A Comparative Study on Facial Recognition Algorithms

Sanmoy Paul and Sameer Acharya,

NMIMS University, Mumbai

Abstract: Facial recognition methods were first explored in security systems to identify and compare human faces and is far superior compared to biometric and iris recognition, this technique has been implemented in iris recognition, image detection etc. Recently these methods have been explored in other fields of study and have become a commercial identification and marketing tool. This paper describes the different algorithms of facial recognition and compared their recognition accuracies. The face is detected through Haar Cascades algorithm which is saved into a database, after that, the study intended to compare facial recognition accuracy of the well-known algorithms Eigen faces with PCA, SVM, KNN, and CNN. The results showed out of the three algorithms we used CNN yielded the maximum accuracy.

Keywords: Eigen Values Haar Cascades Facial Recognition Principal Component Analysis, Convolutional Neural Network (CNN) K-Nearest Neighbour (KNN) Support Vector Machine.

Introduction

Every Human face has a unique characteristic that differs from one other , therefore facial recognition becomes a credible source of identification apart from fingerprint scanners (Rodavia, Bernaldez, & Ballita, 2017)[14]. Face recognition is popular and widely used for personal identification. The automatic facial recognition system involves the application of an intelligent artificial system to recognise the human faces under any circumstances. Today the study of facial recognition system has involved a keen interest in pattern recognition, computer vision and other related fields. A Camera can be used for face recognition system. Facial recognition provides an inexpensive and reliable personal identification which has been explored in many domains (Phankokkruad & Jaturawat, 2017)[13]. It is in expensive than a biometric system.

Recognition accuracy is a vital factor in the facial recognition system. However, there are many factors that affect the recognition accuracy which varies from environmental factors, quality of image, shifting and scaling of images etc. Sometimes these factors make an image non-ideal to use in a recognition system because of their low prediction accuracy (Phankokkruad & Jaturawat, 2017)[13]. Other factors that affect the accuracy are facial shape, texture, specs, hair, illumination etc. In many publications on facial recognition, algorithms used have certain characteristics and aspects which determines the best accuracy for that algorithm.

In this research studies, we have used some well know facial recognition algorithms and compared their recognition accuracies both on train and test set. Eigen faces, SVM, KNN and CNN algorithms have been employed in this experimental study. A variation in facial viewpoints has been used in the experiment to study the effect of recognition accuracy. Advantages and disadvantages of different algorithms can be studied based on the result achieved. Consequently, it will help the developers to choose the best facial recognition algorithm in their field of implementation.

Flow-Chart

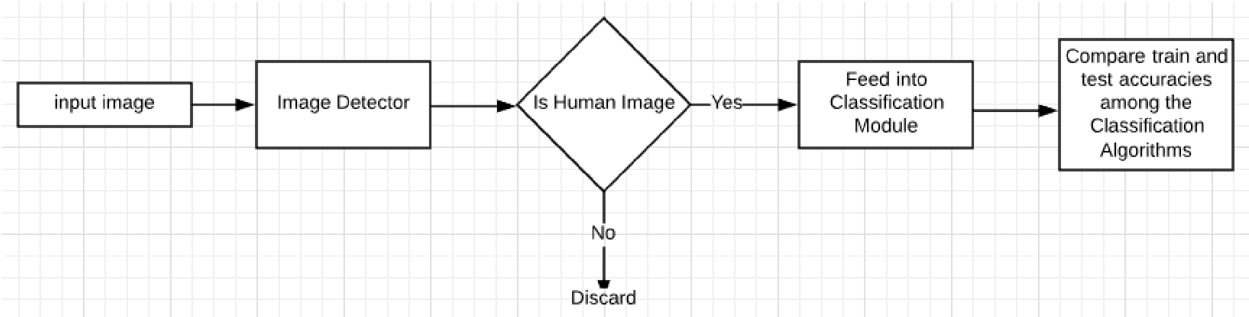


Fig-1: Flow-Chart

Literature Review

1. **Eigen faces algorithm with PCA**

The Eigen faces Algorithm is the most commonly used methods in the field of facial recognition.(Çarıkçı & Özen, 2012)[3] have researched on Eigen faces method with smallest Euclidian distance is the person that resembles the most. The images are converted into to face vectors Γi and then split into a training and test set. The face vectors are then standardised by computing the mean of the face vectors then subtract the mean face from each of the face vectors to get as normalised face vector Φi (Paul, Acharya, & Bhuva, 2018)[18.

$\Phi i= \Gamma i-\psi$,

where $\Psi=1/M \sum_{i=1}^{M} \Gamma I$, ψ is the average face vector.

The Covariance matrix C is given by

$C=1/M \sum_{n=1}^{M}\Phi n= AAT$ (N2xN2 matrix)

where $A=[\Phi 1, \Phi 2, \ldots \Phi M]$, each normalised vectors in each column makes up A, where A is

N^2xM matrix (Paul, Acharya, & Bhuva, 2018)[18. Now choosing k significant vectors from the Eigen vector space becomes a problem. Therefore, we use PCA, where we assume face vector space as a lower dimensional subspace and recomputed the covariance matrix as C=ATA, the covariance matrix becomes of M x M dimension. Similar methods have been further researched down by (Chen & Jenkins, 2017)[5]. Now it becomes easier to find the k significant Eigenvectors from the face vector space. The covariance matrix returns M Eigenvectors each of Mx1 dimension. After PCA, k best Eigen faces are selected that explains maximum variance such that k<M which represents the total training set. The selected k Eigen faces must be of the original dimensionality of the face vector space so we map it back into the original dimensionality by $u_i= A*v_i$, where u_i the Eigen vector of higher dimensionality and v_i is the Eigen vector of lower dimensionality. (Detsing & Ketcham, 2017)[5] also used PCA with Eigen faces in their research. (Paul, Acharya, & Bhuva, 2018)[18] in their review have found that PCA with Eigen faces is most commonly used to extract distinct features with a face from person to person. PCA not only made the computational problem easier but also helped in the reduction of noise which could have impacted the result. Now we each face in the training set is represented as a linear combination of k Eigenvectors along with the average face vector multiplied by the weights w. The weight vector is represented as $\Omega=[w1,w2,\ldots\ldots wk]$, this is the Eigen face representation of the ith face and the weights for each face is calculated. (Chakrabarti & Dutta,

2013)[4] also used PCA with Eigen faces for their research of Facial expressions. The flowchart of our Eigen faces algorithm is mentioned (Refer to figure-2).

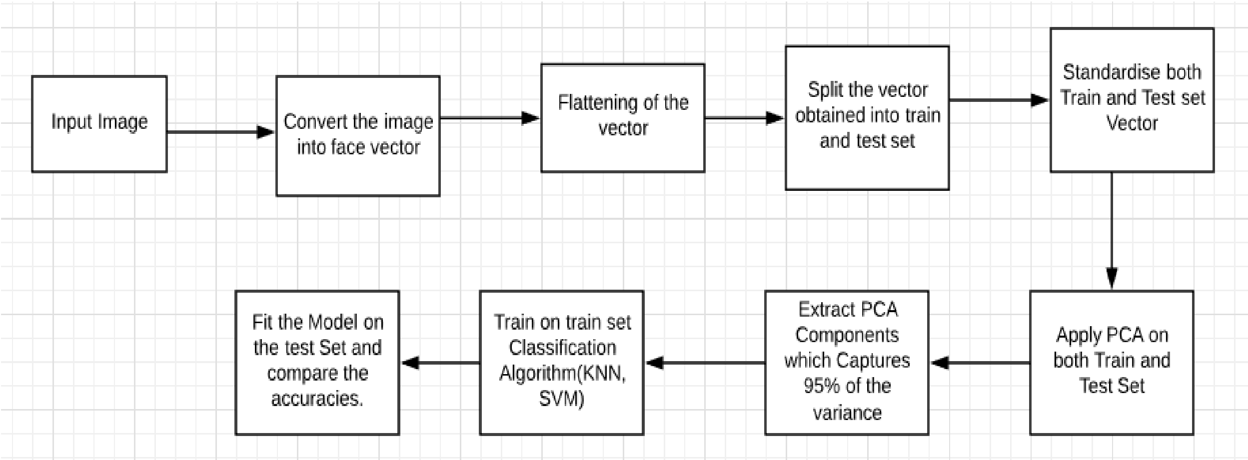


Fig-2: Flowchart for Eigen faces algorithm

2. Haar-Cascade Classifiers Viola Jones

Object detection is a method for identifying the existence of an object of a certain class (Soo, 2014)[16] by a method of image processing. Objects can be classified based on texture, colour and shape. One can a employ colour coded approach to identify a object, colour –coded approach has its downside as lighting conditions play's very important role in detecting the object. Object detection based on features, the shape etc has been employed to overcome the prior approach.

Viola Jones is based on

- Integral Image
- Haar Feature Selection
- Learning Classifier with Ada-Boost
- Cascade structure

Haar-Like Features considers rectangular regions in detection windows, which thereby sums up pixel values and calculates the difference of these sums which is then used to categorize sub sections of an image. In facial features, the area under eyes is darker than the area near cheeks. A rectangular target size is moved over the input images that calculates the haar feature , which is then compared with the learning threshold that differentiates an object from non-object.

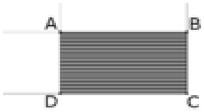
Feature extraction is based on 3 approaches:-

- Holistic
- Feature Based
- Hybrid

The implementation of the open-CV library has led a generation of object detection classifier which uses Haar-Like over image to detect features of a human face like, 2 eyes, Nose, Mouth, Ears.

Viola-Jones have used summed-area tables called as integral images which is a 2 Dimensional lookup table a “form of matrix table” which is equal to original image matrix, it allows summation of rectangular areas in particular position using 4 look ups

Sum= I(C) + I(A) - I(B) - I(D) where A,B,C,D are part of Integral Image I



Viola Jones use Adaptive Boosting to select some features of the face to train the classifier, Ada-boost is a weighted sum of weak classifiers.

Learning Algorithm $h(\mathrm{x}) = \sum_{j=1}^{m} \alpha\mathrm{j} * \mathrm{hj}(\mathrm{x})$

Weak classifier

$$\mathbf{h_j}(\boldsymbol{x}) = \begin{cases} -\boldsymbol{sj}, & fj < \theta j \\ \boldsymbol{sj}, & otherwise \end{cases}$$

3. Support Vector Machine (SVM)

Support vector machine is a supervised learning model with associated learning algorithms (Saitta, 1995)[15], it analyses data for classification and regression analysis. SVM training algorithm differentiates categories making it non probabilistic binary linear classifier. SVM separates categories by a clear gap as wide as possible also known as margin. SVM also performs non-linear classification using Kernel trick. SVM is capable of delivering higher classification accuracy. (Srivastava & Bhambhu, 2010)SVM can be used to detect text, digit, image classification and object detection.

SVM constructs hyper-plane between 2 or more clusters; hyper-plane can be used to detect outliers among data. To achieve optimal parameter setting SVM requires extensive cross validation commonly known as model selection. The choice of a kernel function, the standard deviation of the Gaussian kernel, training data, relative weights of slack variable impacts the overall results. SVM minimizes the empirical classification error and maximizes the geometric margin. SVM is based on Structural Risk Minimization (SRM). SVM maps input vector to a higher dimensional space with maximal separating hyper plane.

$$\{(x1,y1),(x2,y2),(x3,y3),(x4,y4)\ldots\ldots\ldots,(xn, yn)\}.$$

where yn=1 or -1

$$w . x + b = 0 \quad ----(1)$$

where b is a scalar and w is a p-dimensional Vector.

For linear separable data, hyper plane is selected to differentiate datasets

$$w. xi – b \geq 1 \text{ or } w. xi – b \leq -1$$

the distance between the hyper plane is $(2 / \lVert w \rVert)$. Here $\lVert w \rVert$ can be minimised by

$$w. xi - b \geq 1 \text{ or } w. xi - b \leq -1$$

which can be arranged as

$$yi (w. xi - b) \geq 1 , 1 \leq i \leq n \quad \text{------(2)}$$

Hyper plane with the largest margin is defined by M= 2/|w|

$$y_j[w^t.x_j + b] = 1,\ i = 1 \text{ --------(3)}$$

Hyper plane with maximum margin is known as optimal canonical Hyperplane

$$y_j[w^T.x_i + b] \geq 1,\ i = 1,2 \ldots .I \quad \text{--------(4)}$$

$$\sum_{i=1}^{n} \left(\binom{n}{k} l\ \mathrm{LP} = \mathrm{L(w,b,\alpha)} = \frac{1}{2}\|\mathrm{w}\|2 - \sum \quad \alpha\mathrm{i}\ (\mathrm{yi}\ (\mathrm{w^t\ xi} + \mathrm{b}\ ) - 1) = \frac{1}{2}\mathrm{wT\ w} - \sum \quad \alpha\mathrm{i}\ \left(\mathrm{yi}\left(\mathrm{w^T\ xi} + \mathrm{b}\ \right) - 1\right)\right) \quad \text{----------(5)}$$

$$\partial L / \partial w0 = 0 \text{ l i .e } w0 = \sum \alpha i \text{ yi xi} \text{ -----------(6)}$$

And $\partial L / \partial b0 = 0$ l i .e $\sum \alpha i \text{ yi} = 0$ -----------(7)

Substituting 6-7 in equation 5

An optimization problem is solved by Lagrange's Function

It is necessary to optimise saddle point s (w0, b0, α0) Langranges multiplier should be minimised wrt w, b and has to be maximised wrt nonnegative αi (αi ≥ 0) which can be solved by a primal or dual form

The functional equation is $\frac{1}{n}\sum_{i=0}^{n} \max(0{,}1 - \mathrm{Yi}\ (w^{\rightarrow} xi^{\rightarrow} - \mathrm{b})) + \lambda\left|\left|\mathrm{w}\right|\right|^2$

4. Convolutional Neural Networks (CNN)

Convolutional networks Models (LeCun & Bengio, 1998)[10] is a biologically inspired models have been utilized in facial recognition hand written numeral recognition have good success in a real-time image and video recognition it provides better accuracy than other recognition techniques CNN utilize smaller receptive windows size. the training and testing happens densely over the whole image and over multiple scales. Recognition is done in real time with multiple images per person for multiple persons, we did not consider invariance to a high degree of rotation our aim was for rapid classification of images, we have used 25 persons multiple images taken at different times, with facial expression and with/without glasses. Convolutional neural network (CNN) is a feed-forwarding Artificial Neural Network applied for analyzing images, it's multilayer perceptron which doesn't require data pre-processing (Yan LeCun, Leon Bottou, Yoshua Bengio & Patrick Haffner, 1998)[1], also known as shift-invariant artificial neural networks (SIANN) due to their weights and transitional invariance. Derived from human brain (Matsugu, Mori, Mitari, & Kaneda, 2003)[11]neurons major advantage of CNN being no effort and independence from prior knowledge. CNN consist of:-

a. Input layer
b. Output layer
c. Multiple hidden layers

Convolutional Layer passes results to next layer by applying convolutional operations to input image. The convolution emulates response of individual neuron to visual stimuli. The receptive field of neurons are used to process data, the fully connected feed forward neural networks is a better classifier because it learns the features of data (Ng A, Ngiam J, Foo C, Mai Y, Suen C, Coates A, Maas A, Hannun A, Huval B, Wang T, 2015)[13]. The convolutional reduces free parameters and gives better and deeper results (Heravi, Aghdam, & Puig, 2015)[14]. CNN resolves vanishing or exploding gradient problem with back-propagation.

Pooling Layer: CNN may include local /global pooling layers combines output of neuron cluster to single neuron layer.

Flattening: All the values from different cells are stacked in one vector which becomes a feature to ANN. High numbers in the vector refers to specific features in the input image which represents the distinct feature of a face.

Fully connected Layer: It connects every neuron in one layer to all the neurons of other layers

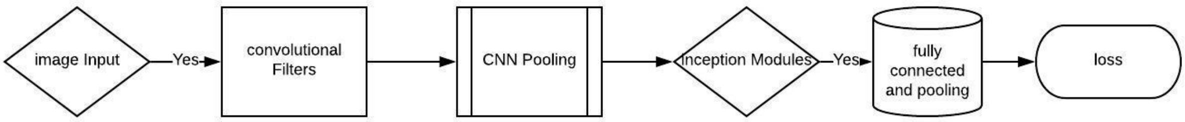


Fig-3: Flow chart for CNN

5. KNN- K nearest Neighbour

KNN is the most practical / non –parametric approach for facial recognition (Altman, 1992)[2] based on features such as eyes, nose, eyebrows, mouth, ears within the source image. It achieves its robustness by normalizing the size and orientation of face (Ebrahimpour & Kouzani, 1996)[16] KNN classify images in lesser time and with better accuracy faster execution time in KNN dominates SVM and other classifier algorithms. KNN classifier is an extension of simple nearest neighbour which employs non-parametric decision of query image based on the distance of its features from other image features. The distance between features can be measured through city-block distance, Euclidean distance or cosine distance.

- $d1(x,y) = \sum_{i=1}^{n=1} |\mathrm{Xi} - \mathrm{Yi}|$ **City-Block Distance**
- $d(x,y) = \left(\sum_{i=1}^{n=1} |\mathrm{Xi} - \mathrm{Yi}|\right)^{1/2}$ **Euclidean Distance**
- $dcos(x,y) = 1 - \frac{X^-}{|X|} + \frac{Y^-}{|Y|}$ **Cosine Distance**

K nearest Neighbour uses a query on closest samples. KNN classifier value depends on the number of samples used and their topological distribution KNN is among the simplest machine learning algorithm (Kaur, 2012)[17].

If K=1 then the image is simply assigned to the nearest neighbour class.

1. Every Data pixel in the dataset has a class label in set Class= {C1, C2, C3….., Cn}.
2. Through distance matrix, k-closest neighbours are found.
3. K-closest data points are analyzed to determine a common class label among the set.
4. The most common class label is assigned to data points.

Parameters selection depends on data larger values of *k* reduce noise in classification. A good *k* is selected through heuristic techniques. Presence of noise (irrelevant features) in dataset reduces the accuracy or if the feature scale is inconsistent. Evolutionary algorithms are employed to reduce noise and optimize feature selection.

KNN Model

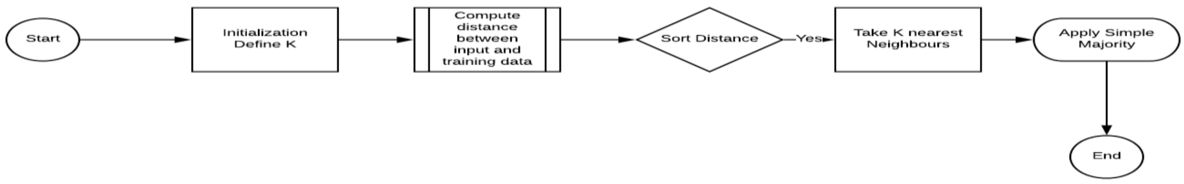


Fig-4: Block Diagram Representation for KNN Model

Implementation

The system architecture diagram of the study describes that a camera will start detect human faces using Haar Cascade algorithm. Given the circumstances that human faces detected, those images are passed into Face Classification Module and then we compare the recognition accuracies of SVM, KNN and CNN.

Face Classification Module

- **Pre-Processing:** Each image is read from a database and converted into matrix of dimension with respect to the image. The images are then standardised and ultimately divided into train and test set in the ration of 0.2.
- **PCA:** PCA is then applied to both train and test in order to extract the distinct features of all the images. Then the Eigen faces are computed from the pca components that explained 95% of the variance.
- **Classification Module:** The matrix thus obtained from PCA is fed into machine learning module for training.
- **Comparison Module:** We then compared the recognition accuracies of SVM, KNN and CNN both on train and test set and found CNN yielded the maximum accuracy.

Results

- **Face Detection:** Human face is detected using Haar Cascade algorithm.

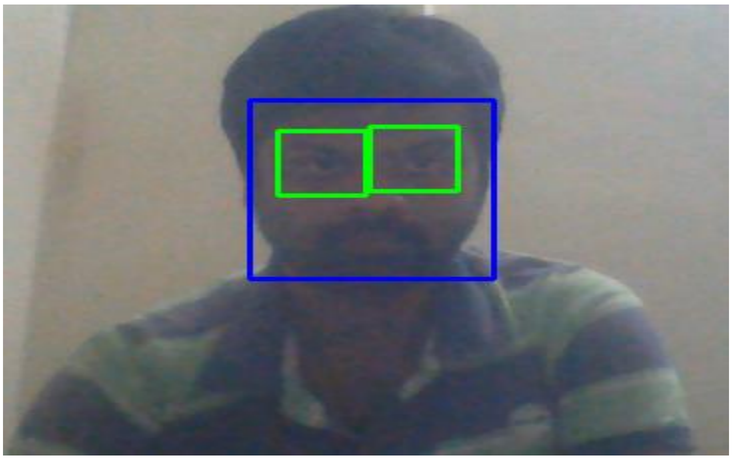


Fig-5: Human Face Detection

- **Image Matrix:**

```
Image Matrix of all images:

[[ 28  38  38 ...  14   9  20]
 [ 22  20  26 ...  15  22  20]
 [ 26  28  28 ...  23  25  32]
 ...
 [163 160 149 ... 106 123 123]
 [197 188 193 ...  69  72  75]
 [205 208 207 ...  72  79  79]]
```

Fig-6: Sample Image Matrix

- **Total variance Explained:**

The explained variance tells you how much information (variance) can be attributed to each of the principal components. This is important as while you can convert 4-dimensional space to 2-dimensional space, you lose some of the information. By applying **explained variance ratio,** one can ought to have first principal component capturing maximum variance and the second principal component contains second maximum variance of the total variance.

```
Explained Variance

[6348.42173806 3016.20553875 2617.99889663 1602.06073802 1372.54819737
  817.26359991  694.69486813  600.35792127  442.81973497  379.3156861
  358.40688644  300.85874822  232.26132967]
```

Fig-7: Variance Explained by PCA Components

- **PCA Components:**

Out of all components we extracted only top 13 components

```
PCA Components

[[-8.16282798e-03 -8.05494607e-03 -8.05187145e-03 ...  2.96807363e-03
   3.26171569e-03  3.52898352e-03]
 [ 4.40478785e-03  5.43701951e-03  6.07363507e-03 ...  1.22678933e-02
   1.20830261e-02  1.16625530e-02]
 [-1.58958359e-03 -1.94653432e-03 -1.82719873e-03 ... -6.13874920e-03
  -5.81410384e-03 -5.50118821e-03]
 ...
 [ 2.07361415e-04  1.63484665e-03  1.17701930e-03 ...  3.07268402e-05
   5.38020918e-04  8.57876205e-04]
 [ 1.51559138e-02  1.27425571e-02  1.20661614e-02 ... -2.18416193e-03
   2.05358893e-03  3.41778497e-03]
 [-3.81934868e-03 -4.78936593e-03 -5.73630761e-03 ... -8.84821877e-03
  -3.84752512e-03 -1.63565478e-03]]
```

Fig-8: Components that explained 95% of variance

- **Transformed Train Matrix**

```
Transformed Train Matrix

[[-7.78133706e+00  7.80735176e+01 -4.72388017e+01 -3.21700722e+01
   3.46629456e+01 -5.30487816e+01 -1.26818559e+00  5.16964679e+00
  -2.81771889e+01 -1.40622504e+01 -1.08598130e+01  4.19495726e+01
  -1.57045720e+01]
 [-9.41493988e+00  8.41856001e+01 -3.46192184e+01 -3.41090115e+01
   3.34065418e+01  1.24809604e+01 -9.91871699e+00  2.94785307e+01
   2.30434071e+01  2.80776949e+01  5.62488591e+01 -9.52507939e+00
  -9.41140860e+00]
 [ 1.92964518e+01 -3.36820529e+00  1.14586418e+01  6.10859923e+00
   6.51157375e+01  4.04269359e+01 -2.45587503e+01  2.12939211e+01
  -3.80374768e+01 -3.23571278e+01  5.76730716e+00 -9.87997253e+00
   1.70037847e+01]
 [-8.32645528e+01 -1.93718238e+01 -8.22711328e+01  1.72027008e+01
  -3.65643409e+01  5.57172196e+00 -4.07625536e+01  2.69994404e+01
  -8.13701019e+00 -1.03929686e+01 -8.33235426e+00 -1.25480384e+01
  -4.78757363e-01]
```

Fig-9: Sample Train Set Matrix after applying PCA.

- **Transformed Test Matrix**

```
Transformed Test Matrix

[[ 9.50620722e+01 -8.63918797e+00  3.48243795e+01  7.64189590e+01
  -1.36631250e+01 -3.36881579e+00  4.64989188e+01  1.55148579e+01
   3.91031022e+00 -1.87248264e+00  1.50271660e+01 -5.75176944e+00
   1.03406153e-01]
 [ 9.99993558e+00 -3.58761375e+01  6.95904989e+00  7.86202193e+00
   6.67478993e+01  5.39771086e+01  2.11007235e+01 -5.75675793e+00
  -2.40112734e+01  8.88884355e+00 -1.53312362e+01  1.88794543e-01
  -3.48753427e+00]
 [ 1.71034299e+00 -3.63774249e+01 -2.52169592e+01  1.59254138e+01
   4.65064756e+01  1.31450405e+01  6.05874530e+00 -2.82819093e+01
   5.88821525e+01  1.79757587e+00 -1.38999123e+01  1.71464343e+01
   9.75692224e-01]
 [-1.54956729e+01  9.63687594e+01  3.92884419e+01  9.58532693e+00
  -6.36848454e+00  1.42536409e+01 -1.00349678e+01 -3.09213464e+00
   1.27294989e+01  1.80184878e+01 -2.87051538e+00  1.84016763e+01
   2.09917134e+01]
 [ 1.08838988e+02 -7.88184293e+01  5.61328454e+01 -2.66226144e+01
  -2.75769115e+01 -2.67246342e+01 -2.43439969e+01  5.34538646e+01
  -2.24826182e+00  1.99543642e+01 -1.69098830e+01 -8.42127930e+00
  -4.05290296e+00]
 [-7.87247008e+00 -3.16742003e+00  8.76921517e+00  3.64989477e+01
   2.52693620e+01  2.14537726e+01 -3.31929777e+01  5.02800364e+00
   2.61706375e+01 -7.72155223e-01  2.06753580e+00  6.76895336e+00
   2.65649897e+00]]
```

Fig-10: Sample Test Set Matrix after applying PCA.

- **KNN**

- ➢ Train Set Size: 27
- ➢ Test Set Size: 6
- ➢ Train Set Accuracy: 1.0
- ➢ Test Set Accuracy: 0.66

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1
1	0.00	0.00	0.00	0
2	1.00	1.00	1.00	1
3	1.00	0.50	0.67	2
4	1.00	1.00	1.00	1
6	0.00	0.00	0.00	0
7	1.00	1.00	1.00	1
avg / total	0.83	0.67	0.72	6

Fig-11: Classification mAtrix for KNN.

- **SVM**
  - ➢ Train Set Size: 27
  - ➢ Test Set Size: 6
  - ➢ Train Set Accuracy: 1.0
  - ➢ Test Set Accuracy: 0.83

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1
1	0.00	0.00	0.00	0
2	1.00	1.00	1.00	1
3	1.00	1.00	1.00	2
4	1.00	1.00	1.00	1
7	1.00	1.00	1.00	1
avg / total	0.83	0.83	0.83	6

Fig-12: Classification mAtrix for SVM.

Prediction using SVM

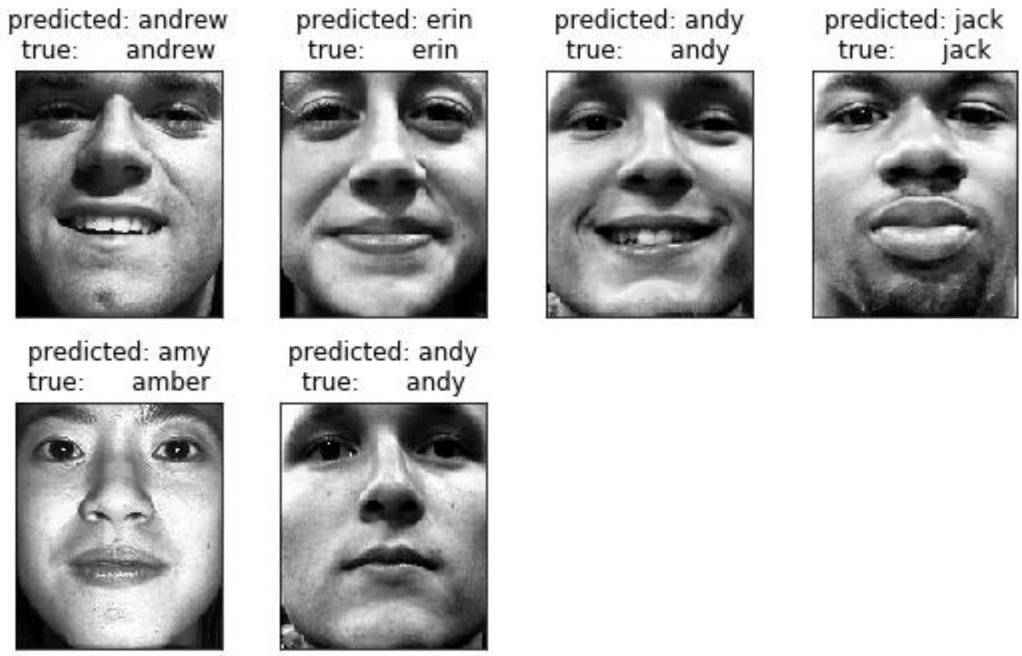
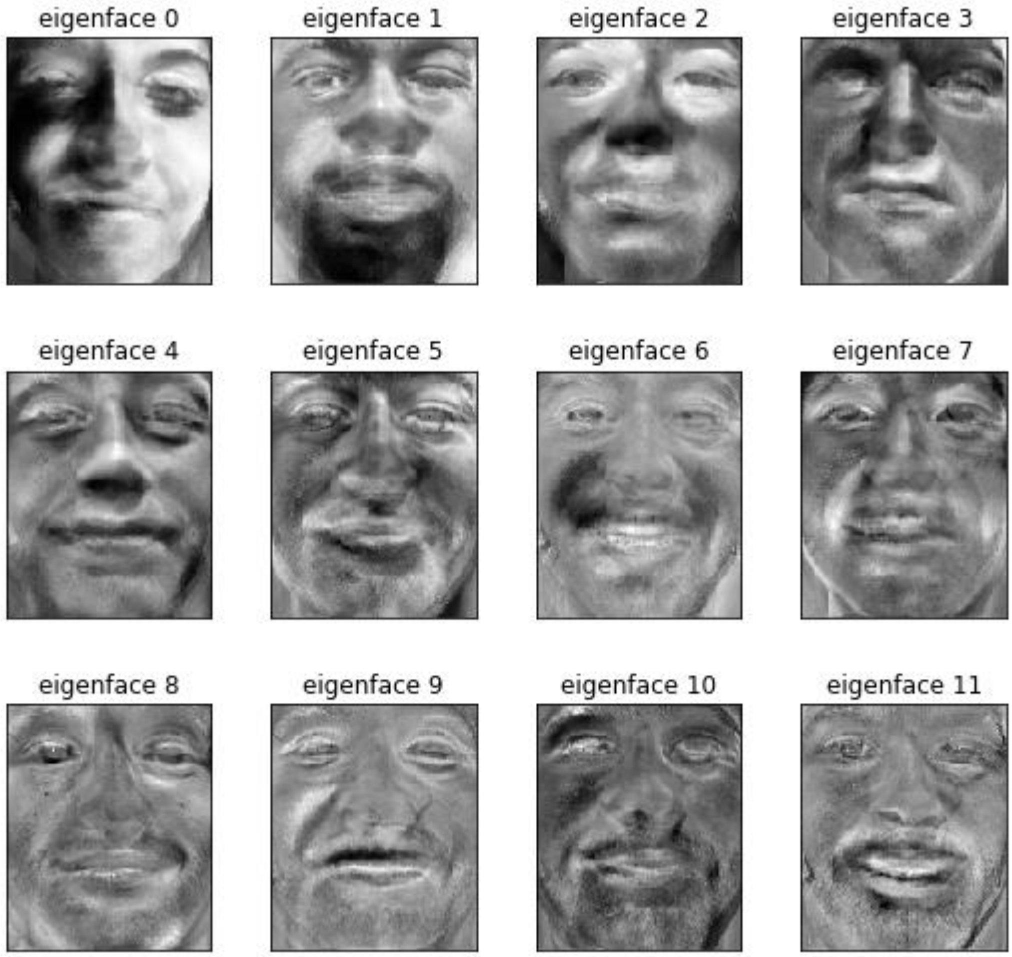


Fig-13: Prediction of Images in Test Set**.**

Eigen Faces



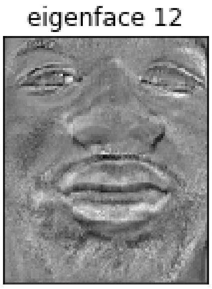


Fig-14: Eigen Faces.

- **CNN**
  - Train Set Size: 162
  - Test Set Size: 59
  - Train Set Accuracy: 1.0
  - Test Set Accuracy: 0.89
  - No of Epochs: 15

Therefore we can conclude that CNN as the best model out of other models being used because it yielded the maximum accuracy.

```
Epoch 12/15
162/162 [==============================] - 145s 894ms/step - loss: 0.0069 - acc: 1.0000
- val_loss: 0.1174 - val_acc: 0.8947
Epoch 13/15
162/162 [==============================] - 144s 891ms/step - loss: 0.0065 - acc: 1.0000
- val_loss: 0.1075 - val_acc: 0.8993
Epoch 14/15
162/162 [==============================] - 145s 897ms/step - loss: 0.0062 - acc: 1.0000
- val_loss: 0.1177 - val_acc: 0.9005
Epoch 15/15
162/162 [==============================] - 163s 1s/step - loss: 0.0058 - acc: 1.0000 -
val_loss: 0.1221 - val_acc: 0.8970
```

Fig-15: CNN Output

$$\mathbf{h}_{\mathrm{j}}(\boldsymbol{x}) = \begin{cases} -\boldsymbol{sj}, & fj < \theta j \\ \boldsymbol{sj}, & otherwise \end{cases}$$

Conclusions:

This paper we have shown comparative results of three different methods of facial recognition. This study detects the human faces using Haar-Cascade algorithm, global features are being extracted using PCA. We have seen that KNN has the lowest recognition accuracy and CNN yielded the best validation accuracy. There are two factors that affect the accuracy of the system: facial recognition and face viewpoints. The challenge is that all the images have to be in same dimension and of same colour depth otherwise the feature extraction will be inconsistent. The biggest challenge for CNN is the number of images in order to achieve a considerable amount of recognition accuracy in CNN the number of training images should be large.

This above research will help developers to choose the best algorithm for facial recognition which can be implement in security systems, retail stores and many other applicable areas. In future efforts can be made to test on large set of images in order to improve the accuracy of CNN. Also efforts can be made to study other machine learning classification algorithms and combine some of them to build a complex system so that they could have larger recognition accuracy and can deal with large amount of data.

References

1. Yan LeCun, Leon Bottou, Yoshua Bengio & Patrick Haffner. Gradient Based Learning Applied to Document Recognition. (1998). Retrieved from http://ieeexplore.ieee.org/document/726791/#full-text-section
2. Altman, N. (1992). An Introduction to Kernel and Nearest- Neighbor Nonparametric Regression. *The American Statistician*, *46*(3), 175–185. https://doi.org/10.2307/2685209
3. Çarıkçı, M. üg., & Özen, F. (2012). A Face Recognition System Based on Eigenfaces Method. *Procedia Technology*, *1*, 118–123. https://doi.org/10.1016/j.protcy.2012.02.023
4. Chakrabarti, D., & Dutta, D. (2013). Facial Expression Recognition Using Eigenspaces. *Procedia Technology*, *10*, 755–761. https://doi.org/10.1016/j.protcy.2013.12.419
5. Chen, J., & Jenkins, W. K. (2017). Facial recognition with PCA and machine learning methods. *Midwest Symposium on Circuits and Systems*, *2017–Augus*, 973–976. https://doi.org/10.1109/MWSCAS.2017.8053088
6. Detsing, J., & Ketcham, M. (2017). Detection and facial recognition for investigation. *2nd Joint International Conference on Digital Arts, Media and Technology 2017: Digital Economy for Sustainable Growth, ICDAMT 2017*, *438*, 407–411. https://doi.org/10.1109/ICDAMT.2017.7905002
7. Ebrahimpour, H., & Kouzani, A. (1996). Face Recognition using bagging KNN. *Int'l Conf. on CVPR, IEEE, Pp.*, (November), 209–216. Retrieved from https://pdfs.semanticscholar.org/4155/e00096d64cd0ca4fb653360bbae1f0625291.pdf
8. Heravi, E. J., Aghdam, H. H., & Puig, D. (2015). Analyzing Stability of Convolutional Neural Networks in the Frequency Domain, (2014), 1–9. Retrieved from http://arxiv.org/abs/1511.03042
9. Kaur, M. (2012). K-Nearest Neighbor Classification Approach for Face and Fingerprint at Feature Level Fusion. *International Journal of Computer Applications*, *60*(14), 975–8887.
10. LeCun, Y., & Bengio, Y. (1998). Convolutional Networks for Images, Speech, and Time Series. *The Handbook of Brain Theory and Neural Networks*, (November), 255–258. https://doi.org/10.1109/IJCNN.2004.1381049
11. Matsugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, *16*(5–6), 555–559. https://doi.org/10.1016/S0893-6080(03)00115-1
12. Ng A, Ngiam J, Foo C, Mai Y, Suen C, Coates A, Maas A, Hannun A, Huval B, Wang T, S. T. (2015). Deep Learning Tutorial. https://doi.org/10.1007/s13218-012-0198-z
13. Phankokkruad, M., & Jaturawat, P. (2017). Influence of facial expression and viewpoint variations on face recognition accuracy by different face recognition algorithms. *Proceedings - 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2017*, 231–237. https://doi.org/10.1109/SNPD.2017.8022727
14. Rodavia, M. R. D., Bernaldez, O., & Ballita, M. (2017). Web and mobile based facial recognition security system using Eigenfaces algorithm. *Proceedings of 2016 IEEE International Conference on Teaching, Assessment and Learning for Engineering, TALE 2016*, (December), 86–92. https://doi.org/10.1109/TALE.2016.7851776
15. Saitta, L. (1995). Support-Vector Networks SVM.pdf, *297*, 273–297. https://doi.org/10.1007/BF00994018
16. Soo, S. (2014). Object detection using Haar-cascade Classifier. *Institute of Computer Science, University of Tartu*, *2*(3), 1–12.

17. Srivastava, D. K., & Bhambhu, L. (2010). Data classification using support vector machine. *Journal of Theoretical and Applied Information Technology*, *12*(1), 1–7. https://doi.org/10.1016/j.jmr.2005.03.018
18. Paul, S., Acharya, S., & Bhuva, K. (2018). A REVIEW ON PROVIDING LOYALTY DISCOUNTS TO CUSTOMERS IN RETAIL STORES, 22(3), 1–8.