IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Face Recognition Using K-NN Algorithm Along With PCA

Nitin Kumar^{1*}, Gaurav^{2}, Deepak Kumar^{3}

1,2,3 Dept. of Computer Science and Engineering, SGT University Gurugram, India

Corresponding Author: nknitinyadav5@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7i5.352354 | Available online at: www.ijcseonline.org

Accepted: 19/May/2019, Published: 31/May/2019

Abstract- Face Recognition is an exciting task in the field of machine learning. Various techniques and methods have been used to solve the problem of face recognition. In this paper, we have shown that how K Nearest Neighbors algorithm along with Principal Component Analysis can be used to recognize a face efficiently. K nearest neighbor algorithm is a non parametric learning algorithm that works on target values of K nearest data points of the query point and finalize the value of the query point. PCA uses the concept of Eigen vectors. An Eigen vector represents an image. PCA finds K Eigen vectors corresponds to K higher Eigen values. So PCA algorithm is an efficient method for feature extraction in face recognition. Implementation is done using python programming language. This paper shows the effect of combination of above mentioned technologies and their edge cutting results.

Keywords-Face Recognition, KNN, PCA, Eigen vectors

I. INTRODUCTION

Face recognition includes the studies of automatically identifying or verifying a person from an image or video sequences [1]. In this paper, we have shown that how K Nearest Neighbors algorithm along with Principal Component Analysis can be used to recognize a face efficiently. K nearest neighbor algorithm is a non parametric learning algorithm that works on target values of K nearest data points of the query point and finalize the value of the query point. If the task is for classification then the class with maximum frequency among K data points will be the class for query point or if the task is for regression then the average of target values of all K data points will be target value of query point.

PCA(Principal Component Analysis) deals with curse of dimensionality i.e. reducing the number of features which are irrelevant and are noise features. Fig. 1 shows the effect of curse of dimensionality. The highly correlated features are removed from dimensions and it does so by using a specific algorithm. Feature extraction, and recognition can now be performed in real-time for images captured under favorable (i.e. constrained) situations [2]. PCA uses the concept of Eigen vectors. An Eigen vector represents an image. PCA finds K Eigen vectors corresponds to K higher Eigen values. So PCA algorithm is an efficient method for feature extraction in face recognition. The problem of face recognition can be efficiently solved using KNN along with PCA.

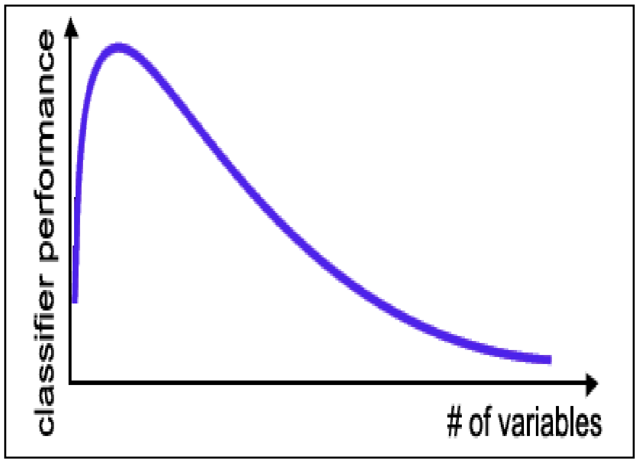

Fig 1. Curse of Dimensionality

II. PROPOSED METHODOLOGY

Now we will see how PCA and KNN can be used for problems such as Face Recognition. General workflow for face recognition is as:

1. Database of known faces
2. New face is presented and query is made to database
3. Database return a set of faces which are closest to the face which is in query
4. We choose a face which is most similar to face in the query

PCA method is used to extract features from face images. PCA calculates the Eigenvectors of the covariance matrix, and projects the unique features onto a lower dimensional feature. These Eigenvectors are also famous as Eigen faces [3]. Using PCA, Each image is represented as an Eigen vector. Consider the set of training images of size (64 X 64). This training image can be represented as a vector of (4096 X 1) using Raster scan. We subtract mean image from the training image to get mean normalized image. We get mean image from the dataset available in the database. Now we have a mean normalized image as a vector of size (4096 X 1). Now suppose we have N training images so we have a matrix of size (4096 X N) representing all training images as normalized images. Let this matrix be M.

To find the covariance matrix we multiply the matrix M with its transpose. Covariance matrix will be of size (4096 X 4096). Let it be R. Now R is represented as:

$R = P * D * P^T$
D : Diagonal matrix of same size as Matrix R (Containing Eigen values)
P : Orthonormal matrix of same size as Matrix R (Containing Eigen vectors)
P^T : Orthonormal matrix of same size as Matrix R (Transpose of P)

Once we have covariance matrix, we find Eigen values for that matrix. A Diagonal matrix D is formed using Eigen values in decreasing order. Now we find Eigen vectors and an orthonormal matrix is formed using these Eigen vectors. Let it be P.

Now we select K Eigen vectors corresponding to K highest Eigen values. So after selecting K Eigen vectors:
P1 : (4096 * K)
D1 : K * K
$P1^T$: (K * 4096)

Now we do dimensionality reduction:
As we considered earlier that our training images are of size (4096 * 4096). To get reduced dimensional image, we multiply $P1^T$ with mean normalized image of a training image.
(K * 4096) * (4096 * 1) = (K * 1).

This way we get reduced dimensional images of all training images. Once we get reduced dimensional images for all training images, we can apply K-NN algorithm to recognize the face given as query point. K-NN algorithm finds the highest matching face from the database of K features or dimensions.

KNN can be performed using Euclidean distance or by using weighted Euclidean distance between features.
e.g.

$$X_i = \{x_{i1},x_{i2},x_{i3},\ldots\ldots\ldots..,x_{in}\}$$

$$X_j = \{x_{j1},x_{j2},x_{j3},\ldots\ldots\ldots..,x_{jn}\}$$

Distance between these two vectors can be calculated as :

$$D(X_i,X_j)= \sqrt{\sum_{m=1}^{n} (x_{im}-x_{jm})^2}$$

Similarly , we can find the distance between all training data points from target data point. We can select those K data points which are nearest to target data point. There are more other methods that can be used to find K nearest neighbors for target data points. Signal to Noise ratio, Pearson coefficients etc. are methods that are used to find relation between training features or training data points.

Here we have seen that PCA finds principal components as Eigen vectors which mainly performs dimensionality reduction. Then we can use KNN to find those K data points or training images in case of face recognition, which are highly matched with the target data point or image. Then finally among these K data points or training images, highest matching image or data point can be selected. However there are other methods also such as OpenCV that is mainly used for such kind of tasks. OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real time computer vision, developed by Intel. The library is cross-platform. It focuses mainly on real-time image processing [4]. It is a classifier that helps in processing the images. The main focus is mainly on image processing and can be implemented on the latest algorithms [5].

III. IMPLEMENTATION USING PYTHON (SCIKIT LEARN)

So far, we have seen that KNN along with PCA can be used for face recognition where training dats is given as vector of images. Each training data represents features which are basically pixel value for the images. PCA performs feature reduction and KNN helps in classifying the target image. To implement these tasks, we can use Python language. Python programming language is highly useful for data science. As data analysis, data visualization are key parts of data science and these tasks can be effectively solved by using python. There are many libraries in python that can be used to solve such tasks [6].
The following are the steps for implementation:

1) Data split into training and testing data

   train_test_split from sklearn.cross_validation can be used to split the data into training and test data. Generally we split the training data as 75% and test data as 25%.

2) Calculation of Eigen faces i.e. Eigen vectors

   RandomizedPCA from sklearn.decomposition can be used to find Eigen vectors of specified size as arguments so that training and test data is converted into principal components.

3) Find the nearest neighbors and visualize the working of algorithm

   KNeighborsClassifier from sklearn.neighbors can be used to set the required number of nearest neighbors and then to visualize the working of algorithm.

Python is highly efficient language to work with any kind of data i.e. text, images, video etc. The Python scripting language is a freeware programming language and is an interpreted, general-purpose high-level programming language. Design philosophy emphasizes code readability [7]. It contains huge number of packages that are used of data analysis, data visualization etc. Usually, in Machine learning based projects, first we preprocess the data then design a model as per requirement using given data and then visualize the working of the model in the form of graphs or charts etc. to find out the accuracy. All these tasks can be performed using python programming language. From data preprocessing to visualizing the result, all kinds of task can be performed using python programming language.

IV. CONCLUSION

Here we can visualize that K-NN and PCA algorithms can be used to recognize a face efficiently. Even the use of python makes it more useful approach. PCA is instance based learning algorithm and is highly efficient when we train model on the basis of instances, not on the basis of classes directly. However as far as more better result is concerned, We can use some other algorithm instead of PCA to check better result as far as feature extraction or feature reduction is required. We can apply different values of K for different kind of data set to get better result. Even neural network is giving exciting results as far as classification is concerned. We can use neural network also for face recognition purpose.

REFERENCES

[1]. Mehran Kafai, Member, IEEE, Le An, Student Member, IEEE, and Bir Bhanu, Fellow, IEEE, "Reference Face Graph for Face Recognition", IEEE, ISSN - 1556-6013 , 2013.

[2]. Kavita , Ms. Manjeet Kaur," A Survey paper for Face Recognition Technologies", International Journal of Scientific and Research Publications, ISSN 2250-3153, Volume 6, Issue 7, July 2016.

[3]. Ashutosh Chandra Bhensle, Rohit Raja," An Efficient Face Recognition using PCA and Euclidean Distance Classification", IJCSMC, ISSN 2320–088X, Vol. 3, Issue. 6, June 2014.

[4]. P Y kumbhar , Mohammad attaullah , Shubham Dhere , Shivkumar Hipparagi, "Real time face detection and tracking using OpenCV", International Journal for Research in Emerging Science and Technology, ISSN: 2349-7610, Volume-4, Issue-4, Apr-2017.

[5]. Manik Sharma, J Anuradha, H K Manne and G S C Kashyap, "Facial detection using deep learning", IOP Publishing, 14th ICSET, 2017.

[6]. Gaurav, Ritu Sindhu, "Python as a key for data science", IJCSE, ISSN-2347-2693, Volume-6, Issue-4, Apr-2018.

[7]. Ing. Zdena Dobesova, "Programming Language Python for Data Processing", IEEE, International Conference on Electrical and Control Engineering (ICECE), **ISBN:** 978-1-4244-8165-1, 2011 .

AUTHORS PROFILE

Mr. Nitin pursued his Bachelor of Engineering from Chaudhary Devi Lal Institute of Engineering and Technology, Guru Jambheswar University, Haryana, He is currently pursuing his Master of Technology from SGT University, Gurugram India .