Research on ML Algorithms For Face recognition

Algorithms Which are used for Face recognition

Traditional Algorithms

1. **Principal Component Analysis (PCA)** – Uses eigenfaces to reduce dimensionality.
2. **Independent Component Analysis (ICA)** – Focuses on statistically independent features.
3. **Linear Discriminant Analysis (LDA)** – Maximizes class separability.
4. **Elastic Bunch Graph Matching (EBGM)** – Uses graph-based representations for face matching.
5. **Viola–Jones Algorithm** – Mainly for face detection, using Haar-like features.

Deep Learning-Based Algorithms

1. **Convolutional Neural Networks (CNNs)** – Extract hierarchical features for high accuracy.
2. **FaceNet** – Maps faces into a Euclidean space for similarity comparison.
3. **DeepFace (by Facebook)** – Uses deep learning to classify faces.
4. **DeepID** – An earlier deep learning approach to face recognition.
5. **ArcFace** – Uses an improved loss function for high-performance facial recognition.

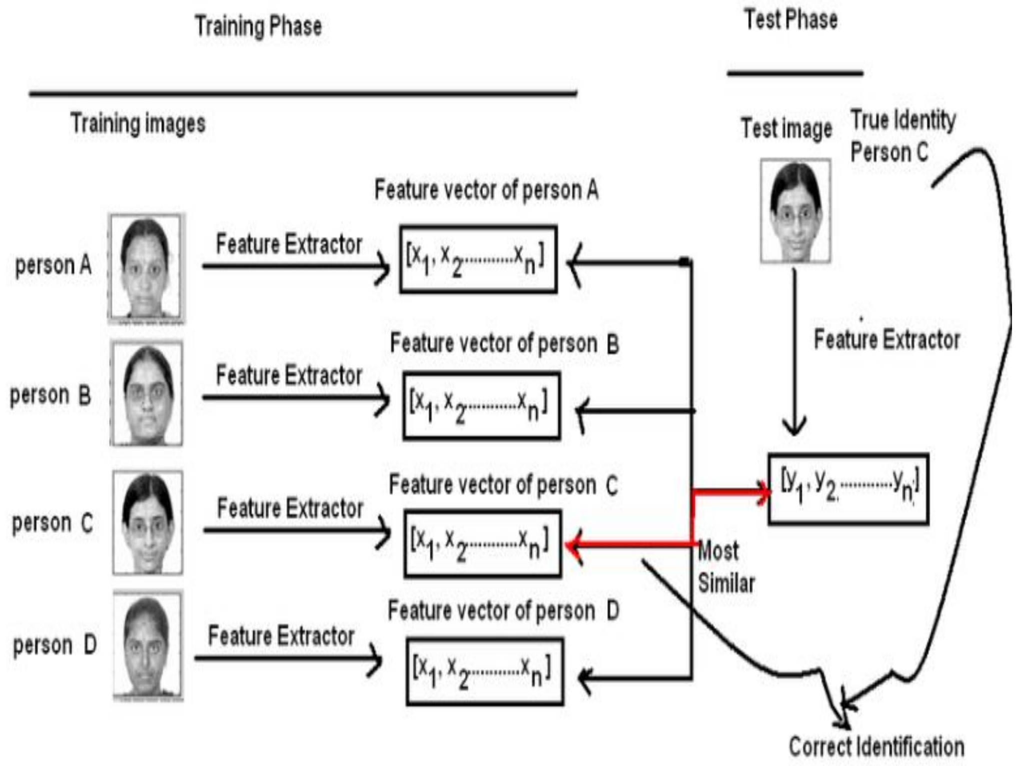
Based on the research the top algorithms are :
Principal Component Analysis (PCA)
Independent Component Analysis (ICA)
Linear Discriminant Analysis (LDA)
Convolutional Neural Networks (CNNs)
FaceNet

Principal Component Analysis (PCA) for Face Recognition

PCA is a technique used to **reduce the dimensionality** of data while preserving important information. It is widely used in **facial recognition** because it allows the system to identify faces efficiently by focusing on key features rather than processing every single pixel in an image.

How PCA Works in Face Recognition (Step by Step)

1. Image Representation as a Vector (Preprocessing Step)
2. Mean Face Calculation
3. Covariance Matrix Computation
4. Eigenfaces: Finding Principal Components
5. Feature Extraction: Projecting Faces into the Eigenface Space
6. Face Recognition: Comparing Feature Vectors



Reaason we use PCA is to solve the problem of overfitting.
PCA just tries to reduce the problem of overfitting
PCA – Converts High dimensions to low dimensions

1. Image Representation as a Vector (Preprocessing Step):

Every image in a dataset is **converted into a numerical format**. Computers don't see images as we do; they see them as a grid of numbers (pixel intensity values).
For example, if we have a **100x100** grayscale image, it has **10,000 pixels**, and it can be represented as a **10,000-dimensional vector**.

Example: A simple **5x5** image represented as numbers:

255	200	180	210	190
220	180	175	200	170
200	160	150	180	160
190	140	130	170	150
180	130	120	160	140

This is **flattened** into a single **feature vector**:
📌 **[255, 200, 180, 210, 190, 220, 180, ..., 140]**

2. Mean Face Calculation

Once we have all face images converted into vectors, we compute the **mean face** by averaging all the pixel values of images in the dataset.
Example:
If we have 1,000 face images in our dataset, we calculate the average value of each pixel position across all images to form a **mean face**.
💡 The mean face is used to **center the data**, meaning we subtract it from each face image to remove unnecessary common features.

3. Covariance Matrix Computation

Since every face is now represented as a **vector**, PCA constructs a **covariance matrix** to find relationships between different pixels.

- The **covariance matrix** captures how pixel values in different images **vary together**.
- It helps in identifying **important patterns** in the data.
- Mathematical formula for covariance matrix:
    - C= N1 i=1∑N (X i − X¯)(X i − X¯) T

$$C = \frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})(X_i - \bar{X})^T$$

Where:

- Xi is the face vector.
- X¯bar{X}X¯ is the mean face.
- C is the covariance matrix.

4. Eigenfaces: Finding Principal Components

Eigenfaces are the key component of PCA for face recognition. **Eigenfaces are a set of features that best describe the variations in faces**.

Steps to Compute Eigenfaces:

- Compute the **eigenvalues** and **eigenvectors** of the covariance matrix.
- Eigenvectors represent **principal components**, which are directions in which data varies the most.
- **Eigenfaces** are the eigenvectors reshaped into images.

Example of Eigenfaces:

The first few eigenfaces contain the most **important features** of faces (like eyes, nose, and mouth).

Feature Extraction: Projecting Faces into the Eigenface Space

Each face is now represented as a **weighted sum of eigenfaces**. Instead of storing the full image, we just store the weights (coefficients) that describe how much each eigenface contributes.

◆ **Formula for Projection:**

$$w_i=u_i^T(X-\bar{X})$$

Where:

X is the input face image.

$\bar{X}$ is the mean face.

u_i is the eigenvector (eigenface).

w_i are the weights (feature vector).

6. Face Recognition: Comparing Feature VectorsNow that every face is represented as a feature vector, recognition is simply done by comparing the new face's feature vector to those stored in the database. ◆ Common Distance Metrics Used:Euclidean Distance: Measures similarity between two feature vectors.Cosine Similarity: Measures the angle between two vectors.

Formula for Euclidean Distance:

$$d=\sqrt{\sum_{i=1}^{N}(w_i-w_{i,train})^2}$$

The smaller the distance, the more similar the faces are!

PROS and Cons

✅ **Reduces Computational Complexity** – Instead of comparing full images, we compare smaller feature vectors.
✅ **Captures Important Facial Features** – Eigenfaces focus on what makes a face unique.
✅ **Works Well in Controlled Environments** – Good accuracy when lighting and poses are consistent.

🚫 **Limitations:**
❌ Sensitive to **lighting changes and facial expressions**.
❌ Doesn't work well with **occlusions (partially covered faces)**.

International Journal of Livestock Research eISSN : 2277-1964 NAAS Score -5.36 Vol 7 (5) May'1

Technical Note

Multivariate Statistical Data Analysis- Principal Component Analysis (PCA)

Sidharth Prasad Mishra*, Uttam Sarkar, Subhash Taraphder, Sanjay Datta, Devi Prasanna Swain[1], Reshma Saikhom, Sasmita Panda[2] and Menalsh Laishram[3]

Department of Animal Genetics and Breeding, Faculty of Veterinary and Animal Science, West Bengal University of Animal and Fishery Sciences, West Bengal, INDIA

[1]Department of Veterinary and Animal Husbandry Extension, Faculty of Veterinary and Animal Science, West Bengal University of Animal and Fishery Sciences, West Bengal, INDIA

[2, 3] Department of Livestock Production and Management, Faculty of Veterinary and Animal Science, West Bengal University of Animal and Fishery Sciences, West Bengal, INDIA

***Corresponding author:** sidpramishra44@gmail.com

Rec. Date:	Mar 17, 2017 11:40
Accept Date:	Apr 15, 2017 11:52
Published Online:	May 01, 2017
DOI	10.5455/ijlr.20170415115235

Abstract

Principal component analysis (PCA) is a multivariate technique that analyzes a data table in which observations are described by several inter-correlated quantitative dependent variables. Its goal is to extract the important information from the statistical data to represent it as a set of new orthogonal variables called principal components, and to display the pattern of similarity between the observations and of the variables as points in spot maps. Mathematically, PCA depends upon the eigen-decomposition of positive semi-definite matrices and upon the singular value decomposition (SVD) of rectangular matrices. It is determined by eigenvectors and eigenvalues. Eigenvectors and eigenvalues are numbers and vectors associated to square matrices. Together they provide the eigen-decomposition of a matrix, which analyzes the structure of this matrix such as correlation, covariance, or cross-product matrices. Performing PCA is quite simple in practice. Organize a data set as an m × n matrix, where m is the number of measurement types and n is the number of trials. Subtract of the mean for each measurement type or row x_i. Calculate the SVD or the eigenvectors of the co-variance. It was found that there were many interesting applications of PCA, out of which in day today life knowingly or unknowingly multivariate data analysis and image compression are being used alternatively.

Key words: Eigenvalue, Eigenvector, Linear Algebra, Matrix, Multivariate, PCA

How to cite: Mishra, S., Sarkar, U., Taraphder, S., Datta, S., Swain, D., & Saikhom, R. et al. (2017). Multivariate Statistical Data Analysis- Principal Component Analysis (PCA). International Journal of

B. Extracting Face Feature

This process can be defined as the process of extracting relevant information from a face image. In feature extraction, a mathematical representation of original image called a biometric template or biometric reference is generated, which is stored in the database and will form the basis (vector) of any recognition task. Later these extracted features are used in recognition. After that greyscale pixel is considered as initial feature.

C. Recognition of Face

In this process, once the features are extracted and selected, the next step is to classify the image. For that appearance-based face recognition algorithms use a wide variety of classification methods. Such as PCA, LDA, Fisher face etc. In classification, the faces are compared for the similarity between faces from the same individual and different individuals after all the face images in database are represented with relevant features. Sometimes feature extraction & recognition process are done simultaneously.

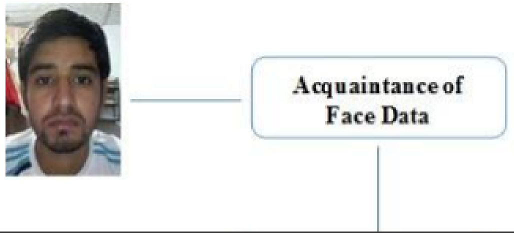
IV. ADVANTAGES AND DISADVANTAGES OF FACE RECOGNITION SYSTEM

- **Advantages:-**

1. Convenient, social acceptability.
2. More user friendly.
3. Inexpensive techniques of identification.

- **Disadvantages:-**

1. Problem with false rejection when people change their hairstyle, grow or shave a beard or wear glasses.
2. Face recognition systems can't tell the difference between identical twins.



V. ALGORITHMS FOR FACE RECOGNITION SYSTEM

There are different types of algorithm that can be used for face recognition. Some of them are listed below.

1. Principal Component Analysis (PCA).
2. Independent Component Analysis (ICA).
3. Linear Discriminant Analysis (LDA).
4. Elastic Bunch Graph Matching (EBGM).
5. Fisherfaces.

Principal Component Analysis (PCA): - It is a statistical approach used for reducing the number of variables in face recognition. It involves the extracting the most relevant information (feature) contained in the images (face). In this process, every image in the training set can be represented as a linear combination of weighted eigenvectors called as "Eigenfaces" [6] [11] [12]. These eigenvectors are obtained from covariance matrix of a training image set called as basis function. The weights are found out after selecting a set of most relevant Eigenfaces. Recognition is performed by projecting a new image (test image) onto the subspace spanned by the eigenfaces and then classification is done by distance measure methods such as Euclidean distance.

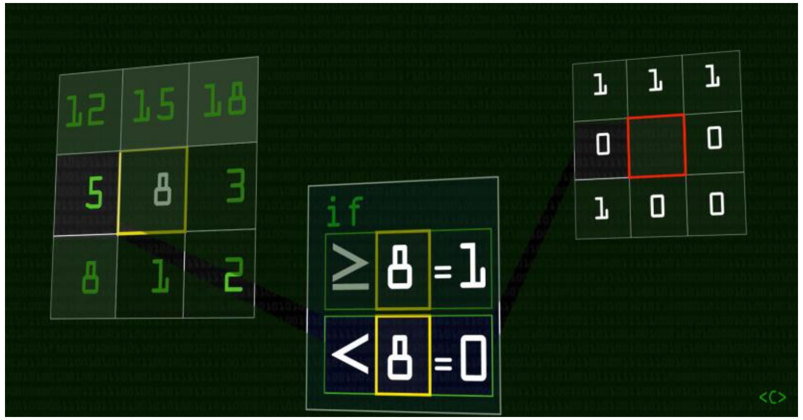
In PCA, faces are represented as a having an effect equal to the input mix of weighted eigenvectors called as Eigenfaces. These eigenvectors are got from covariance matrix of a training image put called as base purpose, use. The number of eigenfaces that got would be equal to the number of images in the training put. Eigen faces take better chances of the similarity between the bits of picture among images in a knowledge with the help of their covariance matrix. These eigenvector formed a new face space where the images are represented.

In PCA based face recognition, increase in the number of Eigen value will increase the recognition rate. However, the recognition rate saturates after a certain amount of increase in the Eigen value. Increasing the number of images and variety of sample images in the covariance matrix increases the recognition rate however noisy image decrease the recognition accuracy. In general, the image size is not important for a PCA based face recognition system. Expression and pose have minimal effect to the recognition rate while illumination has great impact on the recognition accuracy

Independent Component Analysis (ICA):- It minimizes both second-order and higher-order dependencies in the input data and

IJERT

2) Local Binary Patterns Histogram (LBPH)



How LBPH Works (With Images)

Step 1: Convert the Image to Grayscale

Since color information is not needed, we use only **grayscale intensity values**.

Step 2: Divide the Image into Small Regions

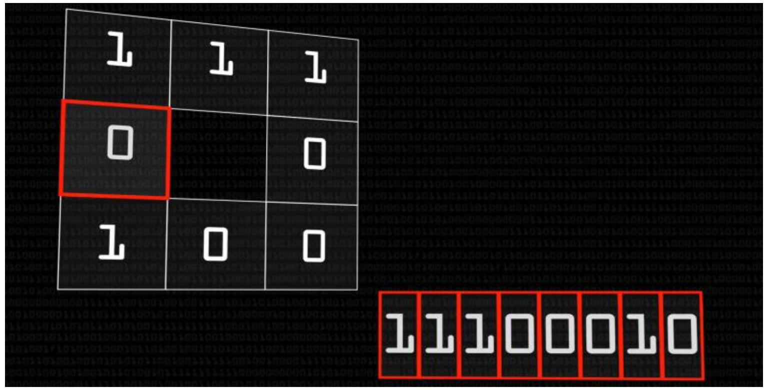
We split the image into **small square grids** (e.g., **8×8 pixels**).

Dividing the Face into a Grid

Step 3: Compute Local Binary Patterns (LBP) for Each Pixel

Each pixel is compared with its **8 neighboring pixels**:

- If the **neighbor is brighter**, mark 1
- If the **neighbor is darker**, mark 0

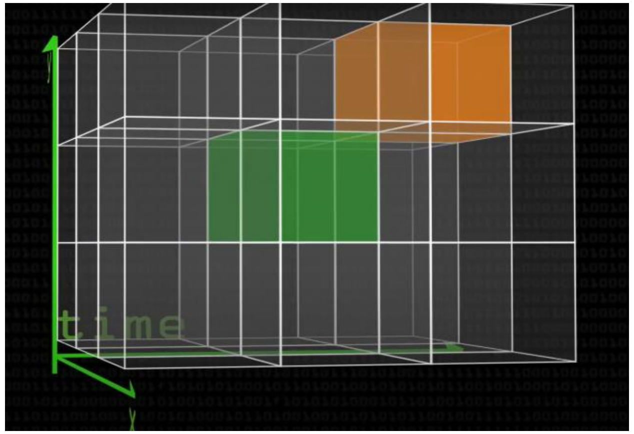


Step 4: Create a Histogram for Each Grid

•Each **small box (grid)** gets a **histogram** (count of different LBP values).

Step 5: Compare Histograms for Recognition

•Each new face **compares histograms** with stored ones.



✅ **Pros:**

✔️ Works well even if lighting changes.

✔️ **Fast and lightweight** (good for real-time).

❌ **Cons:**

🚫 Struggles with **partially hidden faces** (glasses, masks).

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

captured and stored in the dataset. During each session, faces will be detected from live streaming video of classroom. The faces detected will be compared with images present in the dataset. If match found, attendance will be marked for the respective student. At the end of each session, list of absentees will be mailed to the respective faculty handling the session.
The system architecture of the proposed system is given below,

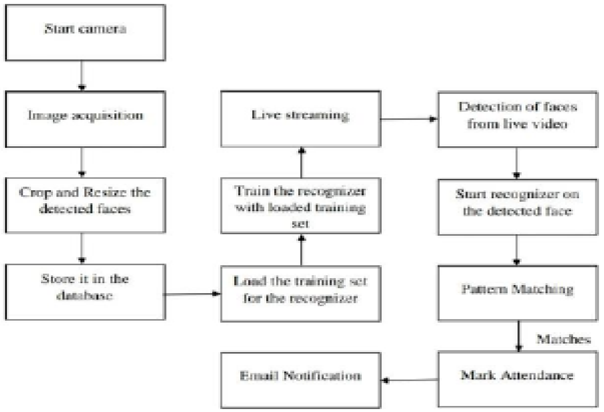


Fig.1. System Architecture

Typically this process can be divided into four stages,

1. Dataset Creation

Images of students are captured using a web cam. Multiple images of single student will be acquired with varied gestures and angles. These images undergo pre-processing. The images are cropped to obtain the Region of Interest (ROI) which will be further used in recognition process. Next step is to resize the cropped images to particular pixel position. Then these images will be converted from RGB to gray scale images. And then these images will be saved as the names of respective student in a folder.

2. Face Detection

Here we are using detectMultiScale module from OpenCV. This is required to create a rectangle around the faces in an image. It has got three parameters to consider- scaleFactor, minNeighbors, minSize. scaleFactor is used to indicate how much an image must be reduced in each image scale. minNeighbors specifies how many neighbors each candidate rectangle must have. Higher values usually detects less faces but detects high quality in image. minSize specifies the minimum object size. By default it is (30,30) [8]. The parameters used in this system is scaleFactor and minNeighbors with the values 1.3 and 5 respectively.

3. Face Recognition

Face recognition process can be divided into three steps- prepare training data, train face recognizer, prediction. Here training data will be the images present in the dataset. They will be assigned with a integer label of the student it belongs to. These images are then used for face recognition. Face recognizer used in this system is Local Binary Pattern Histogram. Initially, the list of local binary patterns (LBP) of entire face is obtained. These LBPs are converted into decimal number and then histograms of all those decimal values are made. At the end, one histogram will be formed for each images in the training data. Later, during recognition process histogram of the face to be recognized is calculated and then compared with the already computed histograms and returns the best matched label associated with the student it belongs to [9].

4. Attendance Updation

After face recognition process, the recognized faces will be marked as present in the excel sheet and the rest will be marked as absent and the list of absentees will be mailed to the respective faculties. Faculties will be updated with monthly attendance sheet at the end of every month.

IV. RESULTS AND DISCUSSIONS

The users can interact with the system using a GUI. Here users will be mainly provided with three different options such as, student registration, faculty registration, and mark attendance. The students are supposed to enter all the required details in the student registration form. After clicking on register button, the web cam starts automatically

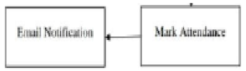


Fig.1. System Architecture

Typically this process can be divided into four stages,

1. Dataset Creation

Images of students are captured using a web cam. Multiple images of single student will be acquired with varied gestures and angles. These images undergo pre-processing. The images are cropped to obtain the Region of Interest (ROI) which will be further used in recognition process. Next step is to resize the cropped images to particular pixel position. Then these images will be converted from RGB to gray scale images. And then these images will be saved as the names of respective student in a folder.

2. Face Detection

Face detection here is performed using Haar-Cascade Classifier with OpenCV. Haar Cascade algorithm needs to be trained to detect human faces before it can be used for face detection. This is called feature extraction. The haar cascade training data used is an xml file-haarcascade_frontalface_default. The haar features shown in Fig.2. will be used for feature extraction.

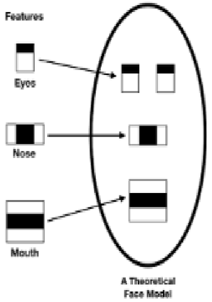


Fig.2. Haar Features

and then compared with the already computed histograms and returns the best matched label associated with the student it belongs to [9].

4. Attendance Updation

After face recognition process, the recognized faces will be marked as present in the excel sheet and the rest will be marked as absent and the list of absentees will be mailed to the respective faculties. Faculties will be updated with monthly attendance sheet at the end of every month.

IV. RESULTS AND DISCUSSIONS

The users can interact with the system using a GUI. Here users will be mainly provided with three different options such as, student registration, faculty registration, and mark attendance. The students are supposed to enter all the required details in the student registration form. After clicking on register button, the web cam starts automatically and window as shown in Fig.3. pops up and starts detecting the faces in the frame. Then it automatically starts clicking photos until 60 samples are collected or CRTL+Q is pressed. These images then will be pre-processed and stored in training images folder.

The faculties are supposed to register with the respective course codes along with their email-id in the faculty registration form provided. This is important because the list of absentees will be ultimately mailed to the respective faculties.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

Histogram of Oriented Gradients (HOG) + SVM

How HOG + SVM Works (With Visuals)

Step 1: Convert Image to Grayscale

•Removes color, keeping only brightness.

Step 2: Compute Edges Using Gradient Calculation

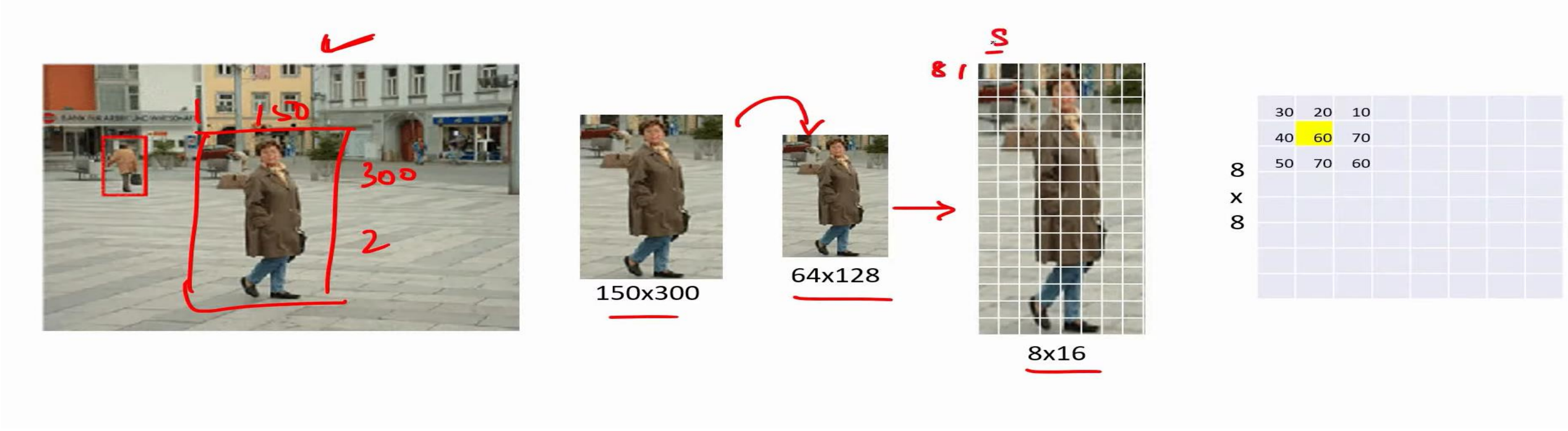
•We look at **changes in brightness** (edges).

•We calculate **how light transitions** from one pixel to another.

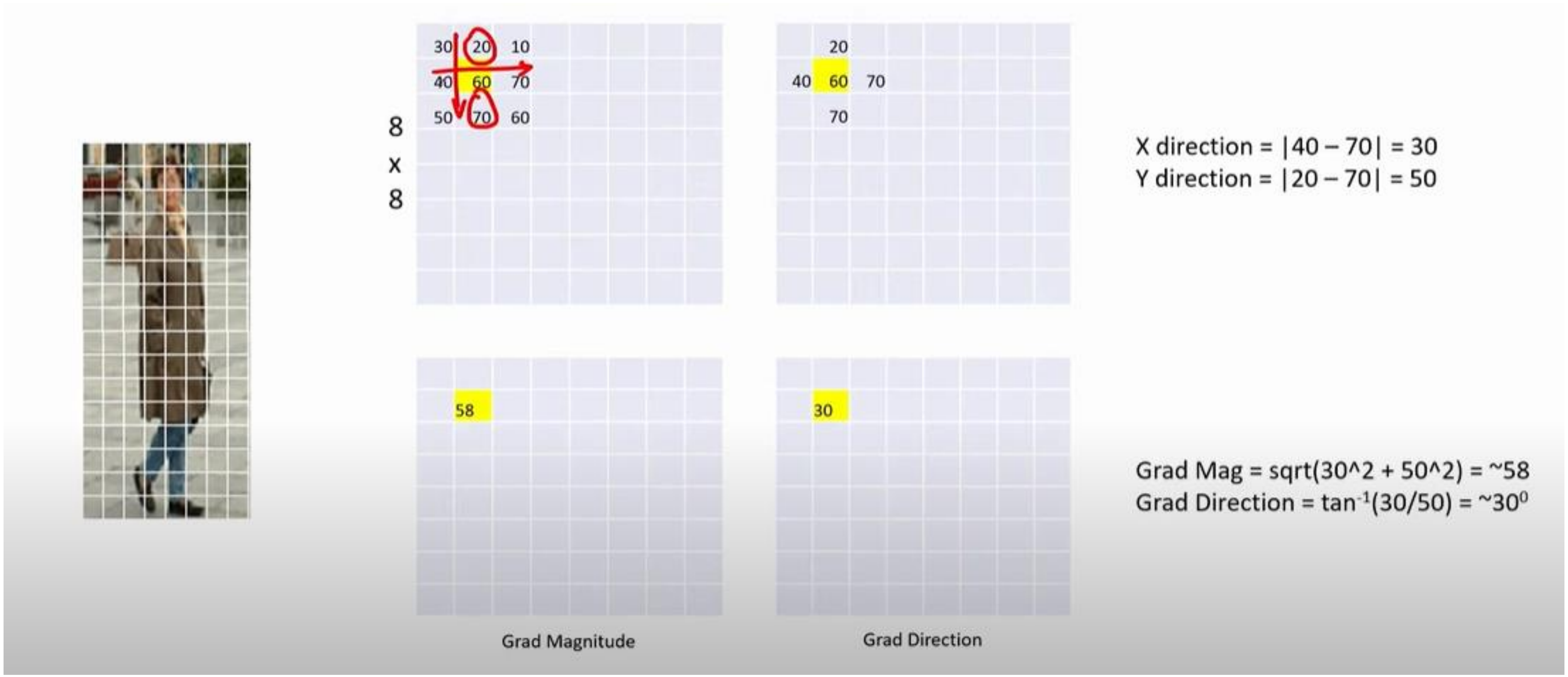
Step 3: Divide Image into Small Cells & Compute HOG Features

•The face is divided into **small squares (cells)**.

•Each **cell stores edge directions** in a **histogram**.



•At each pixel:**Multiply surrounding pixel values** by the kernel values.
•**Sum the results** to get new pixel values.



Step 4: Train SVM (Support Vector Machine) Classifier

•The algorithm learns which **patterns belong to which person**.

Construct HOG Features

•The image is divided into **small cells** (e.g., **8×8 pixels**).

•Inside each cell:

- We compute **histograms of gradient directions**.
- Each direction gets a **vote** based on its gradient magnitude.

✅ **Pros:**

✔️ Works well under **different lighting**.

✔️ **More accurate than LBPH**.

❌ **Cons:**

🚫 Not great at recognizing faces with **masks/glasses**.

B. HOG

The HOG (Histogram of Oriented Gradients) descriptor used for object detection, was presented in [4] by Dalal& Trigs. The object describes in the images by edge directions and the distribution of intensity.

To extract discriminant information, the image is partitioned into cells, and for each cell, a histogram of oriented gradients is computed.

Within each cell, the contribution of each pixel is determined by computing its gradient using the Sobel 1-D operator. The Sobel operator is applied both horizontally and vertically to capture gradient information along these axes. The resulting gradients are then used to compute weighted votes for an oriented histogram within the cell.

Horizontal and vertical gradients using the Sobel 1-D operator are calculated as follows:

It computes Horizontal and vertical gradient for each cell as follows:

$$dx = I(x+1, y) - I(x-1, y) \qquad (1)$$

$$dy = I(x, y+1) - I(x, y-1) \qquad (2)$$

The magnitude and orientation are computed from these gradients as follows:

$$m(x, y) = \sqrt{dx^2 + dy^2} \qquad (3)$$

A. SVM

The Support Vector Machine (SVM) algorithm serve as a useful tool for supervised machine learning algorithms, encompassing both classification and regression tasks. During the training phase, the provided dataset it used to learns how to predict the correct class based on the features it learns. The primary objective is to delineate a decision boundary that effectively segregate different data classes [2].

The factors that affect the decision boundary include its hyperparameters gamma, and C, gamma controls the curvature of the drawn boundary, while C is a regularizer that accepts a certain margin of error in learning stages to prevent overfitting (a problem where the classifier draw decision boundary that fit its training sample only). The decision boundary could be various type of function such as linear, polynomial, Radial Basis Function, etc [3]. Table 1 represent the advantages and disadvantages of SVM.

TABLE 1: SVM ADVANTAGES AND DISADVANTAGES

Advantage	Disadvantage
1.SVM is more efficient in high dimensional spaces. 2. SVMs are effective in scenarios where the number of measurements or features is greater than the number of samples in the training dataset. 3. Its versatility allows for the utilization of different kernel functions to accommodate various type of Data and decision boundaries.	1. When the number of features per data points exceeds the number of training data samples, SVM performance can suffer. 2. There is no probability-based justification for the classification above and below the classification hyperplane, as the support vector classifier operates by putting data points.

C. PCA

PCA (Principal Component Analysis) is a method employed to reduce the number of features (dimensions) by retaining only the most relevant dimensions and discarding unnecessary ones. It works as follows: Initially, it identifies the dimensions that exhibit the highest variance among the data samples.

Subsequently, it ranks these dimensions based on their eigenvalues, which quantify their impact on the dataset. Finally, PCA eliminates the least significant components characterized by the lowest eigenvalues [5]. Table 2 represents the difference between two feature extraction methodologies.

TABLE 2: PCA AND HOG ADVANTAGES AND DISADVANTAGES

	Advantage	Disadvantage
PCA	1.Reduces Overfitting: reducing the number of features. 2. Improves the efficiency of the algorithm: if the input parameters are too high, then use PCA to accelerate the algorithm.	1. Before PCA, data standardization is required: all categorical characteristics must be transformed into numerical characteristics before PCA can be implemented. 2. Variables that are independent become less subject to interpretation: The key components are not as easy to read and analysable as the initial characteristics.
HOG	1. This Guided Histogram gradient displays geometric and picture metric shifts invariance. 2. Raising the object search quality, gamma and image colors should be normalized	1. The HOG uses magnitude values only Without considering adjacent pixels.

Comparsion of all the researched algorithms

Algorithm	Robust to Lighting?	Handles Occlusions?	Works in Real-Time?	Accuracy
LBPH	✅ Yes	❌ No	✅ Yes	⭐⭐⭐
HOG + SVM	✅ Yes	❌ No	✅ Yes	⭐⭐⭐⭐
PCA (Eigenfaces)	❌ No	❌ No	✅ Yes	⭐⭐
LDA (Fisherfaces)	✅ Yes	❌ No	✅ Yes	⭐⭐⭐
SIFT	✅ Yes	✅ Yes	❌ No	⭐⭐⭐⭐⭐

Deep learning algos :
Convolutional Neural Networks (CNNs)

References

- https://youtu.be/83x5X66uWK0?si=J2wU8ce1eYzOdoTQ
- https://www.researchgate.net/publication/316652806_Principal_Component_Analysis
- https://www.ijert.org/research/analysis-of-different-face-recognition-algorithms-IJERTV3IS111235.pdf
- https://www.researchgate.net/publication/341876647_Face_Recognition_based_Attendance_Management_System
- https://arxiv.org/pdf/1811.00116
- https://youtu.be/wpAwdsubl1w?si=xK5NHFlmOywYlp7I
- 10.1109/CoDIT62066.2024.10708142
- 10.1109/WiDS-PSU61003.2024.00034
- https://youtu.be/28xk5i1_7Zc?si=pPPbJyT2kG0GEtN3