering, Indian Institute of Technology Indore, India. His research interest includes image processing, computer vision, pattern recognition, biometrics, and identity and infrastructure management. He has published several research articles in peer-reviewed international journals and conferences.