Neural Networks (CNNs) and Vgg on Real Time Face Recognition System Showkat A. Dar^a, and S.Palanivel^b

Department of Computer Science and Engineering, Annamalai University, India 608002

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: Face Recognition is considered as a heavily studied topic in computer vision field. The capability to automatically identify and authenticate human's faces using real-time images is an important aspect in surveillance, security, and other related domains. There are separate applications that help in identifying individuals at specific locations which help in detecting intruders. The real-time recognition is mandatory for surveillance purposes. A number of machine learning methods along with classifiers are used for the recognition of faces. This work introduces a new real time face Recognition system. The process is broken into four major steps: (1) database collection, (2) face recognition to identify particular persons and (3) Performance evaluation. For the first step, the system collects 1056 faces in real time for 24 persons using a camera with resolution of 112*92.Second step, efficient real time face recognition algorithm is then used to recognize faces with a known database. For real time face Recognition, VGG-16 with Transfer Learning and Convolutional Neural Network (CNN) are used. This proposed system is implemented using keras. Lastly the performance of these two classifiers is measured using of precision, recall, F1-score, and accuracy.

Keywords: Deep learning, Convolutional Neural Network (CNNs), VGG16, Face authentication, real time face images, and classifiers.

1. Introduction

Face Recognition is a relevant topic in biometrics domain. It is being used in a number of applications nowadays due to its emergence in the recent years. The issues and developments in face recognition have been alluring a lot of scientists working in computer vision, pattern recognition, and biometrics domains. Different face recognition algorithms have been used in diverse applications like indexing and video compression that comes under the domain of biometrics. Face recognition concepts can be used in classifying multimedia content and to help in the quick searching of materials that interests the end user. A comprehensive face recognition mechanism can be of assistance in domains like surveillance and forensic sciences. It can also be used in the areas of law enforcement and to authenticate security and banking systems. In addition to that, it also gives control and preferential access to secured areas and authorised users. The issues in face recognition have garnered even more significance after the spike in terrorism in the recent years. It largely decreases the need for passwords and can offer enhanced security. For this, facial recognition should be used with additional security mechanisms.

In spite of facial recognition's rapid growth and acclaim as a critical authentication mechanism, the algorithms used for facial recognition have not been developed significantly. It has been close to two decades since facial recognition has come to the fore but a comprehensive system that is capable of producing positive results in realtime conditions has not been developed yet. The FRVT test (Face Recognition Vendor Test) carried out by NIST (National Institute of Standards and Technology) has demonstrated that modern face recognition mechanisms will not be able to perform optimally under certain circumstances.

Modern face recognition systems intended for complex environments has attracted a huge attention in recent decades. Face recognition systems that are automated is a developing technology that has garnered a lot of interest. There exist different conventional algorithms, which are used in developing colour and still-face images. The data complexity is increased in colour images as the pixels are mapped to a high-dimensional space. This significantly decreases accuracy and processing efficiency of facial recognition [1]. In the recent years, it has been inferred that deep learning works a lot better for large samples. On the contrary, it has also been observed that conventional machine learning mechanisms may perform optimally in relatively smaller datasets.

The proposed study processes colour images to recognize and detect faces with a good deal of accuracy in a real-time scenario. CNN (Convolutional Neural Network) along with VGG-16 has been used to enhance the accuracy of recognition. The most relevant challenge for such a system is recognition and feature extraction. A system has been proposed here that uses deep learning techniques to extract features. System uses deep learning to recognize faces in an accurate manner. The proposed system will be capable of recognizing more number of faces which can be used for searching suspects as the errors are reduced significantly.

2. Literature Review

This section discusses about the details of face recognition methods with both machine learning and deep learning methods.

Conventional face recognition methods depend to a large extent on features like texture descriptors and edges. These features are combined with certain machine learning methods like PCA -Principal Component Analysis, SVM - Support vector Machines, and LDA - Linear Discriminant Analysis. The complexities associated with these methods made the scientists focus on other methods like illumination-invariant methods [5], [6], pose-invariant methods [4], and age-invariant methods [2], [3].

Apoorva et al [7] implemented a Haar classifier that uses surveillance camera for face recognition. The system had four sequential steps that included (1) real-time image training (2) face recognition with the help of Haarclassifier (3) comparing the real-time images with the images that were captured from camera (4) generation of the result as per the comparison. Haar is used to detect the faces in a robust manner in real-time scenarios. Face detection uses an algorithm called as Haar cascading. The faces can be tracked and recognized in an Open CV Library platform. In this case, the accuracy levels are quite high. A system called as Aadhar has been adopted by India for recognizing the citizens. If this is used as a database for the citizens, a local citizen and a foreigner can be easily recognized. This information can be eventually used to identify if the person is a criminal or not.

Raj [8] proposed a PCA for recognizing faces on real time. The proposed system is efficient as it uses a face recognition system based on C++ and OpenCV. System uses different distance classifiers for feature extraction. The major distance classifiers that are adopted here are Mahalanobis distance, Manhattan Distance, and Euclidian distance. The ideal recognition rate attained using Mahalanobis distance is estimated to be about 92.3%. This is better when compared to the 73.1% that is obtained using normal PCA. The performance is much as they system reacts to face recognition queries in an estimated 0.2 seconds.

Bah and Ming [9] proposed a LBP (Local Binary pattern) for identifying faces. This was used along with other image processing methods like Histogram Equalization, Bilateral Filter, Image Blending and Contrast Adjustment for improving overall accuracy of the system. The LBP codes are improved here due to which the performance of the system is enhanced. As per the experimental results, the method is reliable, robust and accurate. It can be used in a real-life setting as an attendance management system.

Kumar et al [10] proposed an AdaBoost with Haar cascade system for identifying faces in real-time. Majority of the variance is collapsed by this system. To identify human faces, Haar cascade is used along with AdaBoost. LDA and a fast PCA helps in recognizing the faces easily. Matched faces are used for marking attendance in laboratory. Being a biometric system, this is growing acceptance as a real-time attendance system that makes use of fast and efficient algorithms. The accuracy rate of this method is also considerably good.

Shieh et al [11] proposed a PCA with SVM-Particle Swarm Optimization for devising real-time systems intended for facial recognition. Majority of the human-robot interaction applications make use of a PCA based method due to its ability to reduce the dimensionality. Here, PSO implements the feature selection while SVMs function as a PSO's fitness functions for classification issues. The results indicate that proposed system simplifies features comprehensively while attaining a high classification accuracy.

Shubha and Meenakshi [12] proposed a LBP to recognize real time face. The face image is represented by using information about the texture and shape. For representing the face comprehensively, the facial area is divided into different sections. LBP histograms are then extorted which are combined to a single histogram. Facial recognition is then using the Nearest Neighbor classifier. Validation of the algorithm is carried out by devising a prototype model that uses raspberry Pi single-board computer and MATLAB. The results indicate that the LBP algorithm's face recognition rate is relatively higher when compared to other approaches.

Zhang et al [13] put forth a robust and efficient algorithm that could carry out face recognition in complicated backgrounds. It is implemented with the help of a sequence of signal processing modes that included PCA, Haar like feature, LBP, cascade classifier, and Ada Boost. This algorithm makes use of a cascade classifier for training eye and face detectors having precision. Facial features are extracted using the LBP descriptor that can detect faces quickly. Eye detection is carried out using the algorithm which also helps in reducing the rate of false face detection. The PCA algorithm is utilized for recognizing faces accurately. Face recognition algorithm is trained using large databases that have images of faces and non-faces. The accuracy rate of the algorithms is high for facial recognition.

The algorithms used for face recognition that involve machine learning tools function well. Unfortunately, the processing time and training period for these algorithms are considerably large. In the recent times, face recognition techniques have been replaced by deep learning. Deep learning has been observed to perform better for large data sets. On the contrary, traditional machine learning functions at an optimum level with relatively smaller datasets. In conventional machine learning, a problem needs to be broken down into individual steps. Subsequently, the steps have to be solved separately. In terms of facial recognition, one algorithm has to be used for feature extraction while the other has to be used for facial detection. Deep learning can give a solution to this. The web has a large repository of faces called faces in the wild that has helped in the gathering of datasets of random faces [14-18]. These contain real-world variations. Face recognition based on CNN [19] are trained with large datasets and have attained a high level of accuracy. The increased use of deep learning has accelerated the research involved in face recognition. CNNs are now widely used in solving computer vision tasks like the estimation of age, analysis of facial expressions, segmentation, object recognition and detection.

Rekha and Kurian [20] proposed a HOG descriptor for real time facial detection. The method takes an image's HOG and estimates the weights associated with the features of the face. Positive weights are given for features like mouth, nose, and eyes. This will help in comprehensively visualizing a face. The algorithm can be used to to detect faces from different angles that include occluded faces. These methods use a conventional computer vision technique that is based completely on features that are decided by humans. The features' strength is determined once the results are system generated. The algorithm selects the best features that are distinct to the problem in deep learning. Over the years, CNN has made radical changes and progress in the computer vision community. The accuracy of the facial recognition has also increased considerably while using CNN. A major reason attributed to this is the presence of a large-scale dataset.

Almabdy and Elrefaei [21] proposed a method that combined SVM with CNN for facial recognition. The study takes into consideration the CNN architecture that has recorded a good outcome in ILSVRC in the recent years. As per the results, the model was able to attain better accuracy when compared to other modern models. The accuracy was observed to be between 94% and 100%. In addition to that, recognition improved significantly up to 39%.

Passos et al [22] proposed a Multi-Layer Perceptron (MLP) and CNN for facial recognition. It is an open code deep-learning based system to perform facial recognition. Deep learning approaches are utilized to extract fiducial point and embedding. SVM is used for classification task since it is fast for both training and inference. The system achieves an error rate of 0.12103 for facial features detection, which is pretty close to state of art algorithms, and 0.05 for face recognition. Besides, it is capable to run in real-time.

Saypadith and Aramvith [23] proposed a CNN for facial recognition that has been implemented on the embedded GPU system. The method uses facial recognition based on CNN along with face tracking and deep CNN facial recognition algorithm. The results have indicated that the system is capable of recognizing different faces. An estimated 8 faces can be recognized simultaneously within a span of 0.23 seconds. The recognition rate has been above 83.67%.

Schroff et al [24] presented a FaceNet system that directly maps a face from different images as Euclidean space. After the production of the space, face recognition tasks, clustering and verification is implemented using certain techniques. The method makes use of a Convolutional network unlike the bottleneck layer that is used in the previous approaches. The main advantage here is the representational efficiency. Face recognition can be attained using 128 bytes per face.

Sun et al [25] proposed two types of deep neural network architectures that were termed as DeepID3 for recognition of faces. They are built from inception layers and stacked convolution proposed in GoogleNet intended for face recognition. A collection of the two proposed architectures attains 99.53% LFW face verification accuracy. In addition to that it also achieves 96% LFW rank-1 accuracy in face identification. More things about the LFW results has been discussed towards the end.

Sun et al [26] proposed used a DeepID approach to recognize faces. Generalization capability of DeepID could be used for increasing faces count in training. The training data set carried out tests in 10.000 face identities that could reduce the number of neurons. The features are generally extracted from different face regions with over-complete representations..

Wu et al [27] discussed about the facial landmark detection problem. As per the analysis, facial images can be divided into different subsets. According to the findings, a CNN architecture detects the faces in certain appearances and poses. To solve the issue of training data shortage, training examples could be generated. The Tweaked CNN could be used to exceed the landmark detection methods. In the final stage, the trained models and make code could was made available.

3. Proposed Mehtodology

The proposed system is broken into three major steps: (1) database collection, (2) face recognition to identify particular persons, and (3) Performance evaluation. For the first step, the system collects the faces in real time. Database consists of 24 different persons are in the form of 1056 images having the resolution of 112*92. In the second step, Convolutional Neural Network (CNN) and VGG-16 Deep Convolutional Neural Network (DCNN) are introduced for enhancing accuracy of recognition. Finally, these two classifiers results evaluation is done using precision, recall, F1-score, accuracy. In real-time, these classifiers recognize faces with high accuracy. Recognition building blocks are shown in the Figure 1.



FIGURE 1. FACE RECOGNITION BUILDING BLOCKS

Convolutional Neural Network (CNN)

CNN or Convolutional Neural Network has been devised recently by scientific community. It has attracted the attention of image processing applications [28-30] as it has a feature extraction capacity from facial images. The CNN model has been illustrated in Figure 2 that has been categorized into the Convolutional, Fully Connected, and Pooling Layer. Initial two layers are convolutions where every of them is followed by a normalized and a maximum pooling layer. Next in line comes the full connection layer followed by the output layer. Convolution layer carries out an analysis of the neural network.





CNN's basic building blocks are explained as:

Convolution Layer – In CNN, over input real time face matrix, a matrix also termed as kernel is passed over for devising a feature map, which is used in subsequent layer. An operation in mathematics termed as convolution is executed using Kernel sliding over input real time face matrix. On each of the locations, multiplication of face matrix is carried out and adds result set onto final feature map. For example, let use consider a 2-Dimensional kernel filter as K, and a 2-Dimensional image input as I. In this case, the convoluted image is calculated as shown in equation (2):

$$S(i,j) = \sum_{m} \sum_{n} I(m,n) k(i-m,j-n)(2)$$

Non-linear activation functions (ReLU) – Node after convolutional layer is called as activation function. Rectified Linear Unit (ReLU) can be considered as piecewise linear function, which will give output as input if

it is positive, or else output will be given as zero. The expression of ReLu function is R(z)=max(0,z) the function and its derivative image are shown in Figure 3.



FIGURE 3. RELU ACTIVATION FUNCTION

Pooling Layer – Major disadvantage of CNN layer's feature map output is that it records all features in input's exact position. This indicates that in rotation, cropping or any other alterations to input, the facial image will result entirely a different feature map. For rectifying this issue, different convolutional layers down sampling is adopted. Down sampling in this scenario is attained by implementing a pooling layer after nonlinearity layer. There are mainly two types of pooling functions in the CNN which is shown in the Figure 4. In this work max pooling is followed for recognition of faces images.



FIGURE 4. TYPES OF POOLING FUNCITON

Fully Connected Layer (FC) - The last Pooling Layer's final output acts as Fully Connected Layer's input in CNN. One or more layers can be present in this. Fully connected shows, every node in initial layer is connected with every node of next layer as illustrated in the Figure 5.



FIGURE 5. FULLY CONNECTED LAYER

The procedure used for implementing the CNN with their parameters are shown below,

```
###########cnn starts
```

```
model = tf.keras.Sequential([
```

```
tf.keras.layers.Conv2D(5, (3, 3), activation = 'relu', inpu
t_shape = (92, 112, 3)), # the nn will learn the good filter to
use #RGB
```

```
tf.keras.layers.MaxPooling2D(2,2),
```

tf.keras.layers.Conv2D(5, (3, 3), activation = 'relu'),
tf.keras.layers.MaxPooling2D(2,2),
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(24, activation=tf.nn.softmax)

])

CNN compilation

model.compile (optimizer = 'adam', loss = 'sparse_categorical_crossentropy', metrics=['accuracy']) The CNN model that was proposed was trained with 5 epochs, batch_size=32 and using Adam optimizer (adaptive moment estimation). Optimiser's learning rate is set at 0.001.

VGG-16 Deep Convolutional Neural Network (DCNN) with transfer learning

VGG16 is considered as be a CNN model that enhances the AlexNet by making replacements of huge kernel-sized filters (11 and 5 in first two convolutional layers) having multiple 3×3 kernel-sized filters sequentially. The images of the faces are passed through the convolutional layers with the filters having minimum receptive field: 3×3 (this is least size that is required for gathering notion of up/down, left/right and center). It make use of 1×1 convolution filters in one configurations that can be viewed as different input channels linear transformation. Spatial pooling is done using 5 max-pooling layers which follow few of convolutional layers (every convolutional layers are not followed by max-pooling). Same arrangement of max pool layers and convolution layers are carried out consistently throughout entire architecture. Towards end it has 2 Fully Connected layers that are immediately followed by a softmax for output. The 16 in VGG16 indicates that it has 16 layers with associated weights. Transfer learning can improve the learning performance significantly. The core idea behind transfer learning is borrowing the labelled facial images in order to attain greater performance in specific domain of interest. Transfer learning extracts the knowledge from past source tasks to target tasks where the latter has very few facial images are used for facial recognition (See Figure 6) [31].



FIGURE 6.TRANSFER LEARNING



FIGURE 7. ARCHITECTURE OF VGG16

The VGG-16 network's exact structure is depicted in Figure 7:

• The first two convolutional layers are made using 64 feature kernel filters where every filter's overall size is 3×3 . As input real time facial image (RGB image with depth 3) passed into first and second convolutional layer, dimensions changes to 224x224x64. Then the resulting output is passed to max pooling layer with a stride of 2.

• Next two convolutional layers are made using 124 feature kernel filters where filter size is 3×3 . A max pooling layer follows these two layers. Eventually, final output is reduced to 56x56x128.

• Next layers up to the seventh layer are convolutional layers having 3×3 kernel size. Three of these uses 256 feature maps.

• From eight to thirteen, two convolutional layers set having 3×3 kernel size is present. All of these convolutional layers have an overall 512 kernel filters. Max pooling layer follows these layer.

• Final two layers are connected using concealed 4096 unit layers. A softmax output layer follows this.

In the above procedure need to freeze a number of layers (First keeps 7 only active layers and freeze rest of the layers check the output). The sequential model indicates that the entire layers of the model will be sequentially arranged. Check the model summary which is created by using the below code.

model.summary()

The output of this will be the summary of the model which is indicated as given below

```
Model: "sequential 3"
```

Layer (type)	Output	Shape	Param #
=== conv2d_6 (Conv2D)	(None,	90, 110, 5)	140
 max_pooling2d_6 (MaxPooling2	(None,	45, 55, 5)	0
conv2d_7 (Conv2D)	(None,	43, 53, 5)	230
max_pooling2d_7 (MaxPooling2	(None,	21, 26, 5)	0
flatten_3 (Flatten)	(None,	2730)	0
dense_3 (Dense)	(None,	24)	65544
=== Total params: 65,914 Trainable params: 65,914 Non-trainable params: 0			

######maketranfer learning vgg model.trainable = False for layer in vgg model.layers[8:]: import pandas as pd pd.set_option('max_colwidth', -1) layers = [(layer, layer.name, layer.trainable) for layer in vgg model.layers] pd.DataFrame(layers, columns=['Layer Type', 'Layer Name', 'Layer Trainable']) from keras.models import Sequential from keras import optimizers input_shape = vgg_model.output_shape[1] model = Sequential() model.add(InputLayer(input shape=(input shape,))) model.add(Dense(512, activation='relu', input_dim=input_shape)) model.add(Dropout(0.4)) model.add(Dense(512, activation='relu')) model.add(Dropout(0.3)) model.add(Dense(512, activation='relu', input_dim=input_shape)) model.add(Dropout(0.3)) model.add(Dense(512, activation='relu')) model.add(Dropout(0.3)) model.add(Dense(512, activation='relu', input_dim=input_shape))

pred = model.predict(validation_features_vgg)

pred_final = np.argmax(pred, axis=-1)

from skleam.metrics import classification_report, confusion_matrix, accuracy_score print(confusion_matrix(testY,pred_final)) print(classification_report(testY,pred_final)) print(accuracy_score(testY,pred_final))

Keep 4 only active checks the result on the real time facial image dataset. The code used in this work is described as follows,

##########train the model with 4 active layers

model.summary()

history = model.fit(x=train_features_vgg, y=trainY, validation_split=0.1,batch_size=5, epochs=5,

verbose=1)

model.summary()

pred = model.predict(validation_features_vgg)

pred_final = np.argmax(pred, axis=-1)

from skleam metrics import classification_report, confusion_matrix, accuracy_score

print(confusion matrix(testY,pred final))

print(classification_report(testY,pred_final))

print(accuracy_score(testY,pred_final))

```
history = model.fit(x=train_features_vgg, y=trainY, validation_split=0.1, batch_size=5, epochs=5, verbose=1)
###accuracy vgg16 with 7 layers
from skleam.metrics import classification report, confusion matrix, accuracy score
```

```
print(confusion_matrix(testY,pred_final))
print(classification_report(testY,pred_final))
print(accuracy_score(testY,pred_final))
```

These two procedures are used for face recognition.

4.Results and Discussion

This section evaluates the performance of two facial recognitions methods such as CNN and VGG16. Database consists of 1056 images each of size 112*92. A confusion matrix needs to be computed for each class $g_i \in G = \{1, ..., K\}$, in such a way that the ith confusion matrix assumes class g_i as the positive class and the remaining classes g_j with $j \neq i$ as negative class. As each confusion matrix pools together the entire observations

labelled with a separate class apart from g_i as the negative class, this method increases the number of true negatives. This gives us:

"True Positive(TN)" for event values that are correctly analyzed.

"False Positive(FP)" for event values that are incorrectly analyzed.

"True Negative(TN)" for no-event values that are correctly analyzed.

"False Negative(FN)" for no-event values that are incorrectly analyzed.

Let us TP_i,TN_i,FP_i and FN_ito indicate the true positives, true negatives, false negatives and false positives, in the confusion matrix associated with the ith class. Let the recall here be indicated by R and precision by P.

Micro average pools the performance over the least possible unit (the overall facial images):

$$P_{micro} = \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FP_i} (3)$$
$$R_{micro} = \frac{\sum_{i=1}^{|G|} TP_i}{\sum_{i=1}^{|G|} TP_i + FN_i} (4)$$

The micro-averaged precision, P_micro, and recall, R_micro, give rise to the micro F1-score:

 $F1_{micro} = 2.\frac{P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}}(5)$

Given that a classifier gets a large [[F1]]_micro, it denotes that it performs exceedingly well. Here, microaverage may not be sensitive to the overall predictive performance. Due to this, the micro-average can be misleading when there is an imbalance in the class distribution.

Macro average averages over bigger groups and over the performance of individual classes than observations:

$$P_{macro} = \frac{1}{|G|} \sum_{i=1}^{|G|} TP_i / TP_i + FP_i (6)$$

$$R_{macro} = \frac{1}{|G|} \sum_{i=1}^{|G|} TP_i / TP_i + FN_i (7)$$

The recall and macro-averaged precision leads to the macro F1-score:

$$F1_{macro} = 2. \frac{P_{macro} \cdot R_{macro}}{P_{macro} + R_{macro}} (8)$$

If $F1_{macro}$ has a bigger value, it points out to the fact that a classifier is able to perform well for each of the individual class.

Multi-class accuracy is termed as the average of the correct predictions:

$$accuracy = \frac{1}{N} \sum_{k=1}^{|G|} \sum_{x:g(x)=k} I(g(x) = \hat{g}(x))$$
(9)

Where, I is defined as indicator function, which returns 1 when there is a match between the classes and 0 otherwise.



FIGURE 8. MACRO AVERAGE-PRECISION RESULTS COMPARISON VS. CLASSIFIERS

Figure 8 shows the macro average-precision results comparison of four different classifiers such as VGG3, VGG7, VGG16 and CNN. These methods give the macro average-precision results of 78.00%, 97.00%, 99.00% and 72.00%. It can be concluded that VGG16 gives higher macro average-precision when compared to CNN classifier.



FIGURE 9. MACRO AVERAGE-RECALL RESULTS COMPARISON VS. CLASSIFIERS

Figure 9 shows the macro average-recall results comparison of four different classifiers such as VGG3, VGG7, VGG16 and CNN. These methods give the macro average-recall results of 80.00%, 97.00%, 99.00% and 74.00%. It can be concluded that VGG16 gives higher macro average-recall when compared to CNN classifier.



FIGURE 10. MACRO AVERAGE-F1-SCORE RESULTS COMPARISON VS. CLASSIFIERS

Figure 10 show the macro average-F1-score results comparison of four different classifiers such as VGG3, VGG7, VGG16 and CNN. These methods give the macro average-F1-score results of 75.00%, 96.00%, 99.00% and 67.00%. It can be concluded that the proposed VGG16 gives higher macro average-F1-score when compared to CNN classifier.



FIGURE 11. ACCURACY RESULTS COMPARISON VS. CLASSIFIERS

Figure 11 shows the accuracy results comparison of four classifiers such as VGG3, VGG7, VGG16 and CNN. These methods give the accuracy results of 75.71%, 96.53%, 99.37% and 69.09%. It concludes that VGG16 gives higher accuracy when compared to CNN classifier.

5. Conclusion and Future Work

The term Biometrics defines the DNA of an individual along with other aspects like their facial features, geometry of the hands etc. In addition to that, the behavioural aspects like hand signatures, tone of voice and keystrokes are also taken into consideration. In many circumstances, the recognition of face is becoming more accepted and acclaimed in bio-metric based technologies. This helps in measuring an individual's natural data. This work puts forth a real time face recognition using classification methods. The proposed system contains three major steps that include (1) collection of facial images (2) comparison of trained real time face images via two classifiers such as CNN and VGG16 with transfer learning (3) Results comparison with respect to metrics like precision, accuracy, F1-score, recall, and precision. In real-time, CNN and VGG16 classifiers recognize faces with higher accuracy. Both classifiers are performed in sequential manner. For VGG16 model is performed based on the transfer learning. Transfer learning intends to extract information from a number of source tasks and applies it to a target task. So VGG16 gives improved accuracy than the CNN classifier. Classifiers are implemented with

1056 face images of 24 different persons. The proposed system can successfully recognize 24 different person faces which are which could be useful in searching suspects as its accuracy is much higher than other methods. This dataset doesn't include human faces that come under a number of conditions that include camera variation, pose, scale, and illumination which will be left as scope of the future work.

References

- Zhu, X. and Ramanan, D. (2012) Face Detection, Pose Estimation and Landmark Localization in the Wild. IEEE Conference on Computer Vision and Pattern Recognition, Providence, 16-21 June 2012, 2879-2886.
- 2. Park U., Y. Tong, and A. K. Jain, "Age-invariant face recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 5, pp. 947–954, 2010.
- 3. Li Z., U. Park, and A. K. Jain, "A discriminative model for age invariant face recognition," IEEE transactions on information forensics and security, vol. 6, no. 3, pp. 1028–1037, 2011.
- 4. Ding C. and D. Tao, "A comprehensive survey on pose-invariant face recognition," ACM Transactions on intelligent systems and technology (TIST), vol. 7, no. 3, p. 37, 2016.
- 5. Liu D.-H., K.-M. Lam, and L.-S. Shen, "Illumination invariant face recognition," Pattern Recognition, vol. 38, no. 10, pp. 1705–1716, 2005.
- Tan X. and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," IEEE transactions on image processing, vol. 19, no. 6, pp. 1635–1650, 2010.
- Apoorva P., H.C. Impana, S.L. Siri, M.R. Varshitha and B. Ramesh, Automated Criminal Identification by Face Recognition using Open Computer Vision Classifiers, IEEE 3rd International Conference on Computing Methodologies and Communication (ICCMC), Pp.775-778, 2019.
- 8. Raj D., A realtime face recognition system using pca and various distance classifiers, CS676: Computer Vision and Image Processing, Pp.1-11, 2011.
- 9. Bah S.M. and F. Ming, An improved face recognition algorithm and its application in attendance management system, Array, 2020.
- 10. Kumar K.S., V.B. Semwal and R.C. Tripathi, Real time face recognition using adaboost improved fast PCA algorithm. arXiv preprint arXiv:1108.1353, 2011.
- 11. Shieh M.Y., J.S. Chiou, Y.C. Hu and K.Y. Wang, Applications of PCA and SVM-PSO based real-time face recognition system, Mathematical Problems in Engineering, 2014.
- 12. Shubha P. and M. Meenakshi, Human Face Recognition Using Local Binary Pattern Algorithm-Real Time Validation, International Conference On Computational Vision and Bio Inspired Computing, Pp.240-246, 2019.
- 13. Zhang X., T. Gonnot and J. Saniie, Real-time face detection and recognition in complex background, Journal of Signal and Information Processing, Vol.8, No.2, Pp.99-112, 2017.
- 14. Sun Y., X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1891–1898, 2014.
- Parkhi O. M., A. Vedaldi, A. Zisserman, et al., "Deep face recognition.," in BMVC, vol. 1, p. 6, 2015.
- Guo Y., L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for largescale face recognition," in European Conference on Computer Vision, pp. 87–102, Springer, 2016.
- Bansal, A. Nanduri, C. D. Castillo, R. Ranjan, and R. Chellappa, "Umdfaces: An annotated face dataset for training deep networks," 2017 IEEE International Joint Conference on Biometrics (IJCB), ,pp. 464–473, 2017.

- Cao Q., L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2:A dataset for recognising faces across pose and age," arXiv preprint arXiv:1710.08092, 2017.
- 19. Trigueros, D.S., Meng, L. and Hartnett, M., 2018. Face recognition: from traditional to deep learning methods. arXiv preprint arXiv:1811.00116.
- Rekha N. and M.Z. Kurian, Face detection in real time based on HOG, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Vol.3, No.4, Pp.1345-1352, 2014.
- 21. Almabdy S. and L. Elrefaei, Deep Convolutional Neural Network-Based Approaches for Face Recognition, Applied Sciences, Vol.9, No.20, Pp.1-21, 2019.
- Passos W.L., I.M. Quintanilha and G.M. Araujo, Real-Time Deep-Learning-Based System for Facial Recognition, Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT), Pp.895-899, 2018.
- 23. Saypadith S. and S. Aramvith, Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System, Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Pp.1318-1324, 2018.
- 24. Schroff F., D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 815-823, Jun. 2015.
- 25. Sun Y., D. Liang, X. Wang and X. Tang, "DeepID3: Face recognition with very deep neural networks" in arXiv:1502.0087, Feb. 2015.
- 26. Sun Y., X. Wang and X. Tang, "Deep learning face representation from predicting 10000 classes", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1891-1898, Jun. 2014.
- 27. Wu Y., T. Hassner, K. Kim, G. Medioni and P. Natarajan, "Facial landmark detection with tweaked convolutional neural networks", Proc. IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 12, pp. 3067-3074, Dec. 2018.
- Wen, L., Li, X., Gao, L. and Zhang, Y., 2017. A new convolutional neural network-based datadriven fault diagnosis method. IEEE Transactions on Industrial Electronics, 65(7), pp.5990-5998.
- 29. Ding, J., Chen, B., Liu, H. and Huang, M., 2016. Convolutional neural network with data augmentation for SAR target recognition. IEEE Geoscience and remote sensing letters, 13(3), pp.364-368.
- Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., Adam, M., Gertych, A. and San Tan, R., 2017. A deep convolutional neural network model to classify heartbeats. Computers in biology and medicine, 89, pp.389-396.
- Tammina, S., Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images, International Journal of Scientific and Research Publications, Volume 9, Issue 10, pp.143-150.