International Journal of Data Science and Big Data Analytics

Publisher's Home Page: https://www.svedbergopen.com/

SvedbergOpen
DISSEMINATION OF KNOWLEDGE

Research Paper

Open Access

A comparative study on facial recognition algorithms

Sanmoy Paul[1] and Sameer Kumar Acharya[2*]

[1]Data Science Department, NMIMS University, Mumbai, India. E-mail: sanmoy.27@gmail.com
[2]Data Science Department, NMIMS University, Mumbai, India. E-mail: sameeracharya.nmims@gmail.com

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 Principal Component Analysis (PCA), Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Convolutional Neural Network (CNN). The results showed out of the three algorithms we used CNN yielded the maximum accuracy.

Keywords: *Eigenvalues Haar Cascades facial recognition Principal Component Analysis (PCA), Convolutional Neural Network (CNN), K-Nearest Neighbor (KNN), Support Vector Machine (SVM)*
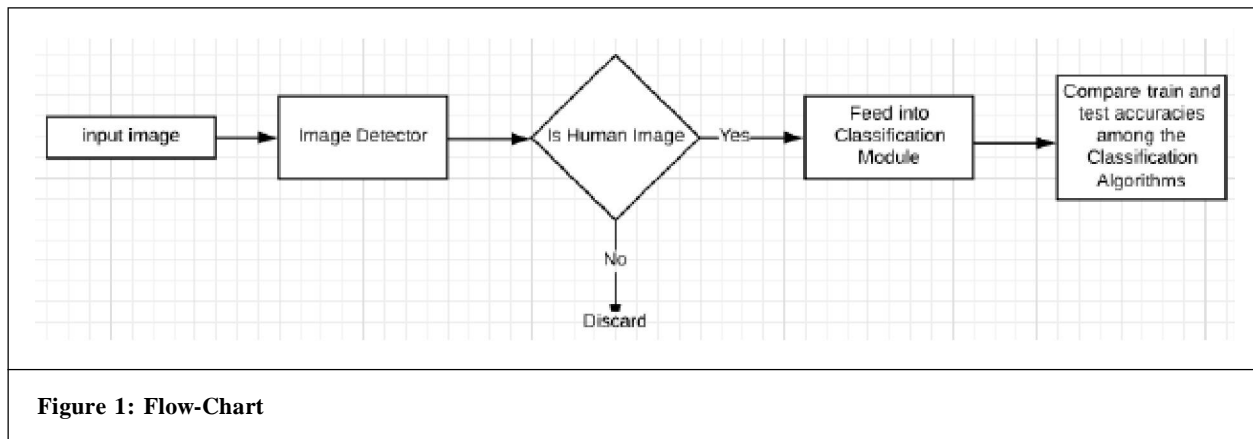
1. Introduction

A human face is a unique characteristic that differs from person to person , therefore face recognition becomes a credible source of identification apart from fingerprint scanners (Rodavia *et al.*, 2017). Face recognition is popular and widely used for personnel identification. The automatic facial recognition system involves the application of an intelligent artificial system to recognize the human faces under any circumstances. Today the study of facial recognition has involved a keen interest in pattern recognition, computer vision and other related fields. Camera is the only device for face recognition system. Face recognition provides an inexpensive and reliable personal identification which is applicable in many fields (Phankokkruad and Jaturawat, 2017). It is cheaper than biometric form of identification and can be used anywhere with low budget costs.

Recognition accuracy is an important factor in facial recognition system. However, there are many factors that affect the recognition accuracy. Environmental factors, quality of image, shifting and scaling of images are the common factors that affect the recognition accuracy. Sometimes these factors makes an image non ideal for recognition with decreased accuracy (Phankokkruad and Jaturawat, 2017). Other factors that affect the accuracy are face shape, texture, specs, hair, illumination etc. There are several external uncontrollable factors that affect the accuracy of the image recognition system. However there is publication of face recognition algorithms which reveals that each algorithm has certain characteristics and provides good accuracy in different aspects.

* Corresponding author: Sameer Kumar Acharya, Data Science Department, NMIMS University, Mumbai, India.
E-mail: sameeracharya.nmims@gmail.com

This research studies about some well know face recognition algorithms and makes a comparison of their recognition accuracies both on train and test set. Eigen faces, Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Convolutional Neural Network (CNN) are chosen in this experimental study. A variation of face viewpoints is the factor that has been used in the experiment to study the effect of recognition accuracy. In this way the advantages and disadvantages of different algorithms can be studied. Consequently it will help the developers to choose the best facial recognition algorithm in their field of implementation.

Flow-Chart

Figure 1: Flow-Chart

2. Literature review

2.1. Eigenfaces algorithm with PCA

The eigenfaces algorithm is the most commonly used methods in the field of facial recognition (Çarıkçı and Özen, 2012) have researched on eigenfaces method with smallest Euclidian distance is the person that resembles the most. We split the set of images into a training and test set. The face images in training set are converted into to face vectors Γ_i. The face vectors are then standardized by computing the mean of the face vectors then subtract the mean face from each of the face vectors to get as normalized face vector Φ_i (Paul *et al*., 2018).
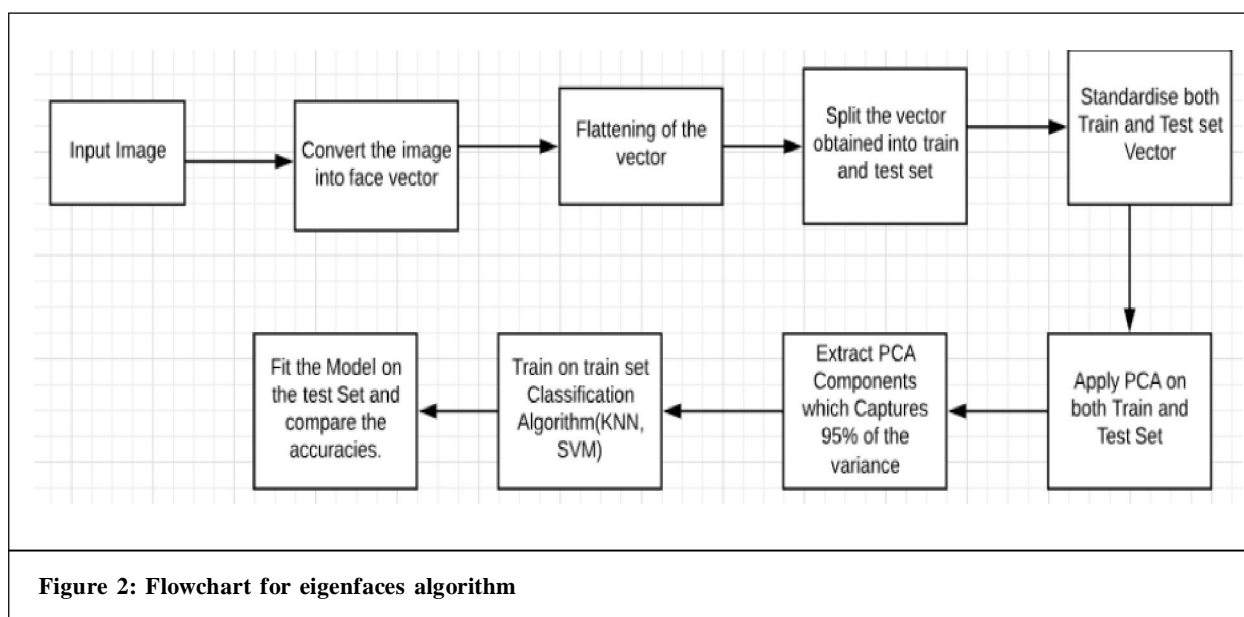
$$\Phi_i = \Gamma_i - \psi$$

where $\psi = 1/M \, \Sigma\_ (i = 1)^\wedge M \, \Gamma_i$, ψ is the average face vector.

The covariance matrix *C* is given by

$$C = 1/m \, \Sigma\_ (n = 1)^\wedge M \Phi_n = AAT \, (N2 \times N2 \, \text{matrix})$$

where $A = [\Phi_1, \Phi_2, ..., \Phi M]$, each normalized vectors in each column makes up *A*, where *A* is

*N*2 × *M* matrix Paul *et al*. (2018). Now choosing *k* significant vectors from the eigenvector space becomes a problem. Therefore, we use Principal Component Analysis (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* × *M* dimension. Similar methods have been further researched down by Chen and Jenkins (2017). Now it becomes easier to find the *k* significant Eigenvectors from the face vector space. The covariance matrix returns *M* Eigenvectors each of *M* × 1 dimension. After PCA, *k* best eigenfaces are selected that explains maximum variance such that *k* < *M* which represents the total training set. Now the selected *k* eigenfaces must be of the original dimensionality of the face vector space so we have to map it back into the original dimensionality by $u_i = A * v_i$, where u_i the Eigenvector of higher dimensionality and v_i is the Eigenvector of lower dimensionality. Detsing and Ketcham (2017) also used PCA with eigenfaces in their research. Paul *et al*. (2018) in their review have found that PCA with eigenfaces is most commonly used to extract distinct features with a face from person to person. The dimensionality reduction not only made the computational problem easier but also helped in the reduction of noise which could have impacted our result. Now we represent each face in the training set as a linear combination of all *k* eigenvectors along with the average face vector multiplied by the weights *w*. The weight vector is represented as $\Omega = [w1, w2, ..., wk]$, this is the eigenface representation of the ith face and the weights for each face is calculated. Chakrabarti and Dutta (2013) also used PCA with eigenfaces for their research of facial expressions. The flowchart of our eigenfaces algorithm is mentioned (Figure 2).

Figure 2: Flowchart for eigenfaces algorithm

3. Haar-Cascade Classifiers Viola Jones

Object detection is a method for identifying the existence of an object of a certain class Soo (2014) by a method of image processing. Objects can be classified based on texture, color and shape. One can a employ color coded approach to identify a object, color-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 three 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, two 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 $h(x) = \sum_{j=1}^{m} \alpha j * hj(x)$

Weak classifier

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

4. Support Vector Machine (SVM)

SVM is a supervised learning model with associated learning algorithms (Saitta, 1995), 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 nonlinear classification using Kernel trick. SVM is capable of delivering higher classification accuracy (Srivastava and Bhambhu, 2010). SVM can be used to detect text, digit, image classification and object detection.

SVM constructs hyper-plane between two 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.

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

where $yn = 1$ or -1

$w . x + b = 0$ ...(1)

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

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

$w. \ xi - b > 1$ or $w. \ xi - b < -1$

the distance between the hyper plane is $(2 / |w|)$. Here $|w|$ can be minimized by

$w. \ xi - b > 1$ or $w. \ xi - b < -1$

which can be arranged as

$yi \ (\ w. \ xi - b) > 1 \ , \ 1 < i < n$ ...(2)

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

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

Hyperplane with maximum margin is known as optimal canonical hyperplane

$y_j[w^T.x_j + b] \geq 1, i = 1, 2 \ldots .I$ ...(4)

$$\sum_{i=1}^{n} \binom{n}{k} 1 LP = L(w,b,\alpha) = \frac{1}{2}\|w\|2 - \sum \ \alpha i(yi(w^t xi + b) - 1)$$

$$= \frac{1}{2} wT \ w - \sum \alpha i(yi(w^T xi + b) - 1))$$ ...(5)

$\partial L / \partial w0 = 0 \ 1 \ i \ .e \ w0 = \Sigma \ \alpha i \ yi \ xi$ ...(6)

And $\partial L / \partial b0 = 0 \ 1 \ i \ .e \ \Sigma \ \alpha i \ yi = 0$ ...(7)

Substituting 6-7 in Equation (5)

An optimization problem is solved by Lagrange's Function

It is necessary to optimize saddle point s $(w0, b0, \alpha0)$ Langranges multiplier should be minimized wrt w, b and has to be maximized wrt nonnegative αi $(\alpha i \geq 0)$ which can be solved by a primal or dual form

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

5. Convolutional Neural Networks (CNN)

Convolutional networks models LeCun and Bengio (1998) 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. 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 *et al.*, 1998), also known as Shift-Invariant Artificial Neural Networks (SIANN) due to their weights and transitional invariance. Derived from human brain Matsugu *et al.* (2003) 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 *et al.*, 2015). The convolutional reduces free parameters and gives better and deeper results (Heravi *et al.*, 2015). CNN resolves vanishing or exploding gradient problem with back-propagation.
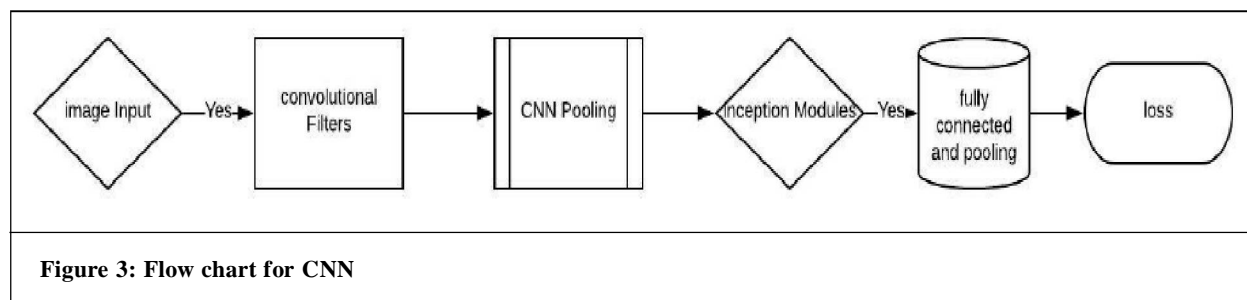

Figure 3: Flow chart for CNN

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.

6. KNN – K Nearest Neighbor

KNN is the most practical/non-parametric approach for facial recognition (Altman, 1992) 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 and Kouzani, 1996) 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 neighbor 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.

• $d1(x,y)=\sum_{i=1}^{n=1}|Xi-Yi|$ **City-Block Distance**

• $d(x,y)=\left(\sum_{i=1}^{n=1}|Xi-Yi|\right)^{1/2}$ **Euclidean Distance**

• $d\cos(x,y)=1\frac{\bar{x}}{|x|}+\frac{\bar{y}}{|y|}$ **Cosine Distance**
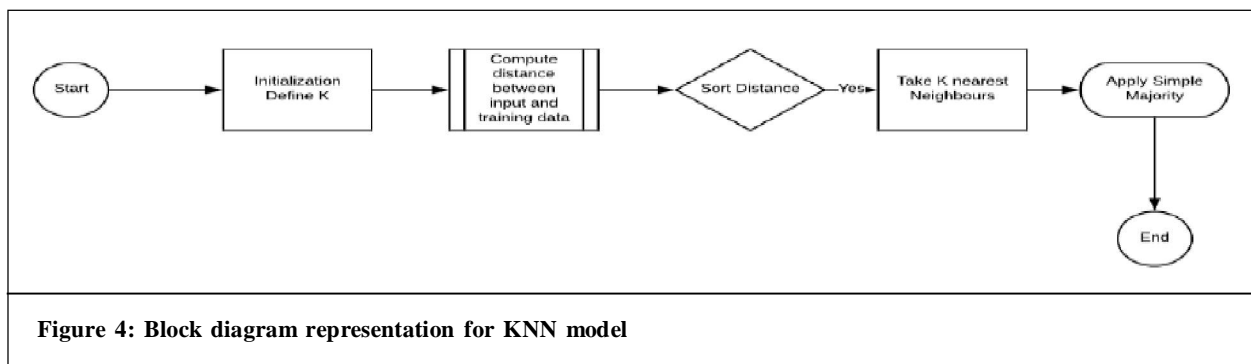
KNN uses a query on closest samples. KNN classifier value depends on the number of samples used and their topological distribution KNN is among the simplest machine learning algorithm (Kaur, 2012).

If $K = 1$ then the image is simply assigned to the nearest neighbor 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 neighbors 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

Figure 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 standardized 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 eigenfaces 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.

7. Results

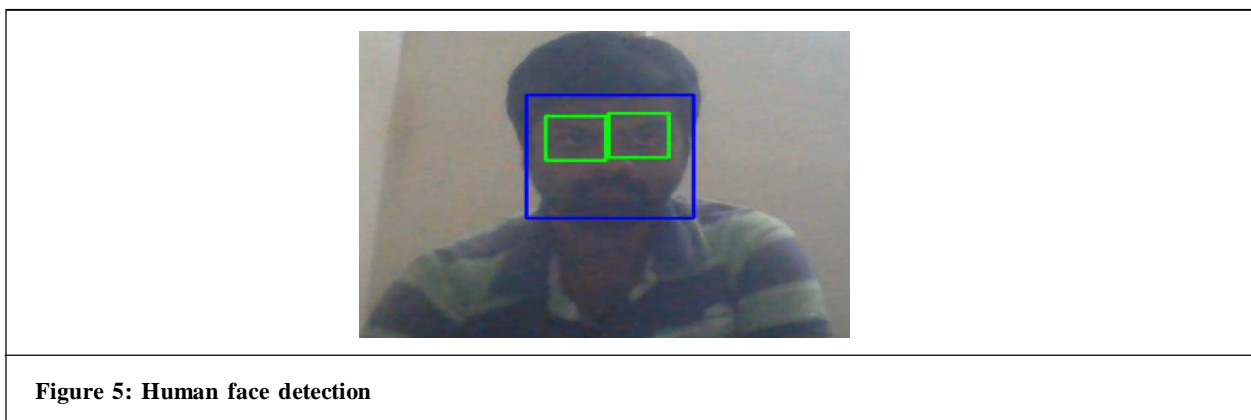- **Face Detection:** Human face is detected using Haar Cascade algorithm.

Figure 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]]
```
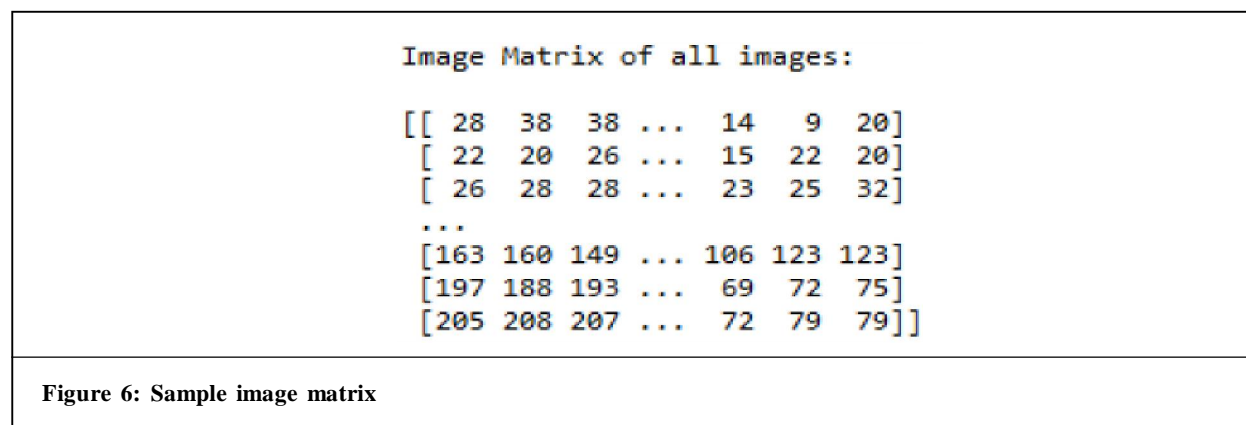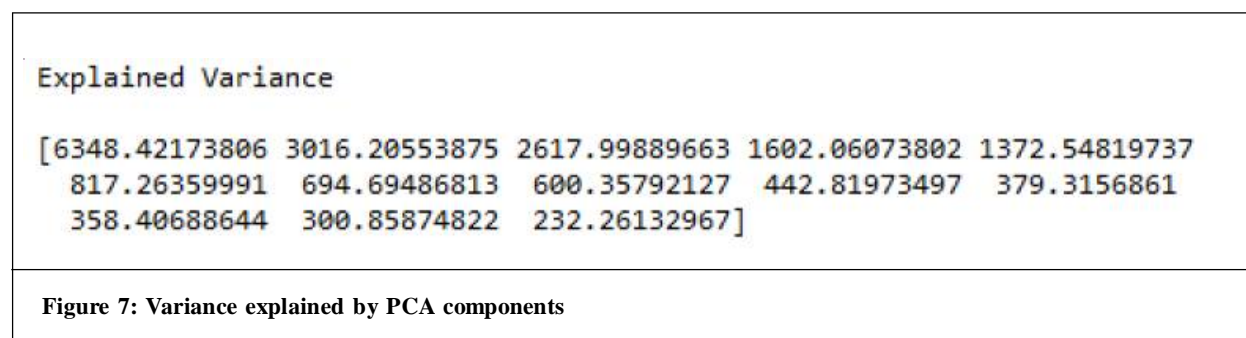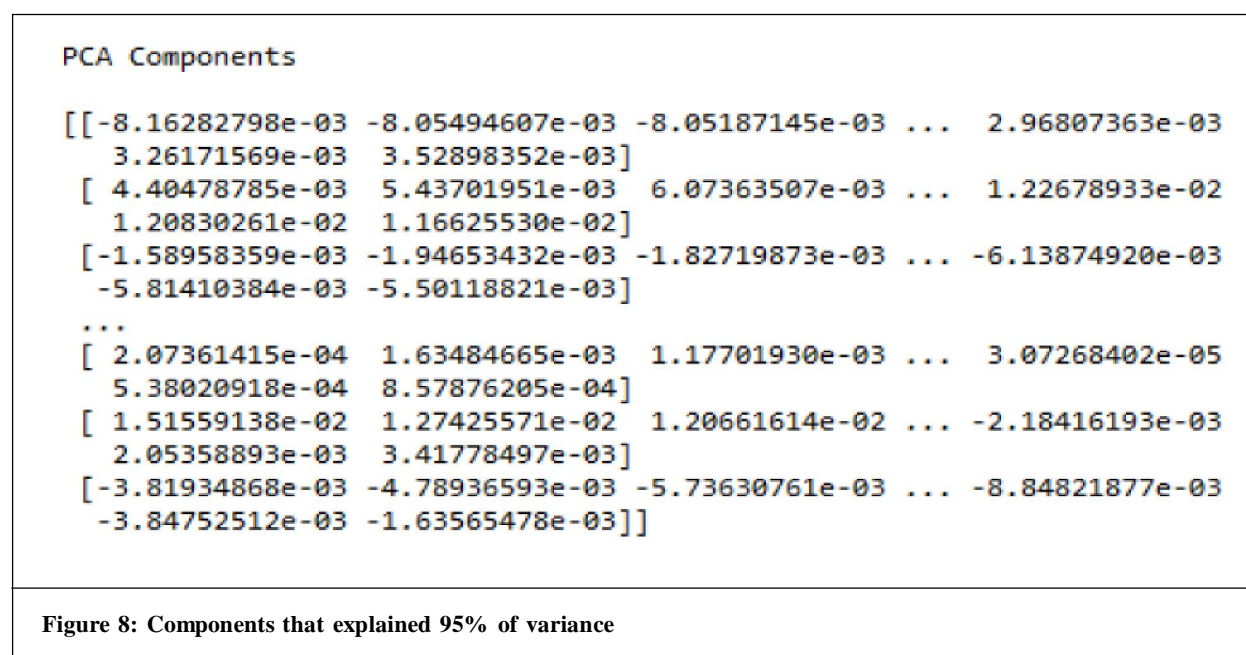
Figure 6: Sample image matrix

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]
```

Figure 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]]
```

Figure 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]
 ￢
```

Figure 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]]
```

Figure 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

Figure 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

Figure 12: Classification matrix for SVM
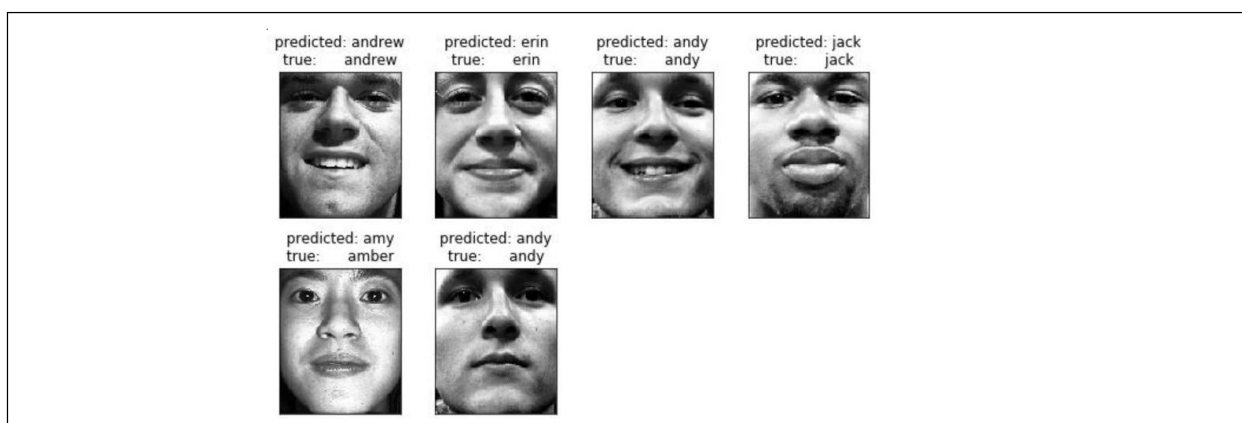
Prediction using SVM

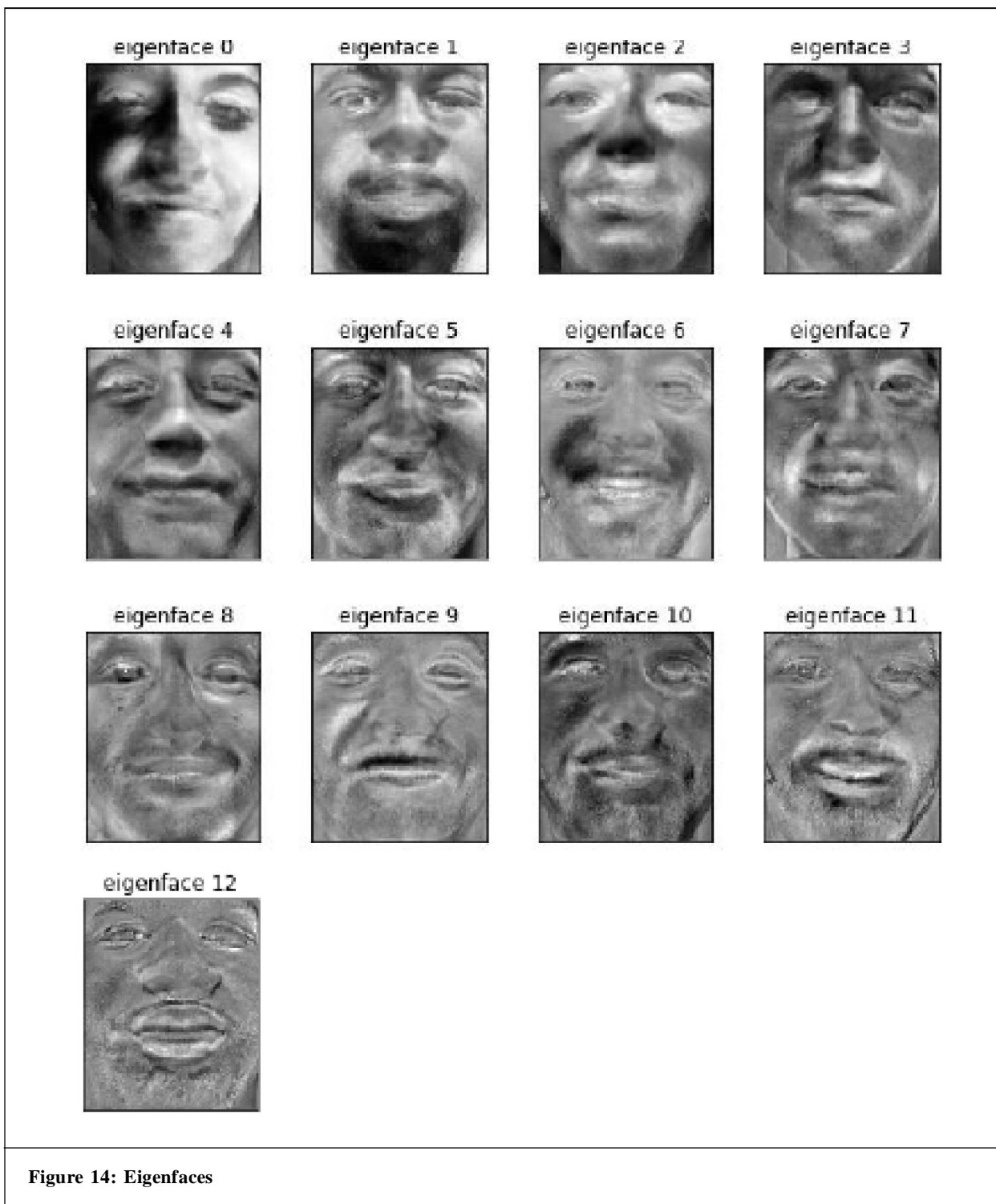**Figure 13: Prediction of images in test set**

Figure 14: Eigenfaces

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
```

Figure 15: CNN output

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

Conclusion

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 color 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

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

Çarıkçı, M. üg. and 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

Chakrabarti, D. and Dutta, D. (2013). Facial expression recognition using eigenspaces. *Procedia Technology*, 10, 755-761. https://doi.org/10.1016/j.protcy.2013.12.419

Chen, J., and 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

Detsing, J. and 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

Ebrahimpour, H. and 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/e00096d64cd0ca4fb653360 bbae1f0625291.pdf

Heravi, E.J., Aghdam, H.H. and Puig, D. (2015). Analyzing stability of convolutional neural networks in the frequency domain, (2014), 1-9. Retrieved from http://arxiv.org/abs/1511.03042

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.

LeCun, Y. and 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

Matsugu, M., Mori, K., Mitari, Y. and 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

Ng, A., Ngiam, J., Foo, C., Mai, Y., Suen, C., Coates, A., Maas, A., Hannun, A., Huval, B. and Wang, T, S. T. (2015). Deep Learning Tutorial. https://doi.org/10.1007/s13218-012-0198-z

Paul, S., Acharya, S. and Bhuva, K. (2018). A review on providing loyalty discounts to customers in retail stores. *Academy of Marketing Studies Journal*. 22(3), 1-8.

Phankokkruad, M. and 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

Rodavia, M.R.D., Bernaldez, O. and 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

Saitta, L. (1995). Support-Vector Networks SVM.pdf, 297, 273-297. https://doi.org/10.1007/BF00994018

Soo, S. (2014). Object detection using Haar-cascade Classifier. *Institute of Computer Science, University of Tartu*. 2(3), 1-12.

Srivastava, D.K. and 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

Yan, LeCun., Leon, Bottou., Yoshua, Bengio. and Patrick, Haffner. (1998). Gradient based learning applied to document recognition. Retrieved from http://ieeexplore.ieee.org/document/726791/#full-text-section