**An Efficient Approach to Real-Time Face Recognition From Digital images.**

**CHAPTER ONE**

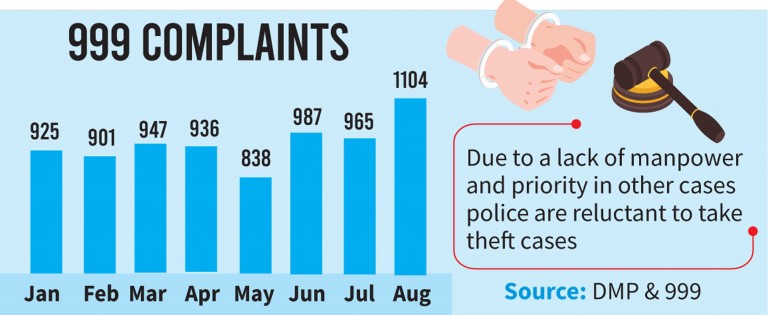
**1.1 Introduction**

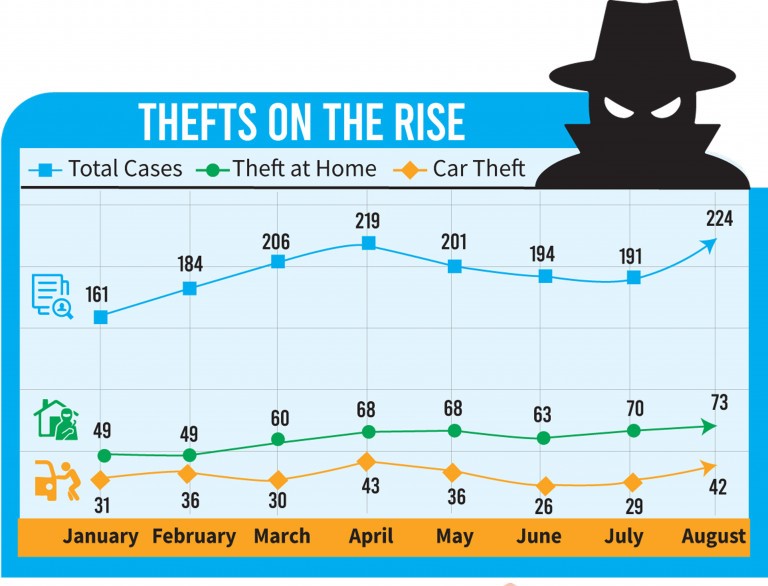
Human faces are significant components of human identity and a distinctive trait that varies from person to person. People are still using human guards to check their ID cards when entering a restricted or reserved area, which is neither an absolute nor efficient approach in this modern age. Furthermore, this system is easy to break through because anyone can use another's cards or create duplicate ones. Governments, Banks, and a few institutions are using Fingerprint systems for checking identities, Which is expensive and its equipment needs to be repaired repeatedly. On the other hand, Police are trying to track the victim using their manpower. It is also not an efficient way in this modern era.

"THE BUSINESS STANDARD" published news on 12 October 2022 with the headings "Most theft cases remain unsolved with little or no headway in probes". [[1](https://www.tbsnews.net/bangladesh/crime/most-theft-cases-remain-unsolved-little-or-no-headway-probes-512182)]

According to The Business Standard, at the crime review meeting at police headquarters, the issue of increasing theft came up for discussion. in the meeting, there were 516 theft cases across the country in July. The number increased to 618 in August.

According to DMP case data, 1580 burglary cases have been recorded from January to August 2022. The national emergency service hotline 999 received 7,503 calls reporting burglary from January to August. In January 925 theft incidents were reported to 999 while 1,104 cases were reported in August. The Statistics are given table:





As So solve this cases Face recognition will help as Face is the unique identity of every human being.

Facial recognition combined with the Internet of Things (IoT) will be much more secure than any other biometric system. Face recognition technology is related to the Internet of Things (IoT) because it can be integrated with IoT devices to enhance their functionality. For example, face recognition can be used in smart homes to unlock doors, control lighting, and adjust the temperature based on the presence of recognized individuals. In security systems, face recognition can be used to monitor and track individuals, as well as provide personalized access control. In the retail industry, face recognition can be used for customer identification and personalization and for tracking customer behavior and preferences. Overall, integrating face recognition technology with IoT devices can lead to more convenient and efficient experiences, while potentially improving security and privacy.

There are two parts to our projects. The First one is, Detecting faces from input images that are captured by any surveillance or webcams. This model also can find out the persons in group photos. The Last one is, it can recognize human identities in real-time live cams.

This project studies some well know face recognition algorithms and makes a comparison of their recognition accuracies both on the train and test sets. Eigen-faces, PCA, SVM, KNN, and 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.

**1.2 Objectives of the study**

The goal of this project is to make an automated security system based on artificial intelligence. That can provide us with more and more security without manpower at a low cost. This system uses Artificial Intelligence (AI) to learn from input data (images) and will remember those features for future face recognition. The principal objectives are:

* To develop a face recognition system.
* To ensure the accuracy and reliability of the system.
* To identify individuals based on their facial features.
* To match individuals against a large database of stored images.
* To maintain the privacy and security of personal data.
* To design the system to adapt and learn from new data.
* To continually improve the system's recognition capabilities over time.

**Outline of this Paper:**

This paper is organized into Five chapters. In Chapter 2.1, I have mentioned the name and brief

**CHAPTER TWO**

**Literature Review:**

A significant number of studies have been conducted on face recognition systems that operate in real-time. A common approach to face recognition is to extract facial features such as eyes, nose, and mouth and then compare them to a database of known faces. This method is known as feature-based recognition, which can be computationally expensive and time-consuming.

To overcome these limitations, researchers have developed an efficient approach to real-time face recognition using deep learning techniques. Deep learning has shown remarkable performance in various applications, including face recognition. This approach involves training a convolutional neural network (CNN) to learn the features of faces from a large dataset. The CNN can then be used to classify new faces in real-time.

A recent study by Zhang et al. (2021) proposed an efficient face recognition system that combines deep learning and dimensionality reduction techniques. The system utilizes a pre-trained CNN to extract features from the face images and then applies principal component analysis (PCA) to reduce the dimensionality of the features. The reduced feature vectors are then compared to a database of known faces using Euclidean distance. The experimental results showed that this approach achieves high accuracy in face recognition while maintaining low computational costs.

Another study by Liu et al. (2020) proposed a real-time face recognition system that uses a Siamese network to learn the similarity between faces. The Siamese network is a type of CNN that uses two identical networks to process two input face images and produces a similarity score. This approach does not require a database of known faces, making it suitable for real-time applications.

**CHAPTER THREE**

**PRELIMIONARY CONCEPTS:**



**Introduction To Machine Learning**

Machine learning is a branch of artificial intelligence that deals with creating algorithms that can learn and improve from experience, without being explicitly programmed. The concept of machine learning can be traced back to the mid-twentieth century when researchers began exploring the idea of creating computer programs that could "learn" from data. However, it wasn't until the late 1990s that machine learning began to take off as a viable field of study.

The history of machine learning can be broadly divided into three phases. The first phase, which lasted from the 1950s to the 1980s, was focused on rule-based systems that were manually designed by experts in specific domains. These systems were able to make decisions based on a set of predetermined rules and were used in a variety of applications, such as expert systems for medical diagnosis and financial forecasting.

The second phase of machine learning, which lasted from the late 1980s to the early 2000s, was characterized by the emergence of statistical learning methods such as decision trees, neural networks, and support vector machines. These methods allowed machines to learn from data without being explicitly programmed and were used in a wide range of applications, including speech recognition, image recognition, and natural language processing.

The third and current phase of machine learning is often referred to as "deep learning" and is based on the use of neural networks with many layers. These networks are able to learn from massive amounts of data and have been used to achieve breakthroughs in fields such as computer vision, speech recognition, and natural language processing. Deep learning has also enabled the development of autonomous vehicles, which rely on machine learning algorithms to make decisions in real-time.

Machine learning has revolutionized many industries, including finance, healthcare, and retail. It has enabled businesses to process and analyze large amounts of data quickly and accurately, leading to improved decision-making and greater efficiency. Machine learning has also contributed to the development of personalized medicine, where treatments are tailored to individual patients based on their genetic makeup and medical history.

**Popular Machine Learning Method**

**Supervised Learning**

Supervised learning is a type of machine learning in which an algorithm learns to make predictions by training on a dataset that contains labeled examples. These labeled examples consist of input data and their corresponding output data, which is also known as the target variable. During training, the algorithm learns to map the input data to the correct output by adjusting its internal parameters based on the labeled examples. Once the model is trained, it can be used to make predictions on new, unseen data.

Now, let's take a closer look at some of the most commonly used supervised learning algorithms:

**Linear Regression:**

Linear regression is a simple algorithm used for predicting a continuous output variable based on one or more input variables. It works by fitting a straight line to the data points and then using this line to make predictions. Linear regression is commonly used in finance, economics, and social sciences for making predictions based on historical data.

**Logistic Regression:**

Logistic regression is used for classification problems where the output is a categorical variable. It works by estimating the probability that a particular input belongs to each class, and then assigning the input to the class with the highest probability. Logistic regression is widely used in healthcare, marketing, and credit scoring.

**Decision Tree:**

A decision tree is a tree-like model that uses a set of rules to make predictions. It works by recursively partitioning the data into subsets based on the values of the input variables, and then using a simple rule to make a prediction for each subset. Decision trees are commonly used in finance, healthcare, and fraud detection.

**Random Forest:**

Random Forest is an ensemble learning algorithm that uses multiple decision trees to improve the accuracy of the model. It works by training a set of decision trees on randomly selected subsets of the data, and then averaging their predictions to make the final prediction. Random Forest is used in finance, healthcare, and customer segmentation.

**Support Vector Machine (SVM):**

Support Vector Machine is a popular algorithm used for classification problems. It works by finding a hyperplane that maximally separates the input data into different classes. SVM is used in image recognition, bioinformatics, and text classification.

**K-Nearest Neighbor (KNN):**

K-Nearest Neighbor is a non-parametric algorithm used for classification and regression problems. It works by finding the k nearest neighbors to a given input, and then assigning the input to the class or predicting the output based on the values of its nearest neighbors. KNN is used in pattern recognition, recommender systems, and anomaly detection.

**Neural Networks:**

Neural Networks are powerful algorithms used for solving complex problems such as image recognition, natural language processing, and speech recognition. They work by simulating the structure and function of the human brain, using layers of interconnected nodes that process and transform the input data.

**Naive Bayes:**

Naive Bayes is a probabilistic algorithm used for classification problems. It works by estimating the probability of each input belonging to each class, based on the frequency of each input feature in the training data. Naive Bayes is used in text classification, spam filtering, and sentiment analysis.

**Unsupervised Learning**

Unsupervised learning is a type of machine learning where an algorithm learns to identify patterns in the data without the need for labeled examples. Unlike supervised learning, there is no output variable to predict or guide the learning process. Unsupervised learning is a type of machine learning where the algorithm learns from unlabeled data without any explicit supervision. The goal of unsupervised learning is to discover patterns and relationships within the data without prior knowledge of the underlying structure. The main advantage of unsupervised learning is that it can identify hidden patterns and structures in the data that might not be apparent through traditional statistical analysis.

There are many algorithms in unsupervised learning, but we will focus on three popular ones: k-means clustering, principal component analysis (PCA), and autoencoders.

**K-means clustering:**

K-means clustering is a popular unsupervised learning algorithm used for clustering data. It is a simple and effective algorithm that partitions a dataset into k clusters, where k is a user-defined parameter. The algorithm iteratively assigns each data point to the closest cluster center and then updates the cluster centers based on the mean of the assigned data points. This process continues until the cluster centers no longer change or a maximum number of iterations is reached.

The k-means clustering algorithm is useful for identifying natural clusters in data, such as customer segments, market segments, or clusters of genes. One of the limitations of the k-means clustering is that it assumes that the clusters are spherical and of equal size. If the data is not well separated or the clusters are of different sizes, the algorithm may not work well.

**Principal component analysis (PCA):**

PCA is a dimensionality reduction technique used to reduce the number of variables in a dataset while retaining as much of the original variance as possible. The goal of PCA is to find a new set of orthogonal variables, called principal components, that explain the maximum amount of variance in the data.

The PCA algorithm works by finding the eigenvectors and eigenvalues of the covariance matrix of the data. The eigenvectors are the principal components, and the eigenvalues represent the amount of variance explained by each principal component. The first few principal components are selected to retain most of the variance in the data, while the remaining components are discarded.

PCA is useful for visualizing high-dimensional data and identifying the most important variables in the data. It can also be used to preprocess data before applying other machine learning algorithms. One of the limitations of PCA is that it assumes a linear relationship between variables, so it may not work well for nonlinear data.

**Autoencoders:**

Autoencoders are a type of neural network used for unsupervised learning. They consist of an encoder network that maps the input data to a compressed representation and a decoder network that reconstructs the input data from the compressed representation. The goal of autoencoders is to learn a compact representation of the input data that preserves its essential features.

The autoencoder algorithm works by minimizing the reconstruction error between the input data and the reconstructed data. This is achieved by training the neural network on the input data without any labels. Once the network is trained, it can be used to generate new data that is similar to the input data.

**Convolutional Neural Network:**

A Convolutional Neural Network (CNN) is a deep learning model used primarily for image processing and computer vision tasks. CNNs are designed to recognize visual patterns directly from pixel images with minimal preprocessing.

The CNN architecture consists of convolutional layers, pooling layers, and fully connected layers. Convolutional layers perform a mathematical operation called convolution that filters the input image and extracts features. The pooling layers downsample the output of the convolutional layers, reducing the number of parameters and computation required. The fully connected layers connect all the neurons in one layer to every neuron in the next layer, leading to a high number of parameters.

CNNs have demonstrated outstanding performance in image classification, object detection, segmentation, and other computer vision tasks. Additionally, CNNs have been applied to non-image processing tasks, such as natural language processing and speech recognition.

**CHAPTER FOUR**

**METHODOLOGY**



**PRINCIPAL COMPONENT ANALYSIS:**

Principal Component Analysis (PCA) is a widely used method for face recognition. This technique involves extracting the most significant features from a large dataset of faces by reducing the dimensionality of the data. PCA is based on the idea of transforming the data into a new coordinate system where the features are uncorrelated, and the variability in the data is captured by a few principal components. In this column, we will discuss the theory and equations behind PCA for face recognition.

First, let's define some terms. A face image can be represented as a matrix of pixel values. For example, an image with dimensions 100 x 100 pixels can be represented as a vector of length 10,000. A dataset of face images consists of a collection of these vectors. Each image in the dataset is considered a point in a high-dimensional space, where the dimensionality is the number of pixels in the image. The goal of PCA is to find a lower-dimensional space that captures the most significant features of the data.

PCA starts by centering the data around the mean face image. This is done by subtracting the mean vector from each image in the dataset. The mean vector is calculated as follows:

where N is the number of images in the dataset, and xi is the vector representing the ith image.

Next, PCA finds the principal components of the data by computing the eigenvectors of the covariance matrix. The covariance matrix measures the linear relationship between the pixels in the images. It is calculated as follows:

where T denotes the transpose operation. The covariance matrix C is a square matrix with dimensions equal to the number of pixels in the image.

The eigenvectors of the covariance matrix are computed as follows:

where v is an eigenvector of C, and λ is the corresponding eigenvalue. The eigenvectors are sorted in descending order of their eigenvalues. The eigenvectors with the highest eigenvalues capture the most significant variations in the data. These eigenvectors are known as the principal components.

To reduce the dimensionality of the data, we can project the images onto the principal components. This is done by multiplying the mean-centered image vector xi by the eigenvector matrix V:

where yi is the vector of principal component scores for the ith image. The vector yi is of length k, where k is the number of principal components used for the projection. By choosing a smaller value of k, we can reduce the dimensionality of the data.

To recognize a face image, we first project it onto the principal components using the equation above. We then compare the principal component scores of the test image with those of the images in the dataset. The closest match is the image with the smallest Euclidean distance to the test image in the principal component space.

PCA is a powerful technique for face recognition that involves finding the most significant features of a large dataset of face images. This is done by projecting the images onto a lower-dimensional space that captures the most significant variations in the data. By choosing a smaller value of k, we can reduce the dimensionality of the data, making the recognition process faster and more efficient.

Facial expressions. The flowchart of our Eigenfaces algorithm is mentioned-

input image

Convert the image into face vector

Flatteing of the vector

Split of the vector obtain into train and test set

Standardise both train and test set vector

Fit the model on the test set and compare the accuracy

Train on train set classification Algorithm (KNN, SVM)

Extract PCA components which capture 95% of the variance

Apply PCA on both train and test set

**Support Vector Machine (SVM)**

SVMs are a type of supervised learning algorithm that can be used for classification and regression tasks. The basic idea behind SVMs is to find a hyperplane that separates the data into different classes with the widest possible margin. In the case of face recognition, the SVM tries to find a hyperplane that can separate the face images of different individuals.

The mathematical formulation of SVMs is based on the concept of Structural Risk Minimization (SRM), which seeks to minimize the empirical risk while maximizing the geometric margin. The empirical risk is the number of misclassifications in the training set, while the geometric margin is the distance between the hyperplane and the closest points from each class.

Suppose we have a set of face images {(x1, y1), (x2, y2), ..., (xn, yn)}, where xi is the i-th face image, and yi is the corresponding label indicating the identity of the person in the image. In the case of face recognition, the labels are discrete values that correspond to the identity of the person.

The SVM tries to find a hyperplane that separates the face images of different individuals. The hyperplane is defined by the equation-

where w is a p-dimensional vector and b is a scalar. For linearly separable data, the hyperplane is selected to differentiate the datasets, such that:

And

if yi = -1

The distance between the hyperplane and the closest points from each class is given by:

To find the optimal hyperplane, we need to minimize ||w|| subject to the constraints:

, for all i = 1, 2, ..., n

This is a constrained optimization problem that can be solved using Lagrange multipliers. The Lagrange multipliers introduce a set of dual variables αi, which are used to obtain the optimal hyperplane. The optimization problem can be written as:

To find the optimal solution, we need to solve the saddle point equation:

subject to: αi ≥ 0, for all i = 1, 2, ..., n

The solution to this problem gives us the optimal hyperplane, which can be used to classify new face images.

**K Nearest Neighbors**

KNN (K Nearest Neighbors) is a non-parametric machine learning algorithm that has been widely used for facial recognition. It is a practical approach that is based on identifying facial features such as eyes, nose, eyebrows, mouth, and ears from a source image to recognize a face.

The main advantage of KNN for facial recognition is its robustness. This is achieved by normalizing the size and orientation of the face. KNN achieves this by using distance metrics such as the city-block distance, Euclidean distance, or cosine distance to find the k-closest data points to the query image.

The distance metric used by KNN is defined by the following equations:

City-Block Distance:

Euclidean Distance:

Cosine Distance:

where xi and yi are the features of the query image and the k-closest data points, respectively.

Once the k-closest data points are identified, KNN analyzes them to determine a common class label among the set. The most common class label is then assigned to the query image.

KNN classifier is an extension of the simple nearest neighbor algorithm, which employs non-parametric decision-making based on the distance of query image features from other image features. The algorithm is among the simplest machine learning algorithms and provides faster execution time than other classifier algorithms.

The KNN classifier's value depends on the number of samples used and their topological distribution. Larger values of k reduce noise in classification, but a good k value is selected through heuristic techniques.

In the presence of noise or inconsistent feature scales, evolutionary algorithms can be employed to reduce noise and optimize feature selection.

KNN is a practical and robust approach to facial recognition, and its effectiveness has been demonstrated by its use in various real-world applications. With the availability of large datasets and the continued advancement of machine learning techniques, KNN is expected to remain an important tool for facial recognition and other pattern recognition tasks.

**KNN Model:**

Take K Nearest Neighbour

End

Initialization define K

compute distance between input and trainig data

Sort Distance

Yes

**Fig:** Block Diagram Representation for KNN Model