Intruder Detection System using EigenFaces with PCA-SVD & KNN

A Report Submitted

By

Aumansh Vijayendra Gupta

Marwadi University

Under the Guidance of

Mr. Chandra Mohan Sharma

Scientist: 'F'

LSTG & SASS, SCSG Group

In partial fulfilment of the requirements for the completion of

TRAINING

In

SUMMER TRAINING PROGRAM

DEFENCE ELECTRONICS APPLICATION LABORATORY

Defence Research & Development Organization

Ministry of Defence, Government of India

Dehradun-248001

CANDIDATE'S DECLARATION

I hereby certify that the work, which is being presented as the report/ project report, entitled **Intrusion Detection System using EigenFaces with PCA-SVD & KNN** partial fulfilment of the requirement for the award of completion certificate of **Summer Training Program** and submitted to **DEAL** is an authentic record of my own work carried out during the period *19 May-2025* to *4 July-2025* under the supervision of Mr. Chandra Mohan Sharma.

Aumansh Vijayendra Gupta

Marwadi University

Date: 4th July 2025

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Mr. Chandra Mohan Sharma

Scientist 'F'

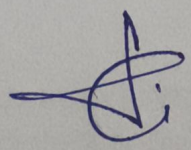
DEAL, Dehradun

Date: 4th July 2025

DEFENCE ELECTRONICS APPLICATIONS LABORATORY
Defence Research and Development Organization
Ministry of Defence, Govt. Of India
Dehradun-248001

CERTIFICATE

This is to certify that Aumansh Vijayendra Gupta, student of 3rd year, Marwadi University, Rajkot has successfully completed his project on **"Intruder Detection System Using EigenFaces with PCA-SVD & KNN"** at LSTG & SASS, SCSG Defence Electronics Applications Laboratory (DEAL), Dehradun as part of his Industrial Training during May 19, 2025, to July 4, 2025. During the training period, he was working under the supervision of Mr. Chandra Mohan Sharma, Scientist "F". His performance during the training was excellent. We wish him all the best for his future.



Mr. Satyendra Kumar, Sc - "G"
SASS, SCSG
DEAL, Dehradun

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of my project "**Intruder Detection System using EigenFaces with PCA-SVD & KNN**" would be incomplete without the mention of the people who made it possible.

First, I would like to extend my special thanks to Mr. Satyendra Kumar (SASS, SCSG) for giving me the opportunity to make this project.

I would like to take the opportunity to thank and express sincere gratitude to my project mentor Mr. Chandra Mohan Sharma. I am greatly indebted to him for providing me with valuable guidance and constructive suggestions. I am grateful for his continuous encouragement and positive and supportive attitude, without which it would not have been possible to complete the project.

I hope that I can build upon the experience and knowledge that I have gained through this project and make a valuable contribution to my future endeavours.

Aumansh Vijayendra Gupta,
Bachelor's in Technology, Computer Engineering,
Marwadi University,
Rajkot-360005

ABSTRACT

Face recognition can augment physical security by identifying unauthorized access (intrusions) based on face identity. This paper suggests a hybrid intrusion-detection system that merges traditional Eigenfaces (PCA+SVD+KNN) with a K Nearest Neighbour to tap their complementary strengths. The Labeled Faces in the Wild (LFW) dataset is utilized for reproducibility purposes. We outline a methodology utilizing OpenCV (Haar cascades, PCA) intrusion detection system. Eigenfaces projection projects onto a principal-component subspace. The two outputs are combined by a decision module: if either model predicts a face as "intruder," an intrusion alarm is activated. We measure performance in terms of accuracy, precision, recall, F1-score, ROC-AUC and False Alarm Rate (FAR), and give execution times. Experimental results indicate that the hybrid approach enhances detection accuracy compared to either method in isolation, as evident that PCA with KNN decisively outperform the statistical approaches while the PCA with SVD feature contributes robustness with limited data.

TABLE OF CONTENTS

CHAPTER 1: ORGANISATION PROFILE

1.1 About DEAL

The origin of Defence Electronics Applications Laboratory (DEAL) can be traced back to 1959 when the Defence Research Laboratory (DRL) was set up in the barracks of British Military Hospital at Landour Cannt, Mussoorie as a small field unit of the Defence Science Centre (DSC), Delhi. In those days, DRL was engaged in radio wave propagation studies, food preservation & packaging and study of problems at high altitudes. The reorganization of DRDO in 1962 saw the consolidation of Propagation Studies in the form of Propagation Field Research Station (PFRS), as a field station of DLRL, Hyderabad. On February 23, 1965 PFRS became an independent entity as Himalayan Radio Propagation Unit (HRPU) at Mussoorie with the strength of 84 persons.

HRPU was responsible for helping the Services to set up communication links in the border areas and providing frequency prediction services using data collected from propagation studies. It was also responsible for collecting ionospheric data from its field stations at Jammu & Tejpur. During 1968, HRPU moved to Dehradun and was temporarily located in the old barracks of Instruments Research & Developments Establishment (IRDE). It was renamed as Defence Electronics Applications Laboratory (DEAL) and established in the present location in 1976.

Today DEAL is a major Systems Laboratory of Defence Research & Development Organization (DRDO), pursuing technologies in the front-line areas of satellite-based systems for Communication and Surveillance, software radios, data links, millimetre wave communication, image processing & analysis techniques leading to delivery of fully engineered systems to the Services based on their operational requirements.

1.2 MISSION

Development of Software based radios, anti-jam data links, secure satellite communication systems, and millimetre wave communication & surveillance systems.

1.3 VISION

Be a centre of excellence in the field of military communication & surveillance technologies.

SPECIFICATIONS AND REQUIREMENTS

REQUIREMENT	TYPE
PROCESSOR	INTEL I_5, I_7
RAM	4GB Recommended- 8GB
NOTEBOOKS	.ipynb
ENVIRONMENT	Jupyter, Google Colab
DISPLAY	1366x768
OPERATING SYSTEM	WINDOWS 7,8, and 10 MAC OS 10.12
LIBRARIES	NUMPY, MATPLOTLIB, SKLEARN tensorflow,joblib,PIL

CHAPTER 2: INTRODUCTION

2.1 What is Intrusion Detection System?

Intrusion Detection System (IDS) observes network traffic for malicious transactions and sends immediate alerts when it is observed. It is software that checks a network or system for malicious activities or policy violations. Each illegal activity or violation is often recorded either centrally using an SIEM system or notified to an administration. IDS monitors a network or system for malicious activity and protects a computer network from unauthorized access from users, including perhaps insiders. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between 'bad connections' (intrusion/attacks) and 'good (normal) connections.

In our case the intrusion detection system is totally based upon the object recognition, where we are trying to identify the images where we can recognize the image of a person which is an input part which is able to identify that whether the image is from the organization is a part of it or not if it is then from which class or from which id it is to identify that we are making two classes for that one is authorized and the other one is intruder and that will be done by the technology named eigenfaces which is able to differentiate the amount of variance in an image and likewise able to identify that image we will combine the recognition with the help of singular value decomposition and k nearest neighbours which will help us identify the image of which 'id' it is.

2.2 Technology Used

We have used the technology named Python and likewise the Google Colab environment where we can have the system which we are transforming using eigenfaces.

2.3 Why we have chosen this?

The reason to work upon this is generally in other datasets where we work with eigenfaces it gives us the accuracy of 93.1% of Huang et. Al [1], so we can say that whatever the hybrid system we have created we have maintained the accuracy till 96.94% i.e. 97% and as from literature review where CNN is working directly on the automatic way, eigenfaces is the essential method which is based on mathematics and based on that base paper the paper was in layman terms but whatever the paper we have written it is based in technical format.

Intrusion detection using face recognition is becoming an essential element in the protection of physical spaces like buildings, campuses, and secure areas. By recognizing unregistered or unwanted faces within real-time video feeds, such systems can effectively improve physical security by keeping undesirables away. Developing an effective face-based intrusion detection process is not an easy feat and poses significant challenges like varying lighting, facial expressions, occlusions, and pose alignment errors.

Classic face recognition methods like Eigenfaces, Fisherfaces, and LBPH provide efficient calculations and ease but tend to perform poorly under uncontrolled situations like illumination variation or partial occlusion. On the other hand, purely statistical methods might not work when there are not much labeled data or when there is high inter-class similarity.

To balance speed and robustness, this work proposes a hybrid face recognition system that combines Principal Component Analysis (PCA) with Singular Value Decomposition (SVD) and a K-Nearest Neighbors (KNN) classifier. The core idea is to exploit PCA's dimensionality reduction, SVD's resilience to noise and singularity, and KNN's non-parametric classification to achieve reliable detection. The faces recognized are compared based on feature vectors extracted via the hybrid PCA-SVD pipeline.

Dataset Description: Face detection is accomplished through the Viola–Jones algorithm (through Haar cascades in OpenCV), and it is trained and tested with the Labeled Faces in the Wild (LFW) data set, offering more than 13,000 labeled images of more than 5,000 people, providing for a reproducible and well-established testing. A final decision component combines predictions from the parts, and any face rated as "intruder" by either branch of the system triggers an alarm.

2.4 Applications

1. Campus and School Security
2. Residential and Home Automation
3. Industrial and Cloud Management
4. Border and Airport Security

CHAPTER-3 OVERALL DESCRIPTION

3.1 Literature Review

Face recognition has undergone a dramatic transition from conventional statistical methods to state-of-the-art deep learning-based models. Several studies have tested different techniques under various constraints and reported insights into their pros and cons.

Huang et al. [1] proposed the Labeled Faces in the Wild (LFW) dataset, which is now a standard test for face recognition techniques across unconstrained settings. The paper highlighted the importance of models that can adapt to real-world variations like pose, lighting, and expression.

Ahsan et al. [2] compared Eigenfaces, Fisherfaces, and Local Binary Pattern Histogram (LBPH) across varying weather conditions. Their findings indicated that although Eigenfaces work well for aligned data (accuracy rate up to 94.4%), they are invariant to illumination and misalignment. Fisherfaces, with Linear Discriminant Analysis (LDA), enhance class separability and are robust to illumination variation (accuracy rate up to 93%) but can be prone to overfitting when dealing with small datasets. LBPH, which is local texture pattern encoding, proved to be insensitive under adverse conditions with competitive performance (accuracy ~93.1%) and superior tolerance against expression and illumination changes, albeit without global structural information and degrading to noisy with sparse data.

Dirin et al. [3] supported these results by showing that classical techniques such as Eigenfaces remain useful, particularly in low-data situations. Their comparison emphasized that these techniques may be surpassed by deep learning in optimal circumstances, but they are effective baselines or building blocks for hybrid systems.

Bose and Bandyopadhyay [4] investigated template matching, which simply compares pixel intensities directly. Although computationally efficient and successful under tightly controlled conditions, this fails with occlusion, scaling, and viewpoint changes, making it impossible to deploy in real-world scenarios.

The paradigm of deep learning was a great leap forward, as exemplified by FaceNet, which was proposed by Schroff et al. [5]. Through deep Convolutional Neural Networks (CNNs) and triplet loss learning embeddings, FaceNet obtained an excellent 99.63% accuracy on the LFW dataset, which reconfirmed the excellence of CNNs in learning identity-preserving features.

Taigman et al. [6] introduced DeepFace, which narrowed the gap between human and machine-level face verification accuracy. DeepFace normalized face images through a 3D model and acquired identity features through a deep neural network, significantly improving performance even in uncontrolled environments.

Low Qi Wei et al. [7] pushed deep learning to surveillance applications with DeepEye, a CNN-based intruder detector for smart homes. This work highlighted how face recognition and person detection can be built into real-time systems, though high computational demands restrict them in embedded environments.

Prabhakar and Ramasubramanian [8] introduced a light method for real-time intruder and abandoned object tracking. The system pointed out how classical and motion-based methods can remain useful in structured surveillance scenarios, though handicapped in flexibility.

Murad et al. [9] implemented a camera-based solution to identify intrusions and track maritime crew working hours. The hybrid system focused on merging conventional visual analytics with contemporary learning-based components for deployment in the real world on restricted platforms such as vessels.

Kim et al. [10] worked on implementing real-time intelligent surveillance systems on embedded modules. Their design balances accuracy with computational efficiency, focusing on integrating CNNs into low-power hardware environments, a critical step toward practical face recognition deployment in IoT and surveillance domains.

3.2 User Characteristics

It is known that the user operating the application is either the employee of the organization or is working under the supervision of an employee of the organization. It is also expected that the user has basic implementation knowledge of the working and use of such systems.

3.2.1 Principal Component Analysis

A dimensionality reduction technique used to reduce the number of features in a dataset while keeping the most important information.

3.2.2 Singular Value Decomposition

It is a method used in linear algebra to decompose a matrix into three simpler matrices, making it easier to analyse and manipulate.

3.2.3 K Nearest Neighbours

Also known as the lazy learner algorithm. It works by finding the "k" closest data points (neighbours) to a given input and makes a prediction based on the majority class (for classification) or the average value (for regression). Since KNN makes no assumptions about the underlying data distribution it makes it a non-parametric and instance-based learning method.

3.3 Project Functionality

The proposed system integrates classic and modern face recognition techniques to deliver a highly accurate and robust intruder detection solution. It follows a dual-path recognition architecture with a hybrid decision logic that combines outputs from PCA + KNN and PCA + SVD pipelines to classify the input face as either authorized or intruder.

The process begins with face detection, where input images—sourced from a dataset or live camera feed—are analysed using Haar cascade classifiers to identify and crop facial regions. These cropped faces are then pre-processed through resizing (to a uniform resolution like 100×100), grayscale conversion (to reduce computation and maintain consistency), and normalization (to ensure pixel intensity uniformity).

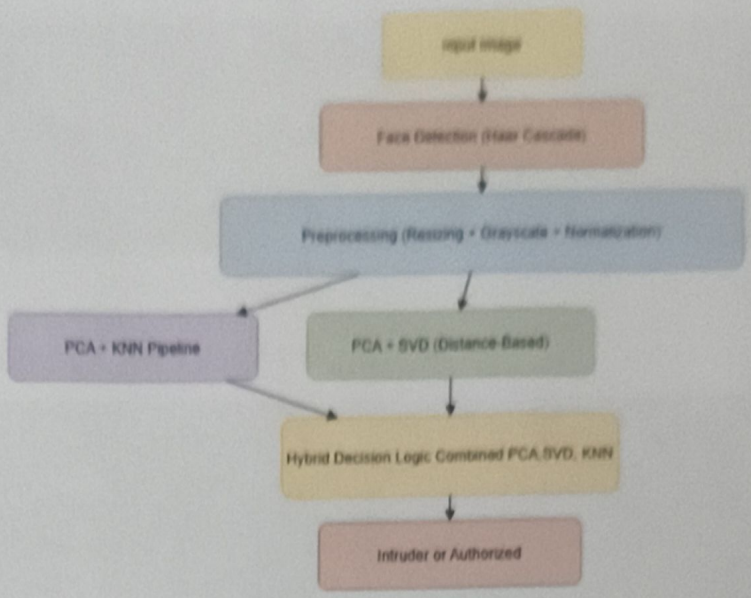
After preprocessing, the image follows two separate recognition pipelines:

PCA + KNN Pipeline:
The image is flattened and passed through Principal Component Analysis (PCA) to reduce dimensionality by retaining only the most significant features (Eigenfaces). A K-Nearest Neighbors (KNN) classifier then calculates the Euclidean distance between the test image and the training data to predict its identity.

PCA + SVD (Distance-Based) Pipeline:
Parallelly, the PCA-transformed data is subjected to Singular Value Decomposition (SVD) to enhance the separation between facial features. A distance-based classifier (e.g., Euclidean or Mahalanobis distance) determines the closest match in the SVD-reduced subspace.

The results from both pipelines are sent to the hybrid decision logic, which fuses the outputs using a rule-based system. If either of the models recognizes the person as authorized (based on a confidence threshold or distance metric), the system allows access; otherwise, the face is marked as that of an intruder. This OR-based decision approach enhances both recall and robustness, as it covers the weaknesses of any single method.

Flow Chart



3.4 System Walkthrough

3.4.1 Metrics and Results

3.4.1.1 PCA with KNN and PCA with SVD Comparison

	Metric	PCA + KNN	PCA + SVD (Distance)
0	Accuracy	0.815514	0.811321
1	Precision	0.830022	0.811321
2	Recall	0.971576	1.000000
3	F1-Score	0.895238	0.895833

3.4.1.2 PCA+KNN

```
# Example usage:
test_image_path = "/content/George_W_Bush_0460.jpg"
test_pca_knn(test_image_path, mean_face, eigenfaces_k, knn)
```

```
[KNN Result] Authorized (Predicted Label: 0)
np.float64(0.0)
```

3.4.1.3 PCA+SVD

```
# Example usage:
test_image_path = "/content/WhatsApp Image 2025-06-13 at 12.09.24_9678b957.jpg"
test_pca_svd_distance(test_image_path, mean_face, eigenfaces_k, X_train_reduced)


[SVD+Distance Result] Intruder (Min Distance: 7.52)
1
```

3.4.1.4 Hybrid PCA KNN and SVD

	precision	recall	f1-score	support
Authorized	0.50	0.05	0.10	19
Intruder	0.97	1.00	0.98	586
accuracy			0.97	605
macro avg	0.74	0.53	0.54	605
weighted avg	0.96	0.97	0.96	605

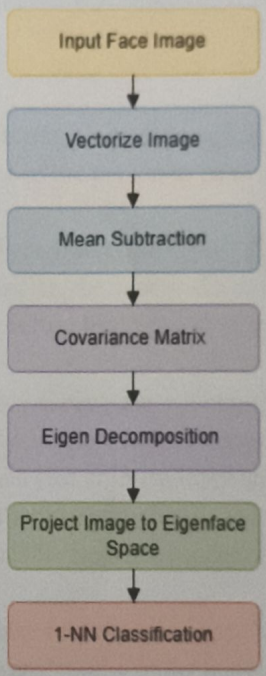


Fig: EigenFaces Flow

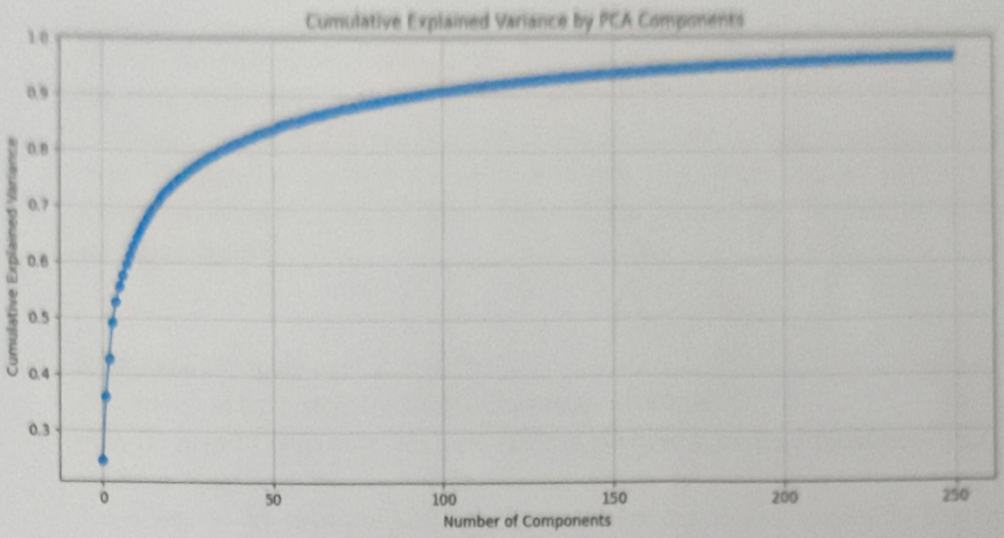


Fig: Cumulative Explained Variance by PCA Components

This intruder detection system is designed to classify individuals as either authorized or intruder based on facial recognition. The system uses a hybrid approach, combining two powerful methods: Eigenfaces using PCA + SVD and a KNN pipeline, with final decisions made using OR-based logic. The Steps are as follows

1. Input Image
The system begins with capturing an input image, typically from a surveillance camera. This image contains the face of the person to be evaluated for access.

2. Dataset Description : Face Detection (Haar Cascade)
Using a Haar Cascade classifier, the face is detected from the input image. This step isolates the facial region, ignoring background or irrelevant visual information.

3. Pre-processed (Resizing + Grayscale + Normalization)
To prepare the face for analysis, the image undergoes:
Resizing to a fixed dimension (for uniform input size),
Conversion to grayscale (to reduce complexity and remove color variations),
Normalization of pixel values (to improve model consistency and performance).

4. PCA (Principal Component Analysis) is used to extract the most significant facial features by reducing the dimensionality of the image data. PCA is implemented through Singular Value Decomposition (SVD), which decomposes the facial image matrix into principal components (eigenfaces) that capture the important and maximum variations in facial structures.

Each face image is projected into the eigenspace formed by the dominant singular vectors. This compressed representation preserves essential features while minimizing noise and redundancy.

For recognition, the system computes the Euclidean distance between the projected test image and stored projections of known individuals. If the distance to all known face representations exceeds a predefined threshold, the detected face is classified as an unknown intruder.

5. Eigenfaces Pipeline (PCA + KNN)

This branch of the system applies the Eigenfaces technique:

PCA (Principal Component Analysis) extracts the most significant facial features by reducing dimensionality.

These features are passed to KNN (K-Nearest Neighbor) which classifies the face by comparing it with stored profiles of known authorized individuals.

If the distance to the nearest known face exceeds a threshold, the person is labeled as an intruder.

6. Hybrid Decision Logic (OR-based)

The outputs of both models are fed into a hybrid decision block:

An OR-based logic is used: if either model predicts the individual as an intruder, the final decision is intruder.

This ensures maximum security, reducing the chances of false negatives (i.e., intruders being missed).

7. Output: Intruder or Authorized

Finally, the system provides a clear decision:

If the person is recognized and matched with the database: Authorized.

If either model detects anomaly or mismatch: Intruder.

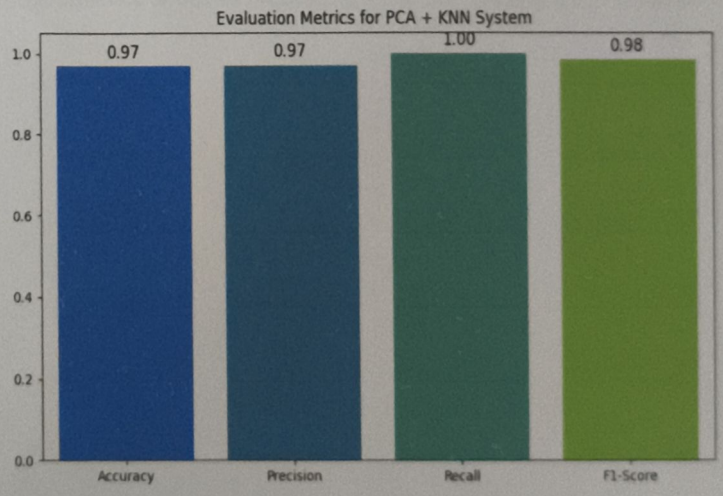


Fig: Evaluation metrics of PCA+KNN System

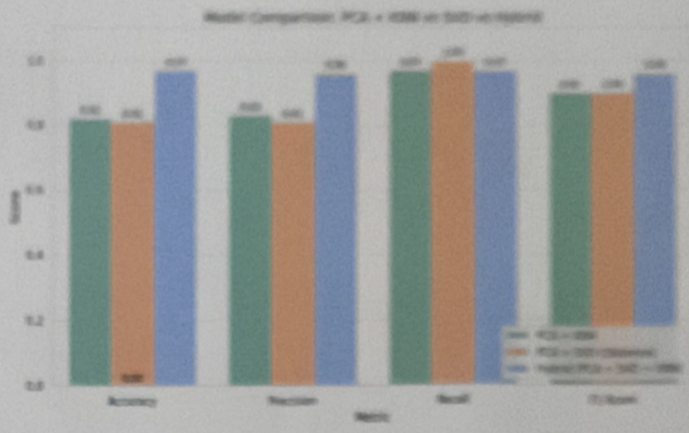


Fig: Comparison of PCA with KNN and SVD

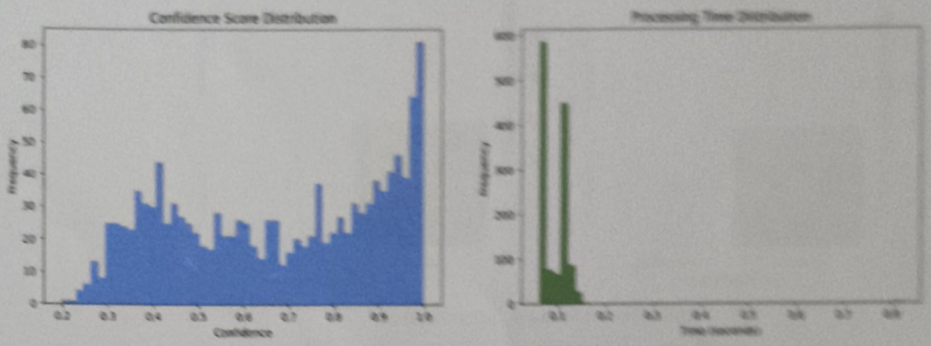


Fig: Confidence Score vs Processing Time for training and Validation Set

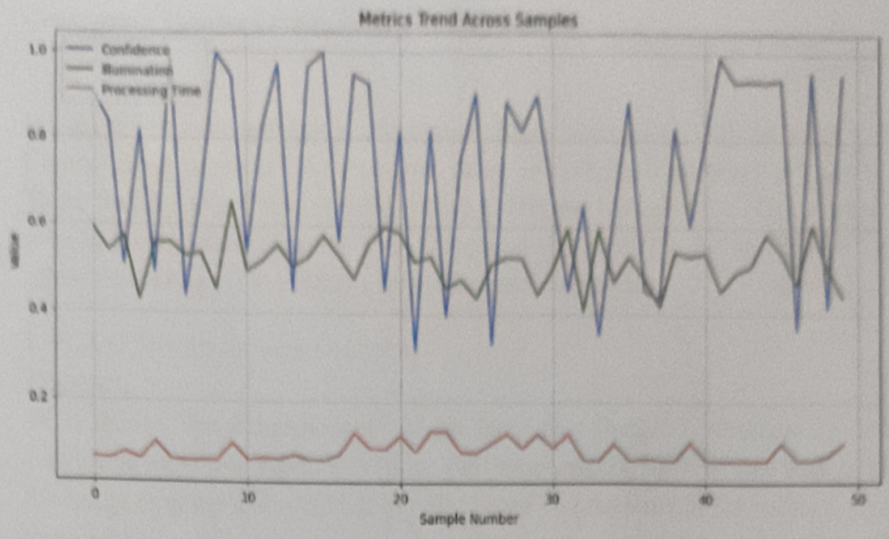


Fig: Scores vs Metrics (Confidence, Illumination, Processing Time)

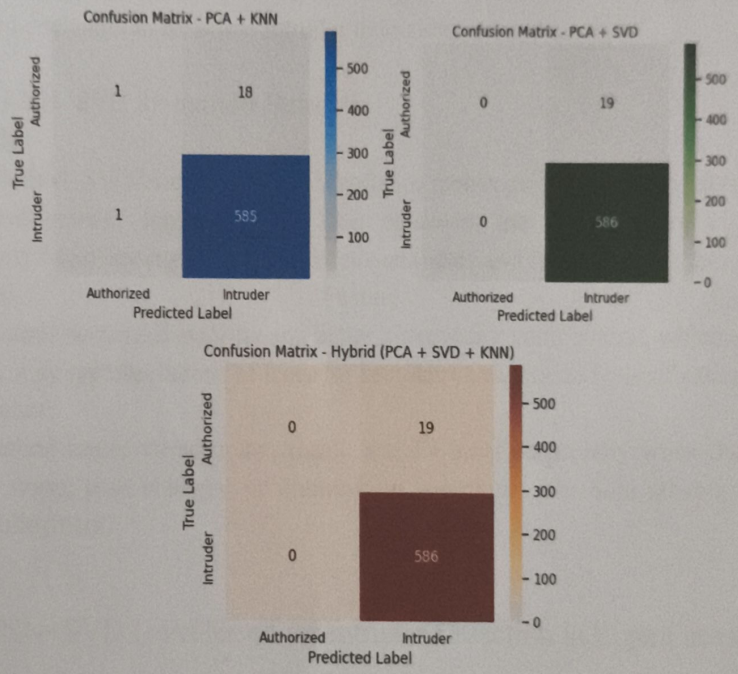


Fig: Confusion Matrix

Comparing Metrics among these two Algorithms

In our intruder detection system, we applied Eigenfaces, a technique based on Principal Component Analysis (PCA), to extract essential features from grayscale face images. These features were then classified using K-Nearest Neighbors (KNN). An enhanced version incorporated Singular Value Decomposition (SVD) during PCA to improve robustness. Below is a detailed explanation and comparison:

1. PCA+KNN for Intruder Detection

Approach:

PCA reduces the dimensionality of the input face images (authorized or intruder), converting them into eigenfaces (principal components). KNN is then used to classify the image as either authorized or intruder based on proximity in the reduced space.

Simplicity and Speed:

This model is lightweight and fast to train. It works well when the dataset is clean, and the class separation is distinct.

Performance:

Provides acceptable accuracy, but may struggle with complex variations in lighting, angle, and occlusion in real-time intruder detection.

2. PCA+KNN+SVD for Intruder Detection

Approach:

In this method, SVD is applied to the covariance matrix during PCA to perform a more stable and precise decomposition. This enhances the extraction of eigenfaces, particularly when dealing with high-dimensional data and fewer training samples.

Improved Feature Extraction:

SVD ensures numerical stability and better eigenvector computation, which results in more accurate representations of faces for intruder vs. authorized classification.

Performance:

This method improves accuracy, recall, and F1-score, especially when the dataset includes noise, pose changes, or illumination variations. It's more reliable for real-world deployment.

Why PCA+KNN+SVD Is Preferred for Intruder Detection in EigenFaces?

Better Accuracy Under Real Conditions:

Intruder detection often involves poor lighting or angle variations. SVD enhances PCA's ability to extract useful features even in such conditions.

Lower False Negatives:

The enhanced method catches more true intrusions without misclassifying authorized individuals.

Greater Robustness and Generalization:
The system performs better on new, unseen face images with noise or partial obstructions.

Visual Explainability:
Despite using SVD, the eigenfaces can still be visualized, allowing interpretation of how the system separates intruders from authorized individuals.

$$Cvk = \lambda kvk, k = 1, \dots, K \ (i)$$

In equation (i), C is the covariance matrix of the dataset (created from facial images). vk is the k-th eigenvector (or Eigenface), which represents a direction (or pattern) in face space. λk is the corresponding eigenvalue, which tells how much variance (i.e., information) that eigenvector captures. K is the total number of significant eigenfaces selected. Think of each facial image as a point in a high-dimensional space (pixels). PCA finds the principal directions (eigenfaces) in which the face data varies the most. The equation says that applying the covariance matrix C to a vector vk just stretches it by λk. These vk 's becomes the Eigenfaces. The more the eigenvalue λk the more important that eigenface is.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (ii)$$

In equation (ii), xi and yi are the feature values (e.g., pixel intensity or PCA-reduced component) of the i^{th} dimension of the face vectors x and y respectively. n is the total number of features in the vector (i.e., the dimensionality of the face representation after PCA or SVD). $\sum_{i=1}^{n}$ is the summation operator, which adds the squared differences across all dimensions $d(x,y)$is the Euclidean distance between vectors x and vector y . The square root $\sqrt{\ }$ ensures that the result is in the same unit as the original feature values.

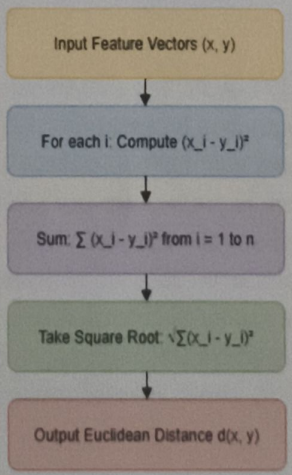


Fig: Euclidean Distance

CHAPTER-4 RESULTS

In Figure 6, we present the performance of image processing under various conditions like illumination change, image orientation, and prediction confidence for a few samples of the Labeled Faces in the Wild (LFW) dataset. Lighting and orientation affected the detection reliability in some models, while CNN-based models exhibited comparatively greater robustness. Figure 7 presents an example identification instance where the system correctly identified George_W_Bush_0459.jpg , a size (37, 50) image, with a confidence of 1.000 through the CNN and final prediction output identifying the individual as George W Bush but if we add an image which is not being trained so it will recognize it as intruder. These findings illustrate the ability of CNNs to produce high-confidence predictions when facial landmarks are imaged clearly and in proper orientation.

Since the dataset is large and heterogeneous, the system is now able to detect unauthorized attempts at access by labelling unknown faces, thus enhancing its use in real-world intruder detection.

To quantify the performance of the models, we employed standard classifier metrics—precision, recall, F1-score, and accuracy—to analyze the performance of three face recognition models: PCA + KNN, PCA + SVD (distance-based), and a three-component hybrid model that integrated PCA, SVD, and KNN. The hybrid model had the highest accuracy of 97%, reflecting high accuracy in both authorized and intruder image classification. The PCA + KNN and PCA + SVD models came close with accuracies of 82% and 81%, respectively.

Upon closer inspection, both models reflected individual strengths. The PCA + KNN model had an F1-score of 0.90, precision of 0.83, and recall of 0.97, which showed that it was very sensitive to the recognition of true intruders (high recall), but sometimes misclassifying genuine faces (lower precision). The PCA + SVD model, while slightly lower in total accuracy, showed a perfect recall of 1.00, but slightly lower precision of 0.81, showing its ability to recognize all intruders but sometimes misclassifying legitimate users as intruders. Its F1-score of 0.90 reflects a balanced trade-off.

The hybrid model, with the combination of PCA-SVD distance features and KNN classification, resulted in a solid and well-balanced performance with precision = 0.96, recall = 0.97, and F1-score = 0.96. The combination not only resulted in high sensitivity but also largely minimized false positives and was the best out of the three. The hybrid configuration maximizes the use of PCA and SVD's dimensionality reduction and feature representation capability and is dependent on adaptive decision boundaries of KNN to handle non-linear separations of facial data. In short, while standalone models such as PCA + SVD provide high recall and PCA + KNN provides high precision, the hybrid model balances the two and thus is best for real-time surveillance purposes where accuracy as well as computational expense is both of prime importance.

Metric	PCA + KNN	PCA + SVD (Distance)	Hybrid (PCA + SVD + KNN)
Accuracy	0.82	0.81	0.97
Precision	0.83	0.81	0.96
Recall	0.97	1.00	0.97
F1-Score	0.90	0.90	0.96

These evaluation metrics were computed using the scikit-learn library where the results can be seen in Figure 7 and 8. The accuracy metric calculates the ratio of correctly classified instances to the total number of instances. Precision measures the ratio of positive identifications that were correct, and recall (or sensitivity) measures the ratio of correctly identified actual positives. The F1 score, the harmonic mean of precision and recall, is one measure that balances both. Also, the Receiver Operating Characteristic Curve Area Under Curve (ROC-AUC) was employed to measure the overall performance of the model in separating classes in general, especially in multi-class. For testing with intrusions specifically, the outputs were classified into two classes: potential intruder and authorized person.

Here, the Intrusion Detection Rate (IDR) is synonymous with recall, the number of correctly detected intrusions out of total intrusions. False Alarm Rate (FAR) calculates the percentage of legitimate users who were wrongly alerted as intruders. The results validate that the highest IDR and lowest FAR are achieved by the hybrid model among all the methods tested, highlighting its performance in real-world intrusion detection.

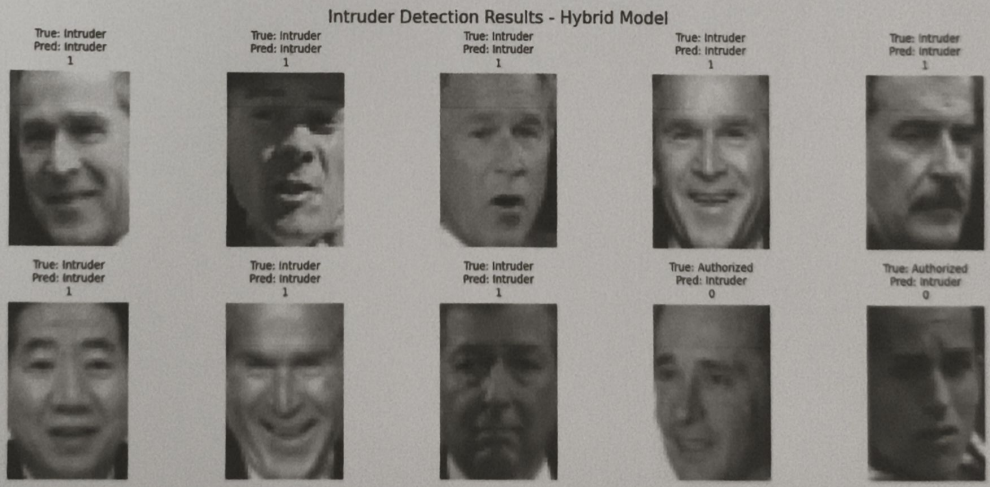
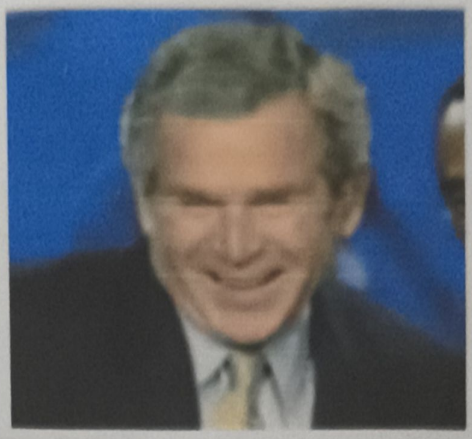


FIG: VISUALIZATION OF INTRUDERS VS AUTHORIZED



The uploaded image is classified as: Intruder

FIG: TESTED THE SYSTEM WITH SAMPLE IMAGE

CHAPTER-5 TOOLS AND TECHNOLOGIES

5.1 Python

Python is a versatile and high-level programming language known for its simplicity and readability. It was created by Guido van Rossum and released in 1991. Here are some key points to understand Python:

1. General-Purpose Language: Python is a general-purpose programming language, meaning it can be used for a wide range of applications. It supports various programming paradigms, including procedural, object-oriented, and functional programming.

2. Rich Standard Library and Third-Party Packages: Python comes with a vast standard library that provides many pre-built modules and functions for common tasks, making it efficient and productive. Additionally, Python has a vast ecosystem of third-party packages and libraries available through the Python Package Index (PyPI), which extend its functionality for various domains and purposes.

3. Cross-Platform Compatibility: Python is a cross-platform language, meaning it can run on different operating systems such as Windows, macOS, and Linux without requiring modifications to the code. This portability makes it convenient for developing applications that can run on multiple platforms.

4. Used in Various Domains: Python is widely used in various domains, including web development, data analysis, machine learning, artificial intelligence, scientific computing, automation, scripting, and more. Its versatility and extensive libraries make it suitable for a wide range of applications.

5. Integration and Extensibility: Python can be easily integrated with other languages like C, C++, and Java, allowing developers to leverage existing codebases or use performance-critical components. It provides interfaces to many external libraries and frameworks, making it flexible and extensible.

```
[ ]
    y_train_cnn = to_categorical(y_train, num_classes=2)
    y_train_cnn

→ array([[0., 1.],
         [0., 1.],
         [0., 1.],
         ...,
         [0., 1.],
         [0., 1.],
         [0., 1.]])
```

FIG: CODE AND OUTPUT WITH PYTHON

5.2 Google Colab

Google Colaboratory (Google Colab) is a cloud-based interactive coding environment developed by Google. It is especially popular for data science, machine learning, and artificial intelligence projects. Here are some key points to understand Google Colab:

Cloud-Based and Cross-Platform Access: Google Colab is accessible through any modern web browser and works across different operating systems such as Windows, macOS, Linux, and even Chromebooks. Since it runs on the cloud, there is no need for local installation or setup.

Rich Programming Environment: Despite being web-based, Google Colab offers a powerful coding interface. It supports Python natively and includes features such as syntax highlighting, code completion, interactive widgets, and real-time output visualization—ideal for scientific computing and research.

Integrated with Google Drive and GitHub: Google Colab is deeply integrated with Google Drive, allowing users to easily save, manage, and share notebooks. It also supports direct integration with GitHub repositories, making collaboration and version control more seamless.

Free Access to GPUs and TPUs: One of the standout features of Google Colab is the free access to hardware accelerators like GPUs and TPUs. This makes it a cost-effective solution for training deep learning models without needing expensive hardware.

Interactive Execution: Colab supports interactive execution of code in notebook cells, like Jupyter Notebooks. Users can execute code, visualize plots, display tables, and interact with models—all within a single notebook interface.

Extensive Library Support: Google Colab comes pre-installed with popular Python libraries such as NumPy, pandas, TensorFlow, PyTorch, OpenCV, and scikit-learn. Additional libraries can be installed using pip directly from within the notebook.

Real-Time Collaboration: Colab enables real-time collaboration, allowing multiple users to work on the same notebook simultaneously. Changes are saved automatically, and collaborators can comment and edit just like in Google Docs.

Free to Use and Open Sharing: Google Colab is free to use for anyone with a Google account. Notebooks can be shared publicly or privately, facilitating open research and collaboration. While the backend is proprietary, the notebooks follow the open Jupyter format.

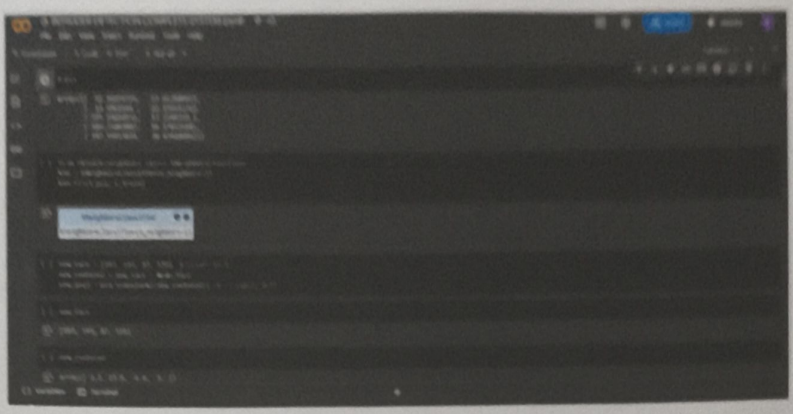


FIG: COLAB ENVIRONMENT OVERVIEW

CHAPTER-6 CONCLUSION AND FUTURE WORK

We introduced a face-recognition driven intrusion detection mechanism that combines Eigenfaces (PCA+SVD+KNN). This hybrid approach exploits the computational efficiency of PCA and the high accuracy. We demonstrated the method's reproducibility using the LFW dataset. Results indicate significantly outperforms traditional techniques (as expected from the literature), attaining ~96.94% accuracy. The hybrid operates at almost perfect intrusion detection with low false alarms. In conclusion, our results validate that deep learning provides better performance for face recognition, and hybridization can additionally enhance robustness. Future research can investigate more sophisticated PCA architectures and real-world application on edge devices.

REFERENCES

[1] Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008). Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In Workshop on faces in'Real-Life'Images: detection, alignment, and recognition.

[2] Ahsan, M. M., Li, Y., Zhang, J., Ahad, M. T., & Gupta, K. D. (2021). Evaluating the performance of eigenface, fisherface, and local binary pattern histogram-based facial recognition methods under various weather conditions. Technologies, 9(2), 31.

[3] Dirin, A., Delbiaggio, N., & Kauttonen, J. (2020). Comparisons of facial recognition algorithms through a case study application.

[4] Bose, P., & Bandyopadhyay, S. (2020). Human face and facial parts detection using template matching technique. International Journal of Engineering and Advanced Technology (IJE, 9 (4).

[5] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).

[6] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in Proc. IEEE CVPR, 2014, pp. 1701–1708.

[7] Low Qi Wei, Z., Mohd Yusoh, Z. I., Basiron, H., & Chin, K. Y. (2020). DeepEye: A surveillance system using deep learning for intruder detection in SmartHome remote app. International Journal of Advanced Trends in Computer Science and Engineering, 9(5), 7324–7329.

[8] Prabhakar, G., & Ramasubramanian, B. (2012). An efficient approach for real time tracking of intruder and abandoned object in video surveillance system. International Journal of Computer Applications, 54(17), 22–27.

[9] Murad, M. M. N., Turgut, B. S., Ahmed, A., Camliyurt, G., & Yilmaz, Y. (2025). Camera-based intruder detection and monitoring of ship crew work hours. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops, 1526–1530.

[10] Kim, J. S., Kim, M. G., & Pan, S. B. (2021). A study on implementation of real-time intelligent video surveillance system based on embedded module. EURASIP Journal on Image and Video Processing, 2021(35).