AI Solution for face recognition

**List of acronyms:**

ANN: Artificial neural network

BPNN: Back propagation neural networks

EBFDA: Elliptical box facial detection Algorithms

MLP: Multilayer perceptron

PCA: Principal component analysis

MPNN: multi layer perceptron neural network

BPP: Back propagation

# Introduction:

**Pattern recognition for Image processing and computer vision :**

Pattern recognition is an interesting field of study. It mainly deals with identifying and analysis of different kind of shapes, size of object and things. The process follows very simple steps to match the corresponding information and store it for analysis. The acquired data are classified for further information similarities on real world datas. pattern recognition uses machine learning algorithms for more robust information results.

**Computer vision**

Computer vision is a field of study where computer or machine are trained to extract different kind of information for analysis. Generally machine/computer are uses visual references in a form of pattern for generating geometric and mathematical classified information for processing. Mainly the computer vision is used in artificial intelligence domain for extracting the series of information from image or video frame.

Computer vision and pattern recognition is closely related to each other, with new design advancement computer are able to recognize pattern not limited with standard data but to broader hardware usage for pattern recognition like usage of webcam or CC camera.

**Unsupervised learning:**

**Dimensionally reduction:**

**Pca**

**ANN**

**Supervised learning**

**Backpropagation**

**PCA**

Introduction

Significantly, the dataset on which PCA technique is usually to be used needs to be scaled. The outcomes are likewise sensitive to the relative scaling. According to layman, it is a process of summarizing data. Imagine various wine bottles on a dining table. Each and every wine is identified by its very own characteristics such as color, strength, and so on. However redundancy will certainly emerge simply because most of them will certainly measure related qualities. What exactly PCA does in cases like this is sum it up every single wine in the stock with much less properties.   
  
Intuitively , Principal Component Analysis are able to provide the user with a lower-dimensional image , a reflection or perhaps "shadow" of this object while observed from its actual most informative viewpoint .

Principal Component Analysis (PCA) is an effective statistical method for variable minimization. It is implemented whenever variables are extremely interrelated. PCA gets to be a necessary tool for multivariate data analysis as well as unsupervised dimension minimization. PCA integrated with AI methods to strengthen functionality of several applications such as image processing, pattern identification, classification and even anomaly recognition.

Principal Component Analysis (PCA) is an effective statistical method for variable minimization. It is implemented whenever variables are extremely interrelated. PCA gets to be a necessary tool for multivariate data analysis as well as unsupervised dimension minimization. PCA integrated with AI methods to strengthen functionality of several applications such as image processing, pattern identification, classification and even anomaly recognition.

How does PCA works

**Background:**

Facial recognition

Facial recognition is a sophisticated implementation of analysis and pattern recognition. The process follows the pattern recognition mathematically and stores as data from face. This piece of software follows learning process and implementation using python. Program is utilizing deep learning algorithm along with one hardware component camera.

Implementation of facial recognition are available widely. With introduction of high quality camera and accessible datas it has been implemented on phones, laptop and other outlets such biometric faceprint in embassy.

**Problem Statement**

Challenge of facial recognition systems.  
scenario

Problem domain

graphs

Linear regression

Supervised learning process

# Related works:

**1987** Sirovich & Kirby = principal component analysis

Sirovich and Kirby (1987) showed that principal component analysis could be used on a collection of face images to form a set of basis features. These basis images, known as eigenpictures, could be linearly combined to reconstruct images in the original training set. If the training set consists of M images, principal component analysis could form a basis set of N images, where N < M. The reconstruction error is reduced by increasing the number of eigenpictures, however the number needed is always chosen less than M. For example, if you need to generate a number of N eigenfaces for a training set of M face images, you can say that each face image can be made up of "proportions" of all this K "features" or eigenfaces : Face image1 = (23% of E1) + (2% of E2) + (51% of E3) + ... + (1% En).

**1991** Turk & Pentland = Eigenface

In 1991 M. Turk and A. Pentland expanded these results and presented the eigenface method of face recognition.[3] In addition to designing a system for automated face recognition using eigenfaces, they showed a way of calculating the eigenvectors of a covariance matrix in such a way as to make it possible for computers at that time to perform eigen-decomposition on a large number of face images. Face images usually occupy a high-dimensional space and conventional principal component analysis was intractable on such data sets. Turk and Pentland's paper demonstrated ways to extract the eigenvectors based on matrices sized by the number of images rather than the number of pixels.  
  
Once established, the eigenface method was expanded to include methods of preprocessing to improve accuracy.[4] Multiple manifold approaches were also used to build sets of eigenfaces for different subjects[5][6] and different features, such as the eyes.[7]

**1996** Etemad & Chellapa = Fisher face

**2001** Viola & jones = AdaBoost + Haar Cascade

The Viola–Jones object detection framework is the first object detection framework to provide competitive object detection rates in real-time proposed in 2001 by Paul Viola and Michael Jones.[1][2] Although it can be trained to detect a variety of object classes, it was motivated primarily by the problem of face detection.

Haar Features Edit  
All human faces share some similar properties. These regularities may be matched using Haar Features.  
  
A few properties common to human faces:  
  
The eye region is darker than the upper-cheeks.  
The nose bridge region is brighter than the eyes.  
Composition of properties forming matchable facial features:  
  
Location and size: eyes, mouth, bridge of nose  
Value: oriented gradients of pixel intensities  
The four features matched by this algorithm are then sought in the image of a face (shown at right).  
  
Rectangle features:  
  
Value = Σ (pixels in black area) - Σ (pixels in white area)  
Three types: two-, three-, four-rectangles, Viola & Jones used two-rectangle features  
For example: the difference in brightness between the white &black rectangles over a specific area  
Each feature is related to a special location in the sub-window

Learning algorithm Edit  
The speed with which features may be evaluated does not adequately compensate for their number, however. For example, in a standard 24x24 pixel sub-window, there are a total of M = 162,336[4] possible features, and it would be prohibitively expensive to evaluate them all when testing an image. Thus, the object detection framework employs a variant of the learning algorithm AdaBoost to both select the best features and to train classifiers that use them. This algorithm constructs a “strong” classifier as a linear combination of weighted simple “weak” classifiers.

# Design solution:

Description and implementation:

For overcoming the design problem faced in previous implementation we’re going to introduce new and streamlined AI explication. The solution will follow certain steps and rules to minimize the complication of facial recognition in order to maximize the accuracy rate of the results. Our solution for facial recognition will use python programming language, specifically Python version 3.7+. Even though Matlab implementation would’ve been easier to adopt. Due to being enterprise software and its high price. we opted for python for its robust library selection, impressive documentation, excellent community for supports and troubleshooting also for being open source.

We will also be using other tools for generating solution some them are mentioned below:  
  
**Jupyter Notebook:**

It is an open-source web application which, is capable of creating and sharing information consitings of complex mathematical equations, numerical simulation, statistical modeling, data visualization and machine learning.

**Matplotlib:** It is a plotting library for numerical results and python language. We will be using this library extensively scatter plot, line plot, and 3-d plot.

**Numpy :**

Numpy is a library mainly used for high level of mathematical functions and also multidimensional arrays and matrices.

**OpenCv:**

Open source computer vision is a library used for computer vision. It also supports the deep learning frameworks and is based on real time computer vision.

**Pandas:**

It is also a library that is used for python programming language for data structures and operations for manipulating numerical tables.

**Scikit-learn:**

A machine learning library for python programming language. It’s and excellent tool for data analysis.

**Implementation (**process illustration**);**

**Design and Implementation**

Description;

When

Design solution follows following important steps

1. Determination of language - python specifically (version:3 )
2. Required library: numpy, openCv, haar cascade
3. Dev Environment : jupyter notebook.
4. Using image classifier of tensor flow.

# 

# 

# 

# Experiments:

Results and descriptions

# Conclusion:

(Analysis and importance)

Refrences

<https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework>

<https://stackoverflow.com/questions/1707620/viola-jones-face-detection-claims-180k-features>

<https://en.wikipedia.org/wiki/Pattern_recognition>

<https://en.wikipedia.org/wiki/OpenCV>

<https://stackabuse.com/implementing-pca-in-python-with-scikit-learn/>