Development of Robust Face Recognition System using Transfer Learning and Fine Tuning

Rashmi Jatain*, Dr. Manisha Jailia
Department of Computer Science, Banasthali Vidyapith, Rajasthan
(*Corresponding author's e-mail: rjatain4@gmail.com)

Abstract
Face recognition using deep learning method has achieved exceptional results in the past few years. Face recognition by convolution neural network (CNN) using Deep learning techniques is very complex in nature mainly due to size of the dataset and the prerequisite of high-performance computing power for dataset training and testing. Sometimes, deep learning-based methods fail to recognize a person if dataset size is small. As a consequence, transfer learning and fine-tuning can be used to solve this problem. After outcomes of learning are transferred, implementation of different applications can be done based on the deep learning model's pre-trained weights, which reduces training time with significant improvement in accuracy. Therefore, in this study, we have explored the development of a face recognition system in real time based on deep learning using transfer learning and fine-tuning. For this purpose, we train ResNet18 deep architecture by the Caltech face database. The architecture is fine-tuned by freezing all internal layers except the fully connected layer and modifying the number to class in the SoftMax layer. Precision, recall, f1-score, and accuracy are used to determine the model's performance. Results are analysed by plotting the ROC curve and precision-recall curve.

Keywords: Convolution neural network,ResNet, VGG, Caltech, transfer learning, fine-tuning

1. Introduction

To secure a resource from unauthorized access, person recognition is required. There are many ways to recognize a person, such as monitoring how they walk, signature pattern, typing on the keyboard, face, fingerprint, iris,etc. Walking pattern (GAIT), signature, keystroke style comes under behavioral biometrics while face, fingerprint, etc., are an example of physiological biometrics. It has been proved that physiological biometrics are more secure and cannot be easily spoofed by hackers compared to behavioral biometric traits. Furthermore, among all physiological biometric traits, facial recognition system is the most popular and commonly used biometric system. These systems are easy to install and require low maintenance; hence it is cost-effective. Facial recognition systems don't require physical contact and work with uncooperative users also. However, there are some limitations to these systems. The system fails to recognize a person if a face is covered by cloth or hair. Real-time data are heterogeneous in nature. The images collected through the camera may be low resolution and blur samples. With such types of noisy images, performance of the system degrades. Another factor that affects the accuracy of the facial biometric systems is the quality of camera to be used for surveillance. The low-quality camera comes at a low cost, which reduces the cost of the biometric system but quality of samples collected by these sensors are very low [1][15][16]. It could not store high bit depth images consisting of all color information; the low spatial resolution is often captured by these sensors, which is one of the reasons for performance degradation. Capture environment is also one of the reasons which affect

the accuracy of the system. Surveillance in an indoor environment is sometimes difficult because of the absence of natural light or improper illumination. The performance of face recognition system also depends on the types of features extracted from image samples. Two types of feature extraction methods are Traditional methods and Deep Learning [17]. In tradition methods of feature extraction, each input image is pre-processed using image processing algorithms and handcrafted features are extracted form image samples which are further used for classification/prediction. However, these handcrafted features fail to produce acceptable accuracy if the image samples are of low quality which is very common to be acquired in real-time. Therefore, deep learning takes advantage of automated feature extraction which is proved to be better than tradition means of features extraction methods.To recognize a person in real-time, In this research, we proposed a facial recognition system based on transfer learning and fine tuning. For this purpose, we utilized ResNet18 architecture where fine tuning is done by freezing all layer and modifying SoftMax to the quantity of classes.Overfitting/underfitting issue is avoided by Transfer learning.

Organization of the rest of the section is done as follows. Literature survey is available in Section 2. The architecture of ResNet is discussed in Section 3. Section 4 shows the proposed method of training deep model using transfer learning and parameter tuning. Section 5 presents experiment results. Finally, conclusion of the work is done in section 6.

2. Literature Survey

Recognition of face is widely used in biometric system. Though it is widely used, various challenges are still associated with face recognition especially changing conditions of illumination and poses so it is not a fully solved problem. To overcome the issue of varying illumination, in [1], a convolutional neural network is trained. Model is trained by Yale Face dataset. From the results, a significant improvement in the accuracy (4.96%) is observed. A similar study is done by Le in [2],whereAdaBoost and Artificial Neural Network (ABANN) are combined to create a novel hybrid model.A 2D local texture model based on Multi-Layer Perceptron is used to align the labelled face.Detection and alignment of face is done by evaluating proposed methods using MIT and CMU database. Finally, the experimental results of all phases on the CalTech database indicate the viability of the proposed model.Deep learning methods has the features of high recognition rate and strong robustness as compared to traditional machine learning methods.Therefore, to speed up the recognition process, in [3], a deep learning based fast facial recognition system is developed.Testing is done on image samples of 17 people from the CMU-PIE face database where each person has 170 facial image samples. In the result, it is reported that the rate of recognition is 99.25%, which is greater than the eye of human.Several applications like surveillance, shopping stores, traffic etc are using Face recognition concept.Due to the computation complexity of the processing, recognition of multiple face in real-time is very challenging. Also, it takes much time to process captured image and recognize subjects which affects the recognition speed of system. Keeping in mind the real-time processing, a multiple face recognition framework is proposed by Saypadith et al., [4].Here,deep CNN face recognition algorithm is used forface tracking and detection. Results of the experiment showed thatmultiple faces limit up to 8 faces simultaneouslycan be recognised by the proposed system. This recognition is done in real time with maximum0.23 secondsprocessing time and with the minimum rate ofrecognition 83.67%. A similar study is done by Vizilteret al. in [7]. Convolutional Network with Hashing Forest (CNHF) is used to obtain the family of real-time face representations. Output hashing transform is learned by Boosted Hashing Forest (BHF) technique.

CASIA-WebFace dataset is used to train and theproposed method is evaluated by LFW dataset.The accuracy ofsingle CNN is found to be 97% on LFW.Development of smart and auto attendance management system is one of the useful applications of real-time face recognition system. Improvement of smart attendance system is done by face recognition which is proved to be one among the most useful application in biometrics. Therefore, to make attendance system automatic, reliable and robust, in [8], attendance management system based on real-time face recognition using deep learning is proposed. The proposed method uses Eigenface values, Principal Component Analysis (PCA) and Convolutional Neural Network (CNN) for this purpose. Transfer learning is a technique to avoid overfitting/underfitting of deep model. It is the process of transferring knowledge learned in the past while training deep model by standard dataset. This method id usually adopted when data is less, i.e., it generated good accuracy even with small dataset. To prove this, in [9], Wei Ng et al. proposed an emotion recognition system based on deep learning using transfer learning technique.In the beginning a network pre-training is done on the generic ImageNet dataset, then the authors performsupervised fine-tuningon the network following a two-stageprocess,first stagewhere facial expressions relevant datasets are taken which is followed by next stage performing on the contest dataset. This is a concept of cascading fine-tuning whose results of the experiments shows that this approach achieves better results, compared to fine-tuning following single stage process with the combined datasets. A similar study based on transfer learning is done in [10]. Here, robust Head Pose Estimation system is proposed by authors using CNN augmented by algorithmof transfer learning enabling system to adjust with the changing environment.

3.Material and Methods: Architecture of ResNet

Among all deep learning architectures, VGG16 [5] and its variants got more attention from researchers. Because of its depth, it can extract the most dominants feature and produces excellent result. Due to the extensive number of layers in its network architecture, VGG gets popular, but it also increased the training time and cost. To overcome this issue, residual network architecture, also known as ResNet 18 has been proposed. It has 18 layers but having a skip connection. Before going to discuss about skip connection, here, first, we will see the design of each network. Here, 3x3 filters are used by both VGG and ResNetarchitecture and down sampling is done with Convolutional neural network layers having stride 2. Both architectures have “a global average pooling layer and a 1000-way fully-connected layer with Softmax in the end”. Figure 1 presents the architecture of the VGGplain and VGG with residual blocks.

- Plain VGG: The philosophy of VGG nets [5] (Figure1, left) inspires plain baselines (Figure 1(a), middle). Mostly 3x3 filters are used by convolutional layers and two simple design rules are followed: (i) For the same output feature map size, the layers have the same number of filtersand (ii) If the function map size is halved, the number of filters is doubled, it’s donefor theperseverance of the time complexity per layer.

- Residual Network: On the basis of the plain network explained above, a shortcut connection is made (Figure 1(a), right) due to which the network is turned into its counterpart residual version. When the input and output are of the same dimensions, the identity shortcuts can be

directly used (shortcuts by solid linein Figure 1(a)).Again, there are many variants of ResNet like ResNet18, ResNet 34, ResNet50, ResNet 101 and ResNet152. The details of the architectures are shown in Figure 2. However, in our study, we used ResNet18.

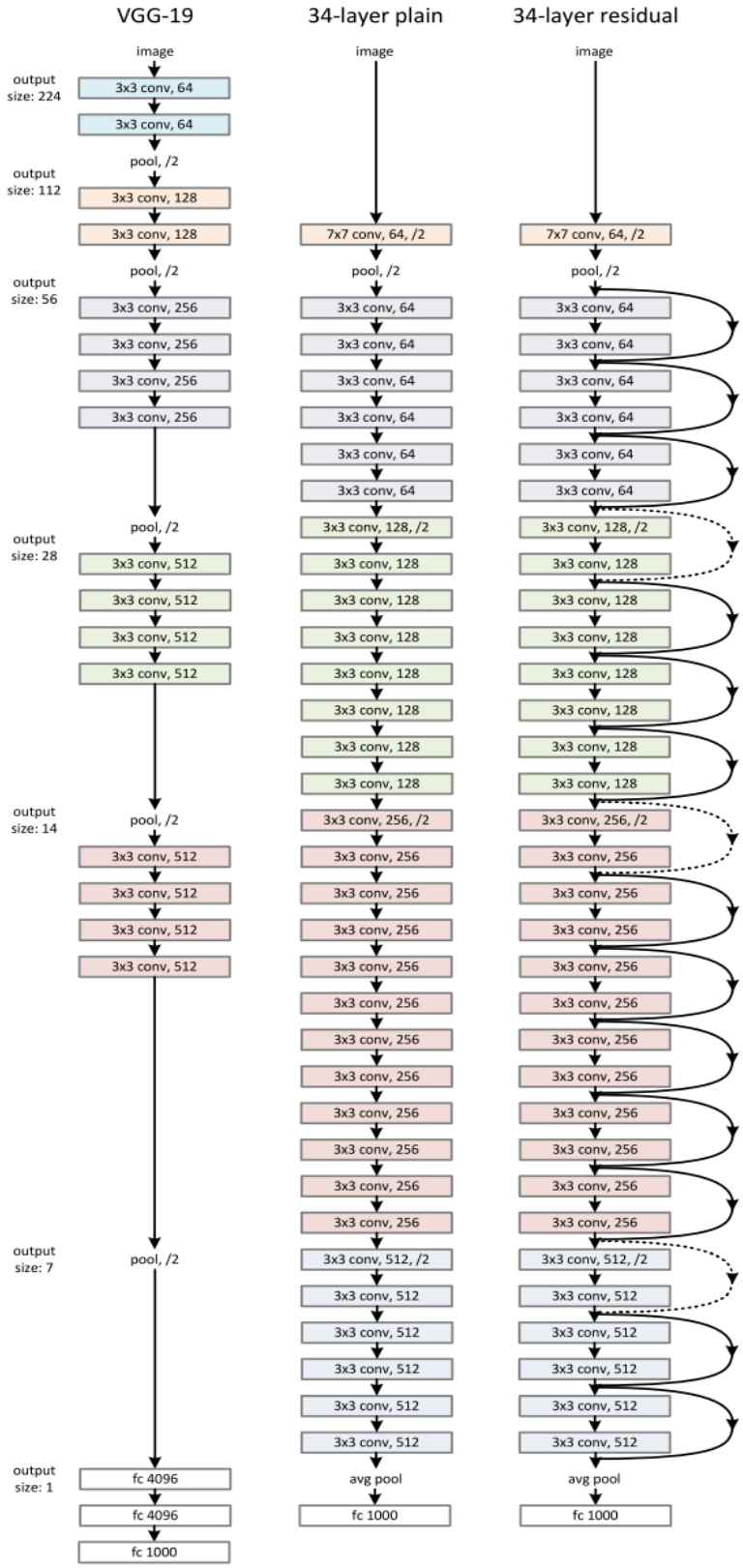

Figure 1(a): Plain VGG and VGG with Residual Blocks [6]

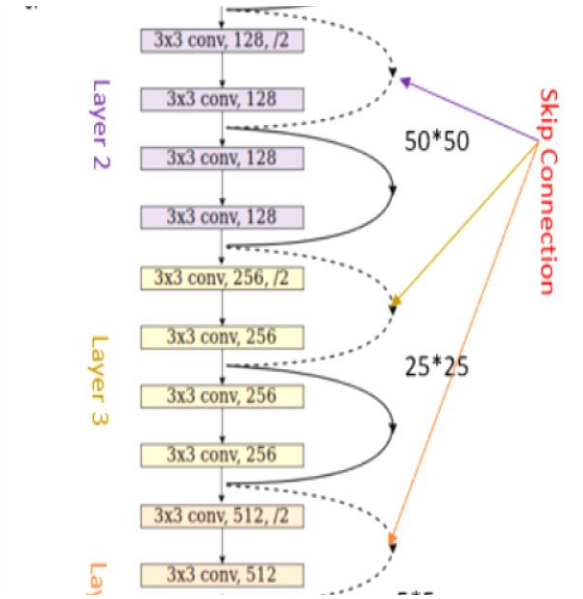

Figure 1(b): The architecture of ResNet18.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 2: Variants of ResNets [6]

A clear representation of architecture of ResNet18 is shown in Figure 1(b). The ResNet18 accept input image of fixed size, i.e., 224x224 pixel resolution and after the execution of each block, reduces it dimensions by half. Finally, it downgrades the image-to-image resolution 7x7, which are passed to 4096 neurons after flattening operation. After each block operation, input image size downgrades from 224x224, 112x112, 56x56, 14x14 to 7x7, and number of filters used increases from 64, 128, 256 up-to 512. In other words, as depth of the network increases, input image size decreases, and number of filters increases. The original ResNet18 architecture is trained on ImageNet dataset, which consists of 1000 objects. Therefore, it has 1000 neurons in fully connected layers. Finally, SoftMax classifies input image to one of the 1000 categories.

4. Assessment Process and Simulation Techniques

The main idea of ResNet is announcing a so-called “identity shortcut connection” that is used to skip one or more than one layers. It is claimed that the network performanceshouldn’t be degraded

by stacking of layersbecause identity mappings could be simply stacked (means the layer that don't have anything to do can be stacked) on the current networkand the resulting architecture's output will be identical. This demonstrates that the "deeper model" would not produce higher levels of "training error" than its shallower counterparts. They assume that fitting a residual mapping to the stacked layers is simpler than fitting the desired underlay mapping directly and this is allowed precisely by theresidual block above explicitly.The image shown in Figure 3 illustrates the basic Residual Block of ResNet.

Two kinds of residual connections are there:

- when the input and output are of the same dimensions, the identity shortcuts (x) can be useddirectly

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}.$$

- When the dimensions are changed:

  1) Identity mapping is still performed by the shortcut, with the increased dimension padding the extra zero entries.
  2) The dimension (done by 1*1 conv) ismatched by the projection shortcut by the following formula

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}.$$

No extra parameters are added in first case, the second one adds in the form of W_{s}
VGG's full 3×3 convolutional layer design is followed by ResNet. Two 3*3 convolutional layers with the same number of output channels make up the residual block. A batch normalisation layer and a ReLU activation function follow each convolutional layer. Then, these 2 convolution operations are skipped and added to the input directly before the final ReLU activation function. The result of the two convolutional layers must be of the similar shape as the input in order to be added together in this kind of design.

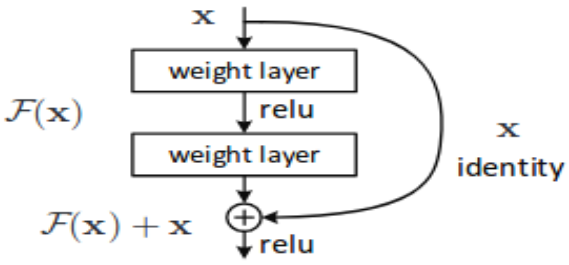

Figure 3: Residual Learning: A building block [6]

4.2 Training technique

There are two ways to train a deep model: Transfer learning with fine tuning and training from scratch [14].

1. *Transfer learning with fine tuning*: In this approach of training, we transfer knowledge of a model that has already trained with another dataset. In other words, in this technique, we use a pre-trained model and fine tune network architecture by modifying layer that is last (softmax) to the no. of classes we want to classify. Except for the last layer, we usually freeze all layers and don't train them.

   This technique is mostly used when we have a smaller number of images in the dataset. This approach reduces the training time and cost as well as also proved to improve the accuracy of the system.

2. *Training from scratch*: In this approach of training, a model is trained from scratch. An input image is passed through each layer and backpropagate feedback/error/loss. After each epoch, the model learns some features and repeats them again till loss is converges to a minimum.

   This approach of training is suitable for large datasets. The greater number of images, the more accuracy of the model we can expect. The drawback with this approach is that it requires a comparatively large amount of time and cost to train the model from scratch.
   In our study, keeping in mind the number of images in the database, we preferred 1st approach. In other words, we used Transfer learning with fine tuning approach to train ResNet18 deep model architecture.

4.3 Network Architecture Fine Tuning

The basic architecture of ResNet 18 is discussed in section 3. However, in our experiment, we are training model using transfer learning by freezing all layers except the last layer and modifying the last layer to the number of classes we have [12][13]. Therefore, the architecture of network used to train with database under study is shown in Figure 4.

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	26	

Figure 4: ResNet18 architecture with fine tuning

If we analyze the architecture of basic ResNet18 and fine-tuned ResNet18, we can notice that, in the basic model, there are 1000 objects to classify, while in our case, we need to classify images into twenty-six classes. Rest all the internal layers are frozen.

4.4 Database used

Markus Weber at the California Institute of Technology gathered the "Caltech" face database, which is a frontal face dataset. It consists of a total of 450 face images from 26 unique people. The images in the database are stored in the spatial resolution of 896 x 592 pixels and in .jpeg format. The database presents diversity among image samples in terms of expressions, illuminations, and background colors. Few samples of imagesof the dataset are (Figure 5).

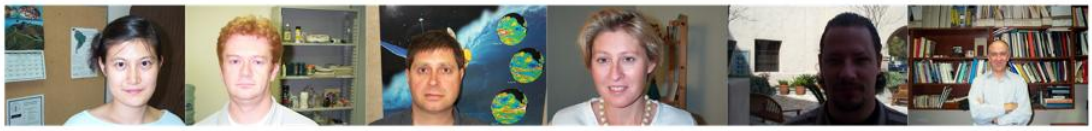

Figure 5: Sample imagesof the database with different level of illumination, expression and background.

4.5 Model training

Our contribution:Few experiments have been conducted for image recognition based on transfer learning. However, there are very limited relevant study related to biometric application. In those studies, a limited type of transformation is used and variation in the degree of transformation is less. However, in our experiment, we extend types of transformation by adding width and height shift range, shear in both (horizontal and vertical direction), flipping the images in horizontal and vertical directions and filling the pixels which are modified due to these transformations using interpolation. These types of transformations generate images of high diversities. The model is trained with these diverse image helps to understand the robustness of developed face recognition system. Also, the dataset used for experiment itself presents the image samples which are acquired in different scenarios which help to better understand the performance and robustness of developed model [30]. Throughout this study, we used ResNet18 deep learning architecture to train the model by the Caltech Face database. We freeze all layers except the last one. The number of classes/labels is changed to the required number of classes. For this purpose, we fine-tuned the network architecture to the number of classes we want to classify. One of the advantages of transfer learning and fine-tuning is that it reduces the training time and resource utilization cost. The use of transfer learning is proved to be more robust to overfitting/underfitting. The tuned ResNet18 architecture is shown in Figure 4. We trained the model multiple times with the objective to improve the performance. We set a low learning rate, i.e., 0.001, momentum 0.9, and trained with 200 epochs, which was experimentally found suitable to yield a better result with overfitting/underfitting. We also employed data augmentation by performing multiple types of transformations. The details of the experimental study are discussed below.

Data Augmentation:"Data augmentation is a method to artificially create new training data from existing training data"[18-21]. This is achieved by using domain-specific techniques to turn

examples from the training data into new and special training examples.Image data augmentation is the most well-known type of data augmentation, which involves transforming images in the training dataset into transformed versions that belong to the same class as the original image.Shifts, turns, zooms, and other operations from the field of image processing are included in transforms. In this experiment, we are generating data by performing data augmentation using following transformation.

1. *Rotation*: We are rotating images randomly by 30 degrees in clockwise and counter clockwise direction.
2. *Zoom*: We are performing zooming operation by the scale of 20%.
3. *Width shift range*: By a factor of 20%, we are shifting width.
4. *Height shift range*: similar to width shift, we are shifting height by 20%
5. *Shear:* We are performing shear transformation by a factor of 15%
6. *Flip:* Flipping of images in horizontal and vertical direction are performed
7. *Fill mode*: Whenever due to transformation, filling of pixels is required, we used nearest neighbour interpolation.

4.6 Hyper parameters:

Learning rate: "The step size, also known as the "learning rate," is the amount by which the weights are updated during training.The learning rate is a configurable hyperparameter that has a small positive value, usually between 0.0 and1.0, and is used in the training of neural networks.In our experiment, the model is trained with learning rate = 1e-3. i.e., in floating point notation, it is equivalent to 0.001

DecayWhen training neural networks, it's popular to use "weight decay," in which the weights are multiplied by a factor slightly less than 1 after each update.This is similar to gradient descent on a quadratic regularisation term and prevents the weights from becoming too high.In our experiment we set decay=1e-5 and found suitable to converge accuracy/loss.

Momentum: The momentum by which learning rate will fluctuate is set to 0.9

Optimization algorithm:"Gradient Descent is a popular optimization technique in Machine Learning and Deep Learning", and it can be applied to nearly all learning algorithms."A gradient is the slope of a functionwhich measures the degree of change of a variable in response to the changes of another variable". In this study, we used Stochastic Gradient Descent (SGD) as an optimizer.

The parameters θ of the objective $J(\theta)$ updated using the standard gradient descent algorithmas,

$$\theta=\theta-\alpha\nabla\theta E[J(\theta)]$$

where the above equation's expectation (E) is approximated by evaluating the cost and gradient over the entire training set. Stochastic Gradient Descent (SGD) simply removes the assumption from the update and computes the parameter gradient using either a single or a few training examples.

The new update is given by,

$$\theta=\theta-\alpha\nabla\theta J(\theta;x(i),y(i))$$

with a pair $(x(i), y(i))$ from the training set.

The database images are divided into two groups: testing and training. Training set contains 80% images, while testing of model is done with 20% of images. Due to a lesserquantity of images in the database, we performed data augmentation. For each image, facial region is detected and cropped

using Viola–Jones algorithm [11]. Cropped face image is passed to deep model followed by augmentation.We trained model with 200 epoch which found sufficient to converge validation accuracy and loss [27-29].

5.Results and Discussion

We plot learning curve by varying number of epochs with respect to accuracy/loss. The curve is shown in Figure 6. From this figure, we observe that as no. of epochs increases,"training accuracy" and "validation accuracy" also increases while "training loss" and "validation loss" decreases. After 180 epoch we found there is no improvement in the validation accuracy so we stop training process there to avoid overfitting/underfitting.

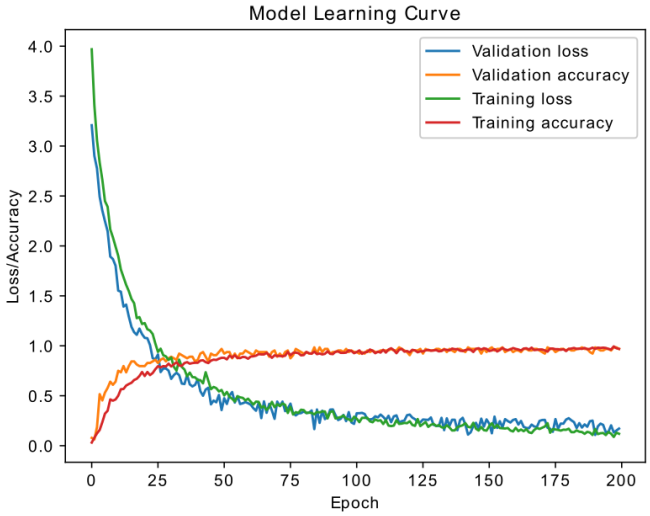

Figure 6: Learning curve showing plot of accuracy/loss with respect to epochs.

With 20% of test set we evaluate the performance of model by calculating confusion matrix."A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which a classification model is confused when it makes predictions. It gives insight not only into the errors being made by a classifier but more importantly the types of errors that are being made" [18].
From the confusion matrix, we calculated precision, recall, f1 score and accuracy of model and calculated precision, recall and f1 score of each class as well as average (macro and weighted) value.

Average	**Precision**	**Recall**	**F1 score**
Macro average	0.90	0.92	0.91
Weighted average	**0.95**	**0.97**	**0.96**

Table 1: Performance results of experimental study

Precision tells how model can precisely predict the level of a given image. In other words, it tries to answer the question "When the model predicts positive, how often is it correct?"Precision refers to how accurate a model is correct out of those predicted positive outcomes and how many of those outcomes are actually positive [22][23].Mathematically, it can be calculated as:

$$Precision = a/ (a + b)$$

Where '*a*'and'*b*'denote True positive and false positive. The higher the value means better the classification. In our experiment, we got weighted average precision 95%.

Recall measures the ratio of true positive and sum of the true positive and false negative. "It actually calculates how many of the Actual Positives our model captures through labelling it as Positive (True Positive)" [22][23]. It can be written as:

$$recall = a / (a + c)$$

where '*c*' represents true negative.

From the table 1, we can notice the calculated recall value is 97%.

"F1 is an overall measure of a model's accuracy that combines precision and recall, in that weird way that addition and multiplication just mix two ingredients to make a separate dish altogether means, a good F1 score means that you have low false positives and low false negatives, so we're correctly identifying real threats and you are not disturbed by false alarms" [22-24]. When F1 score is 1, it is considered perfect, whereas F1 score 0 shows total failure of the model. The mathematical representation of F1 score is:

$$F1 = 2 x (precision x recall) / (precision + recall)$$

Finally, accuracy of the model is calculated by dividing sum to total positive and true negative by all number of examples. The equation of accuracy can be written as:

$$Accuracy = (a + c) / total\ example$$

From the experiment we found trained model is 97% accurate.

ROC curve.i.e., receiver operating characteristic curveis plotted [26]which is "a graphical plot that shows the diagnostic ability of a classifier system as its discrimination threshold is varied". The resultant ROC curve is shown in Figure 7.

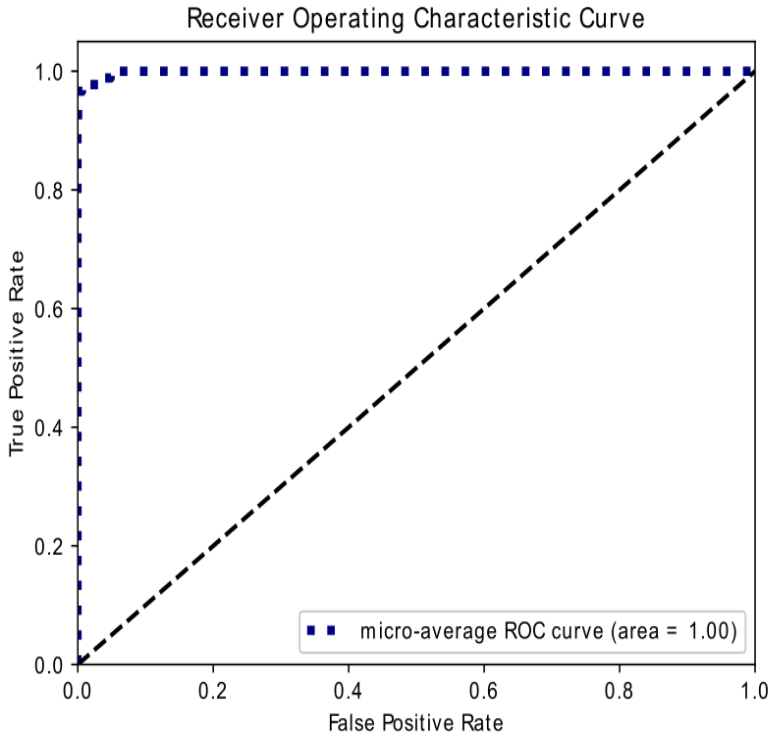

Figure 7: ROC curve

Discussion: From the curve we can see that the area under curve (AUC) is = ~1. The classification is said to be perfect if we have more area under curve. A good classifier has objective to yield AUC value 1. The low value of AUC might be because of smaller amount of data. The area under curve can be improved by adding more images for each class in the database. The ROC curve shows the change in true positive rate with respect to false positive rate at different value of thresholds[27]. The selection of threshold depends on the type of application. Usually, the threshold at which false positive rate becomes equal to true positive rate, is considered for the development of application.

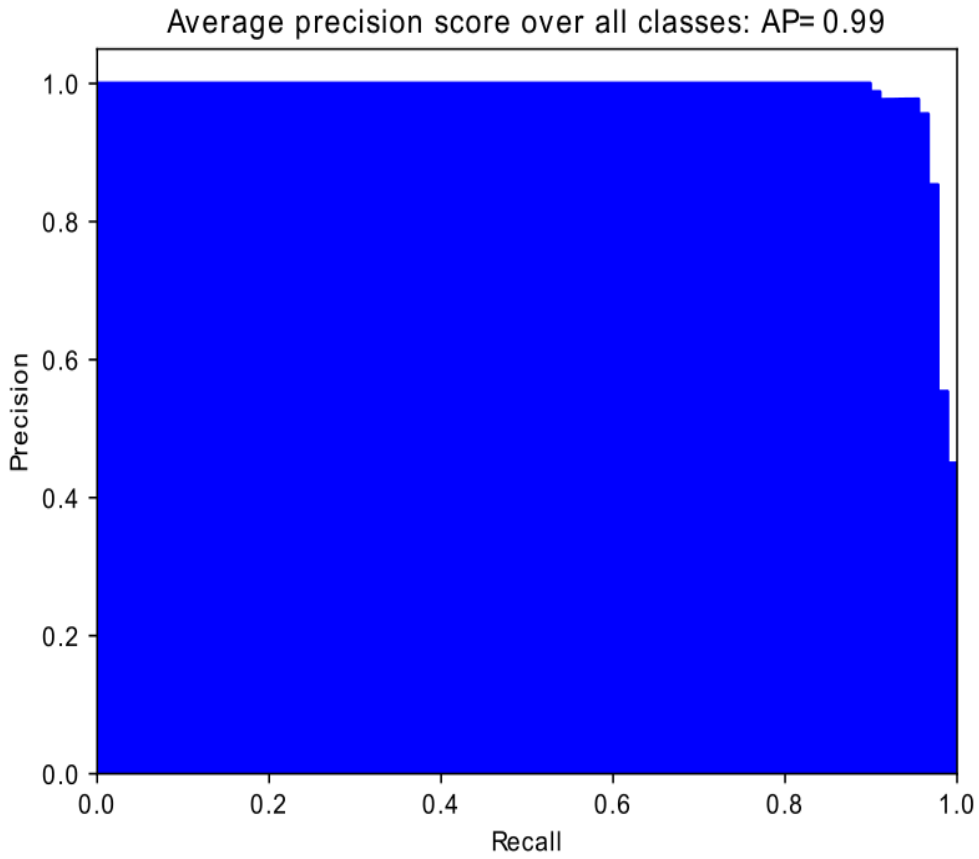

Figure 8: Precision-Recall curve

Discussion: In Figure 8, we plot precision-recall curve. It shows validity of database. If images in the database is unbalanced, i.e., number of images in each class is not equal. In that case, precision-recall curve may produce low score [25]. Since in used database is balanced and image samples are enough to train using transfer learning, because of that we achieved average precision equal to 0.99. Precision-recall curve also shows the suitability of database.

6. Conclusion

Throughout this experiment, we proposed a method to recognize a person using face as biometric traits. For this purpose, we trained deep model such as ResNet18 byCaltech face dataset. Transfer learning with fine tuning is used to train model. Since, we havetwenty-sixsubjects, we modified architecture by freezing all layers except last one. Modification of end layer is done and it is modified to the no. of classes we have, i.e.,twenty-six. Model is trained and tested with database image dividing it in the ratio 80:20. From test set, we evaluated the performance of trained model

by calculating confusion matrix. To analyze the overfitting and underfitting, we plot learning curve. Model performance is also shown and analyzed byprecision recall curve (PR curve)and receiver operating characteristic (ROC) curve. We calculated average area under curve while plotting ROC and average precision score using PR curve. From the test set, we also calculated precision, recall and F1 score of each class as well as average and weighted average values. These results are represented in Table 1. Finally calculation of accuracy is done. From the results we found that model is 97% accurate with precision 95%. Therefore, we can conclude that the proposed method produces acceptable accuracy and hence can be used in real-time application.

References

[1]Ramaiah, N. Pattabhi, Earnest Paul Ijjina, and C. Krishna Mohan. "Illumination invariant face recognition using convolutional neural networks." In *[15 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, IEEE, 2015, pp. 1-4.

[2] Le, Thai Hoang. "Applying artificial neural networks for face recognition." *Advances in Artificial Neural Systems* 2011 (2011).

[3]Qu, Xiujie, Tianbo Wei, Cheng Peng, and Peng Du. "A fast face recognition system based on deep learning." In *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 1, IEEE, 2018,pp. 289-292.

[4] Saypadith, Savath, and SupavadeeAramvith. "Real-time multiple face recognition using deep learning on embedded GPU system." In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2018, pp. 1318-1324.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In dvances in neural information processing systems, 2012, pages 1097-1105.

[6] Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun. Deep Residual Learning for Image Recognition. In Computer vision and Pattern Recognition, 2015.

[7] Vizilter, Yuri, Vladimir Gorbatsevich, Andrey Vorotnikov, and Nikita Kostromov. "Real-time face identification via CNN and boosted hashing forest." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 78-86.

[8] S. Sawhney, K. Kacker, S. Jain, S. N. Singh and R. Garg, "Real-Time Smart Attendance System using Face Recognition Techniques," *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2019, pp. 522-525, doi: 10.1109/CONFLUENCE.2019.8776934.

[9] Ng, Hong-Wei, Viet Dung Nguyen, Vassilios Vonikakis, and Stefan Winkler. "Deep learning for emotion recognition on small datasets using transfer learning." In *Proceedings of the 2015 ACM on international conference on multimodal interaction*, 2015, pp. 443-449.

[10] P. Sreekanth, U. Kulkarni, S. Shetty and M. S. M., "Head Pose Estimation using Transfer Learning," 2018 International Conference on Recent Trends in Advance Computing (ICRTAC), Chennai, India, 2018, pp. 73-79, doi: 10.1109/ICRTAC.2018.8679209.

[11] Wang, Yi-Qing. "An analysis of the Viola-Jones face detection algorithm." *Image Processing on Line* 4 (2014), pp. 128-148.

[12]Masi, Iacopo, Yue Wu, Tal Hassner, and Prem Natarajan. "Deep face recognition: A survey." In *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, IEEE, 2018, pp. 471-478.

[13] Prakash, R. Meena, N. Thenmoezhi, and M. Gayathri. "Face Recognition with Convolutional Neural Network and Transfer Learning." In *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, IEEE, 2019, pp. 861-864.

[14] Weiss, Karl, Taghi M. Khoshgoftaar, and DingDing Wang. "A survey of transfer learning." *Journal of Big data* 3, no. 1 (2016), pp. 1-40.

[15] Bruce, Vicki, and Andy Young. "Understanding face recognition." *British journal of psychology* 77, no. 3 (1986), pp. 305-327.

[16] Phillips, P. Jonathon, Patrick J. Flynn, Todd Scruggs, Kevin W. Bowyer, Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, and William Worek. "Overview of the face recognition grand challenge." In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, IEEE, 2005, pp. 947-954.

[17] Jafri, Rabia, and Hamid R. Arabnia. "A survey of face recognition techniques." *journal of information processing systems* 5, no. 2 (2009), pp.41-68.

[18] Lv, Jiang-Jing, Xiao-Hu Shao, Jia-Shui Huang, Xiang-Dong Zhou, and Xi Zhou. "Data augmentation for face recognition." *Neurocomputing* 230 (2017), pp. 184-196.

[19] Leng, Biao, Kai Yu, and Q. I. N. Jingyan. "Data augmentation for unbalanced face recognition training sets." *Neurocomputing* 235 (2017), pp. 10-14.

[20] Zhicheng Yan. 2016, Hierarchical deep convolutional neural network for image classification,U.S. Patent 20160117587A1

[21] Pei, Zhao, Hang Xu, Yanning Zhang, Min Guo, and Yee-Hong Yang. "Face recognition via deep learning using data augmentation based on orthogonal experiments." *Electronics* 8, no. 10 (2019), pp.1088.

[22] Wang, Xiang, Kai Wang, and Shiguo Lian. "A survey on face data augmentation." *arXiv preprint arXiv:1904.11685* (2019).

[23] Powers, David MW. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." *arXiv preprint arXiv:2010.16061* (2020).

[24] Goutte, Cyril, and Eric Gaussier. "A probabilistic interpretation of precision, recall and F-score, with implication for evaluation." In *European conference on information retrieval*, Springer, Berlin, Heidelberg, 2005, pp. 345-359.

[25] Hripcsak, George, and Adam S. Rothschild. "Agreement, the f-measure, and reliability in information retrieval." *Journal of the American medical informatics association* 12, no. 3 (2005), pp. 296-298.

[26] Ari Teman. 2016, Method and system for authenticating user identity and detecting fraudulent content associated with online activities, U.S. Patent 20160005050A1

[27] Buckland, Michael, and Fredric Gey. "The relationship between recall and precision." *Journal of the American society for information science* 45, no. 1 (1994), pp. 12-19.

[28] McClish, Donna Katzman. "Analyzing a portion of the ROC curve." *Medical Decision Making* 9, no. 3 (1989), pp.190-195.

[29] Bradley, Andrew P. "The use of the area under the ROC curve in the evaluation of machine learning algorithms." *Pattern recognition* 30, no. 7 (1997), pp. 1145-1159.

[30] Wang, J.; Chen, Y.; Hao, S.; Peng, X.; Hu, L. Deep learning for sensor-based activity recognition: A survey. Pattern Recognit. Lett. 2019, 119, 3–11.