

WEB-BASED FACIAL RECOGNITION AND AUTHENTICATION

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Engineering degree in Computer Science and
Engineering with specialization in Artificial intelligence

By

VINAY MANNAM (Reg.No - 40731059)

MSIDDHU MAHANANDA (Reg.No - 40731096)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A++” by NAAC | Category I University by UGC |
Approved by AICTE JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VINAY MANNAM (40731059)** who carried out the project entitled "**Web based facial authentication**" under my supervision from November 2023 to April 2024.

B. Balaigayathri 13/04/24
Internal Guide

Ms. B. Balasaigayathri M.E.,

B. Balaigayathri
Head of the Department

Dr. L. Lakshmanan M.E., Ph.D.,

Submitted for **Viva voce Examination** held on

HEAD OF THE DEPARTMENT
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai - 600 119.

L. Lakshmanan 13/04/24
Internal Examiner

Dr. L. Lakshmanan 14/04/2024
External Examiner

DECLARATION

I, Vinay Mannam (Reg.No - 40731059), hereby declare that the Project Phase-2 Report entitled **"WEB BASED FACIAL RECOGNITION AND AUTHENTICATION"** done by me under the guidance of **Ms. B. Balasai Gayathri, M.E.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 13/04/2024

PLACE: Chennai



SIGNATURE OF CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D., Dean**, School of Computing, **Dr. L.Lakshmanan M.E., Ph.D., and Dr.S.Vigneshwari M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms. B. Balasai Gayathri M.E.** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of our project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Web-based facial recognition and authentication have gained significant attention in recent years due to the growing need for secure and convenient identity verification methods. This technology utilizes advanced algorithms to capture, analyze, and compare facial features for authentication purposes. The primary goal is to accurately verify the identity of individuals accessing online services or digital platforms, mitigating the risk of unauthorized access and fraudulent activities. The web-based approach allows users to conveniently access and utilize the system from any location, using devices such as smartphones, tablets, or computers with internet connectivity. The process involves capturing the user's facial image through the device's camera, then extracting and encrypting facial features to ensure privacy and security. The encrypted data is transmitted securely to a cloud-based server for processing, where it is compared to a stored database of registered users. The matching algorithm evaluates the similarity between the captured image and the stored templates, providing a confidence score for authentication.

Keywords: facial recognition, authentication, web-based, identity verification, algorithms, secure, convenient, online services, digital platforms, unauthorized access, fraudulent activities, smartphones, tablets, computers, internet connectivity, camera, facial image, encrypting, privacy, security, cloud-based server, processing, matching algorithm, database, registered users, confidence score. Online information systems currently heavily rely on the username and password traditional method for protecting information and controlling access. With the advancement in biometric technology and popularity of fields like AI and Machine Learning, biometric security is becoming increasingly popular because of the usability advantage. This paper reports how machine learning based face recognition can be integrated into a web-based system as a method of authentication to reap the benefits of improved usability. This paper includes a comparison of combinations of detection and classification algorithms with FaceNet for face recognition. The results show that a combination of MTCNN for detection, Facenet for generating embeddings, and Linear SVC for classification out performs other combinations with a 95% accuracy. The resulting classifier is integrated into the web-based system and used for authenticating users.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF TABLE	ix
1	INTRODUCTION	1
2	LITERATURE SURVEY	4
	2.1 Inferences from Literature Survey	4
	2.2 Open Problems in Existing System	7
3	REQUIREMENTS ANALYSIS	8
	3.1 Feasibility Studies of the Project	8
	3.2 Software Requirements Specification	8
4	DESCRIPTION OF PROPOSED SYSTEM	12
	4.1 Selected methodology or process model	12
	4.2 Architecture of the proposed system	13
	4.3 Description of software for implementation	14
	4.4 Project management plan	15
	4.5 Financial report	18
5	IMPLEMENTATION DETAILS	19
7	RESULTS AND DISCUSSIONS	20
8	CONCLUSION	23
	REFERENCES	24

APPENDIX

A	SOURCE CODE	25
B	SCREENSHOTS	33
C	RESEARCH PAPER	37

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.2.1	Artificial neural network	10
3.2.2	Convolutional neural network	11
4.1	Architectureof the proposed system	13
4.2	Architectureof the existing system	14
6.1	Results oflinear svc	21
6.2	Login screen	22
6.3	During facialrecognition	22
6.4	Dashboard	22
B.1	Image bit	33
B.2	Image processing	33
B.3	Django server code	34
B.4	Facial training	34
B.5	Website cart	35
B.6	Bootstrapjavascript	35
B.7	For training	36

LIST OF TABLES

TABLE NO	NAME	PAGE NO.
1	Comparison of face detection algorithms	20
2	Classification algorithm metrics	20
3	Datasets comparison	21

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
CNN	Convolutional Neural Network
IDE	Integrated development environment
NMS	Non-Maximum Suppression
MTCNN	Multi-Task cascaded convolutional neural networks

CHAPTER 1

INTRODUCTION

Information and Communication Technology usage has witnessed rapid growth in the past decade all around the world. A bigger percentage of the population has laptops, personal computers, and smart phones making it easy to access the internet and thus changing lives of millions of people. All web-based systems that have users and store personal information about the users require a mechanism to keep track of their users' information. Most commonly every user of the system is assigned an instance in the database that represents them (their identity). To protect the user identity, an authentication and authorization mechanism is implemented to control access to certain information. The most common method in web-based systems is authentication using passwords. Users regularly provide a combination of their username and password through a form to access a remote account. Passwords have well-known disadvantages in both usability and security. This leads to promotion of biometric-based authentication. The primary motivation of biometric authentication is usability: users are not required to remember the passwords, there is nothing for them to carry, biometric systems are generally easy to use, and scalable in terms of the burden exerted onto the users. Biometric technology can be used to control the risk of sharing, forgetting, losing, and embezzlement of passwords.

Biometric Authentication

Biometric authentication is the automatic authentication that identifies a person by analyzing their physiological and/or behavior features. In the past two decades, both physiological and behavior characteristics have been used in biometric authentication systems. Physiological features are attributed to the person's body shape such as the face, palm prints, DNA, etc. Behavioral characteristics are attributed to the person's behavior: typing speed, voice, and articulation

Artificial Intelligence

Artificial Intelligence is impacting the world with its ability to achieve the never thought tasks, the accurate resource knowledge used can highly affect the execution. In recent years we all have witnessed an exceptional growth in the improvement of state-of-the-art accuracy for multiple domains such as text classification, speech recognition, image processing and natural language processing. The challenges faced by hard-coded systems convey that artificial intelligence systems require an ability to extract and

acquire knowledge from raw data supplied to them. The idea of systems learning from data, identifying patterns and making decisions with little human intervention is known as machine learning. Real world AI applications face a difficulty caused by the fact that the factors of variation influence every unit of data that can be observed. For example, individual pixels in a blue object image may be very close to those of a black object at night. The shape of the object depends on the angle of observation. Most of the applications require us to extract the most important factors and discard the ones we do not care about. Deep learning, a subset of machine learning implements this by introducing factors represented in terms of other and simpler factors. Deep learning allows the computer to build complex problems out of simpler problems utilizing a hierarchy of artificial neural networks to carry out machine learning processes. Artificial neural networks are inspired and built like the biological neural network of a human brain. It consists of neuron nodes connected together. Traditional programs develop analysis with data in a linear way, while the hierarchical style of deep learning systems gives machines the ability to process data with in nonlinear way.

Face Recognition

In the present time, face recognition is a popular and important technology integrated in many applications such as payment, door unlock, video monitoring systems, etc. A face recognition system is a technology that can identify and verify people from digital images and footage. The state-of-the-art face recognition approaches for web authentication include Luxand and SmileID from ElectronicsID. These are both provided as commercial products. To recognize faces (human faces), images are taken using a digital camera. The first step is to perform face detection, this is performed as a preprocessing step to locate the face in an image. Faces are generally unique depending on the environment the images were taken: illumination, brightness, facial expressions, and the person's face orientation. The next step is the face recognition: a set of features are computed utilizing a special algorithm. The set of features generated is classified to represent each person's face uniquely. The response from the classifier is the owner of the facial features. A number of face recognition studies are carried out but these disclose only the pipeline representation stage. This was started years ago by the mathematical background. In any case, from the academic perspective, researchers from the Carnegie Mellon and Oxford universities have shared both network structures and pretrained models. From the commercial perspective, companies (Google and Facebook) share only the network structures.

Lucky for us there is an encouraging open-source community that has emerged who train these models from scratch and share them. There are many face recognition algorithms in the academic field, part of the research results are relatively mature. The state-of-the-art face recognition models include VGGFace from the University of Oxford, DeepFace from Facebook, OpenFace from the Carnegie Mellon University, and FaceNet from Google. A study that compares the four state-of-the-art recognition models was conducted during the development of the lightFace framework (A python framework that provides facerecognition support for each of these models). It showed that Google's facenet was the best model among the four.

CHAPTER 2

LITERATURE SURVEY

2.1 INFERENCES FROM LITREATURE SURVEY

1. Master Face Attacks on Face Recognition Systems(2022): Huy H. Nguyen; Sebastien Marcel; Junichi Yamagishi; Isao Echizen

Face authentication is now widely used, especially on mobile devices, rather than authentication using a personal identification number or an unlock pattern, due to its convenience. It has thus become a tempting target for attackers using a presentation attack. Traditional presentation attacks use facial images or videos of the victim. Previous work has proven the existence of master faces, i.e., faces that match multiple enrolled templates in face recognition systems, and their existence extends the ability of presentation attacks. In this paper, we report an extensive study on latent variable evolution (LVE), a method commonly used to generate master faces. An LVE algorithm was run under various scenarios and with more than one database and/or face recognition system to identify the properties of master faces and to clarify under which conditions strong master faces can be generated. On the basis of analysis, we hypothesize that master faces originate in dense areas in the embedding spaces of face recognition systems. Last but not least, simulated presentation attacks using generated master faces generally preserved the false matching ability of their original digital forms, thus demonstrating that the existence of master faces poses an actual threat.

2. A Privacy-Preserving Edge Computation-Based Face Verification System for User Authentication(2019): Xiang Wang; Heyu Xue; Xuefeng Liu; Qingqi Pei

The face recognition has become a common means of identity authentication because of the advantages of uniqueness, non-invasive and not easy to be stolen. The outsourcing of face recognition to the service provider is a typical manner nowadays. However, it raises vital concerns about the privacy of outsourcing server due to the sensitivity of face data. Therefore, a frame of identity authentication based on the technology of privacy- preserving face recognition is presented in this paper. The convolutional neural network is used for face feature extraction. To overcome the issue of privacy leaked, a method of secure nearest neighbor that can compute the cosine similarity over encrypted feature vectors is proposed. What's more, the edge

computing is introduced in our frame to increase the authentication efficiency by removing some operations from the cloud to the edge of the Internet. Moreover, we also propose a secret sharing homomorphism technology which is used for distributed computing to improve the fault-tolerance of our identity authentication system. The experimental results show that the proposed schemes are secure and effective.

3. A New Deep Neural Architecture Search Pipeline for Face Recognition(2020):

Ning Zhu; Zekuan Yu; Caixia Kou

With the widespread popularity of electronic devices, the emergence of biometric technology has brought significant convenience to user authentication compared with the traditional password and mode unlocking. Among many biological characteristics, the face is a universal and irreplaceable feature with simple detection methods and good recognition accuracy. Face recognition is one of the main functions of electronic equipment propaganda. The previous work in this field mainly focused on converting loss function in traditional deep convolution neural networks without changing the network structure. With the development of AutoML, neural architecture search (NAS) has shown remarkable performance in image classification tasks. In this paper, we first propose a new deep neural architecture search pipeline combined with NAS technology and reinforcement learning strategy into face recognition. We quote the framework of NAS, which trains the child and controller networks alternately. At the same time, we optimize NAS by incorporating evaluation latency into rewards of reinforcement learning and utilize the policy gradient algorithm to search the architecture automatically with the cross-entropy loss. The network architectures we searched out have achieved state-of-the-art accuracy in the large-scale face dataset, which achieved 98.77% top-1 in the MS-Celeb-1M dataset and 99.89% in LFW dataset with relatively small network size.

4. An Efficient Android-Based Multimodal Biometric Authentication System With Face and Voice(2020): Xinman Zhang; Dongxu Cheng; Pukun Jia; Yixuan Dai; Xuebin Xu

Multimodal biometric authentication method can conquer the defects of the unimodal biometric authentication technology. In this paper, we design and develop an efficient Android-based multimodal biometric authentication system with face and voice. Considering the hardware performance restriction of the smart terminal, including the random access memory (RAM), central processing unit (CPU) and graphics processor unit (GPU), etc., which cannot efficiently accomplish the tasks of storing and quickly processing the large amount of data, a face detection method is

introduced to efficiently discard the redundant background of the image and reduce the unnecessary information. Furthermore, an improved local binary pattern (LBP) coding method is presented to improve the robustness of the extracted face feature. We also improve the conventional endpoint detection technology, i.e. the voice activity detection (VAD) method, which can efficiently increase the detection accuracy of the voice mute and transition information and boost the voice matching effectiveness. To boost the authentication accuracy and effectiveness, we present an adaptive fusion strategy which organically integrates the merits of the face and voice biometrics simultaneously. The cross-validation experiments with public databases demonstrate encouraging authentication performances compared with some state-of-the-art methods. Extensive testing experiments on Android-based smart terminal show that the developed multimodal biometric authentication system achieves perfect authentication effect and can efficiently content the practical requirements.

5 Face Authentication Software Test Automation(2020): Debdeep Banerjee; Kevin Yu

The 3D face authentication has become a hot trend for researchers and developers in the recent years, due to its many advantages over the 2D face recognition feature. The 3D reconstruction of a human face using both near-infrared and depth sensors is a complex process on mobile phones. It commonly involves algorithms like face detection, face landmark detection, facial feature extraction, and depth information analysis. The 3D face authentication feature is critical for the user as per as security and also providing a convenient way to authenticate by the correct user. Therefore, the testing of 3D face authentication algorithms and applications in terms of functionality, performance, and stability is critical. However, the research on 3D face authentication application level validation and testing method is lacking in the field. Most testers are still validating the application manually. In this paper, we propose a robotic-arm-based test automation for testing the 3D face authentication feature on mobile phones. We programmed a 6 degree of freedom (6-DOF) robotic arm to perform 3D face authentication automated tests that were executed manually before. Our test automation also benchmarked the performance of an in-house developed 3D face authentication application and a 3rd party application which yielded promising latency and accuracy comparison results under different performance impacting test scenarios.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

Face recognition as a method of authentication in a web-based system has seen significant advancements, but there are still several open problems and challenges to address. Here are some of the key issues and open problems in the existing system:

- Protecting against spoofing attacks is a major challenge. Techniques such as liveness detection need improvement to ensure that the system cannot be easily tricked by photos or videos.
- Addressing privacy concerns is crucial, as biometric data (facial images) must be securely stored and handled to prevent misuse. Improving the overall accuracy of face recognition, especially in challenging scenarios, remains an open problem. This includes scenarios with partial occlusion or low-resolution images.
- Addressing these open problems in face recognition as a method of authentication in web-based systems will lead to more secure, reliable, and privacy-aware authentication methods, making them suitable for various applications, including online banking, e-commerce, healthcare, and more.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

Risk analysis is a critical step in any project, especially when implementing face recognition as a method of authentication in a web based system. Here's a risk analysis for such a project:

Data Breaches: Unauthorized access to the facial recognition database can result in a data breach, exposing sensitive user information.

Spoofing: Attackers may use photos or videos to impersonate users, bypassing the system's authentication. Implementing effective anti spoofing measures is crucial.

Data Privacy: Collecting and storing biometric data raises concerns about user privacy.

Failure to protect this data can result in legal and reputational risks.

Consent: Obtaining user consent for data collection and storage is essential to mitigate privacy risks. have contingency plans in place. Engaging with legal and compliance experts, addressing ethical considerations, and ensuring transparency with users can help mitigate some of the potential challenges associated with face recognition as an authentication method in a web-based system. Regular security audits and

To mitigate these risks, it's essential to conduct a thorough risk assessment, develop risk mitigation strategies, and assessments are also crucial to identify and address evolving risks.

3.2 SOFTWARE REQUIREMENTS SPECIFICATION

Hardware Requirements

RAM : 4GB or above 4GB

Processor of Frequency : 1.5GHz or above

Processor : Intel Pentium 4 or higher

Software Requirements

Operating System : Windows 11

Languages : Python

Tools used : Visual Studio Code

Python:

Python is a scripting language that is high-level, interpreted, interactive, and object-oriented. Python is intended to be extremely readable. It commonly employs English terms rather than punctuation, and it has fewer syntactical structures than other languages.

Python Features

- **Easy-to-learn** - Python has a small number of keywords, a basic structure, and a well-defined syntax. This enables the pupil to swiftly learn the language.
- **Easy-to-read** - Python code is more clearly defined and visible to the eyes. □ **Easy-to-maintain** - Python's source code is relatively simple to maintain.
- **A broad standard library** - The majority of Python's library is portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** - Python features an interactive mode that enables for interactive testing and debugging of code snippets.
- **Portable** - Python can operate on a broad range of hardware devices and has the same interface across them all.
- **Extendable** - The Python interpreter may be extended using low-level modules. These modules allow programmers to enhance or adapt their tools to make them more efficient.
- **Databases** - Python provides interfaces to all major commercial databases.
- **GUI Programming** - Python can construct and port GUI programmes to numerous system calls, libraries, and windows systems, including Windows MFC, Macintosh, and Unix's X Window system.
- **Scalable** - Python provides a better structure and support for large programs than shell scripting.

Artificial Neural Network:

Artificial Neural Network Tutorial provides basic and advanced concepts of ANNs. Our Artificial Neural Network tutorial is developed for beginners as well as professions. The term "Artificial neural network" refers to a biologically inspired subfield of artificial intelligence modeled after the brain. An Artificial neural network is usually a computational network based on biological neural networks that construct the structure of the human brain. Similar to a human brain has neurons interconnected to each other, artificial neural networks also have neurons that are linked to each other in various layers of the networks. These neurons are known as nodes.

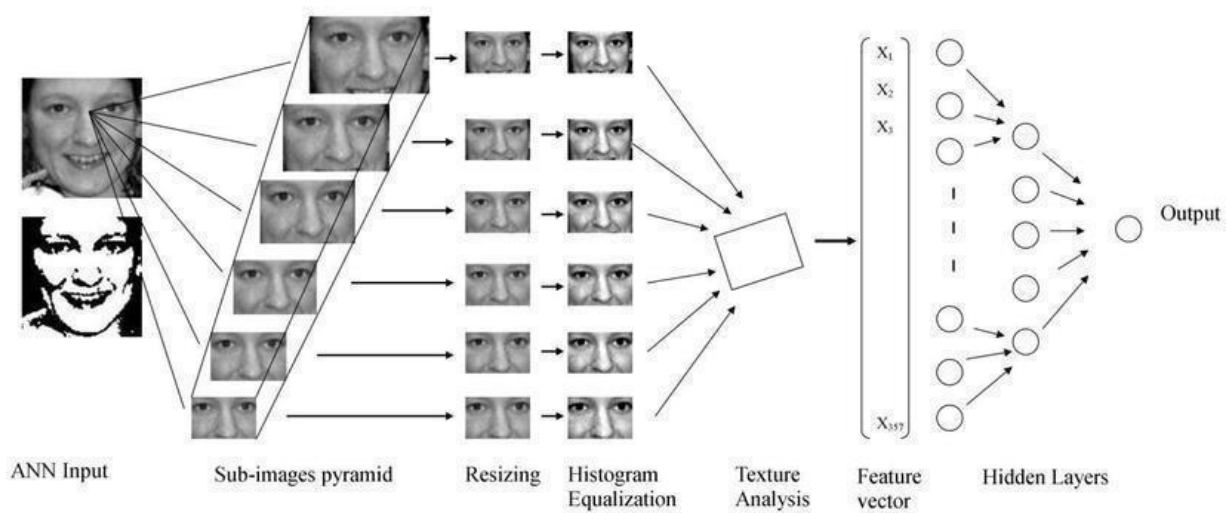


Fig.3.2.1. Artificial neural network

Convolutional Neural Network:

A convolutional neural network, or CNN, is a deep learning neural network sketched for processing structured arrays of data such as portrayals. CNN are very satisfactory at picking up on design in the input image, such as lines, gradients, circles, or even eyes and faces. This characteristic that makes convolutional neural network so robust for computer vision. CNN can run directly on a underdone image 17 and do not need any preprocessing. A convolutional neural network is a feed forward neural network, seldom with up to 20. The strength of a convolutional neural network comes from a particular kind of layer called the convolutional layer. CNN contains many convolutional layers assembled on top of each other, each one competent of recognizing more sophisticated shapes. With three or four convolutional layers it is viable to recognize handwritten digits and with 25 layers it is possible to differentiate human faces.

The agenda for this sphere is to activate machines to view the world as humans do, perceive it in a alike fashion and even use the knowledge for a multitude of duty such as image and video recognition, image inspection and classification, media recreation, recommendation systems, natural language processing, etc.

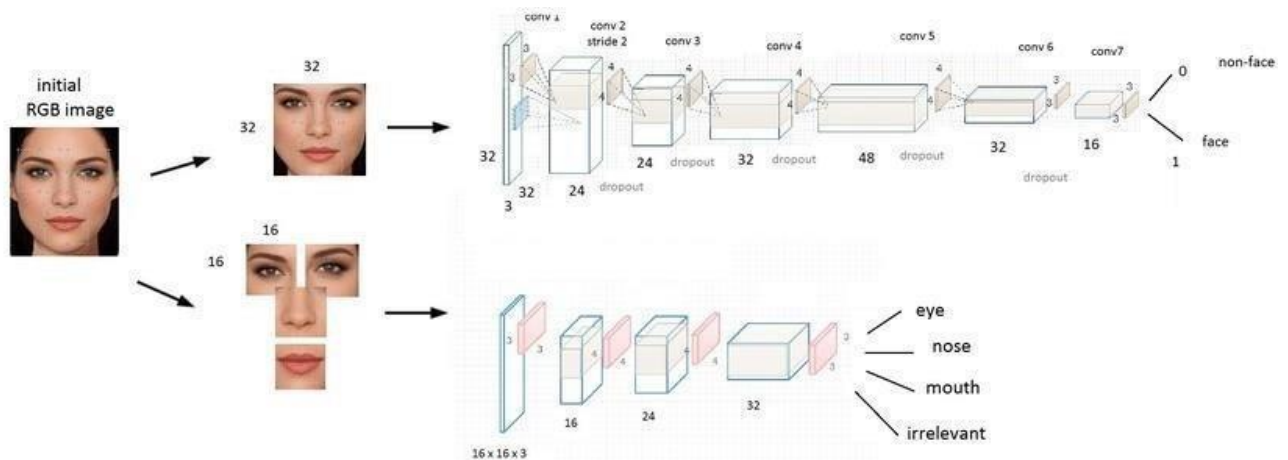


Fig.3.2.2 Convolutional neural network

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

Our Approach on face recognition consists of three stages, that is given an image, we perform face localization, face embeddings generation using Google's FaceNet pretrained model, and classification using the linearSVC classifier.

4.1 SELECTED METHODOLOGY OR PROCESS MODEL

Data Collection :

In a case study on 100 different people, a total of 1424 images from the volunteers. Captured using a 12MP dual-lens camera of the tecno camon 12 air smartphone, the images appeared in different orientations and lightings. The number 1424 is that which was obtained after performing data cleaning.

Face Detection :

For the case of face detection, two of the most commonly used ways of face detection were compared;

The Viola-Jones Haarcascade classifier: It is based on the Viola Jones Object Detection framework. This framework has a quite high (true positive rate) detection rate and a very low false positive rate making the algorithm a robust one that also processes the images quickly. The main objective is face detection not recognition: it distinguishes faces from non-faces which is the first (preprocessing) step recognition. The framework follows four main steps: Haar Feature Selection, Creating an Integral Image, Ada boost Training and the Cascading Classifiers.

The Multi-Task Cascaded Convolutional Neural Network: that is based on Neural Network Based Face Detection. A Neural Network inspired by human brain composed of simple artificial neurons also known as perceptrons are connected to each other in multiple layers.

R-Net further reduces the number of candidates (which further rejects a large number of false candidates) performs calibration with bounding box regression and employs Non-Maximum Suppression (NMS) to merge overlapping candidates. Layer 3: The Output Network (O-Net): Similar to the R-Net the O-Net aims to describe the face in detail outputting five facial landmarks' positions for the main face features like the eyes, nose and mouth. Face detection and localization was performed on the faces using the MTCNN algorithm. The localized face is resized into a 160×160 RGB image. A face dataset is created.

Face Recognition

A FaceNet model pre-trained on a VGGFace2 dataset consisting of 3.3M faces and 9000 classes in an inception ResNet v1 architecture was used. The pretrained model expects 160×160 RGB images. The model was converted into a keras compatible version. The converted model was used to generate embeddings for all images in the faces dataset. For each face image in the faces dataset, an array of 512 different weights uniquely representing the face was generated. A new dataset of face embeddings was prepared and compiled for classification purposes. The generated embeddings are used to create a new dataset consisting of face embeddings. Table I below shows information about the pretrained Facenet model.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

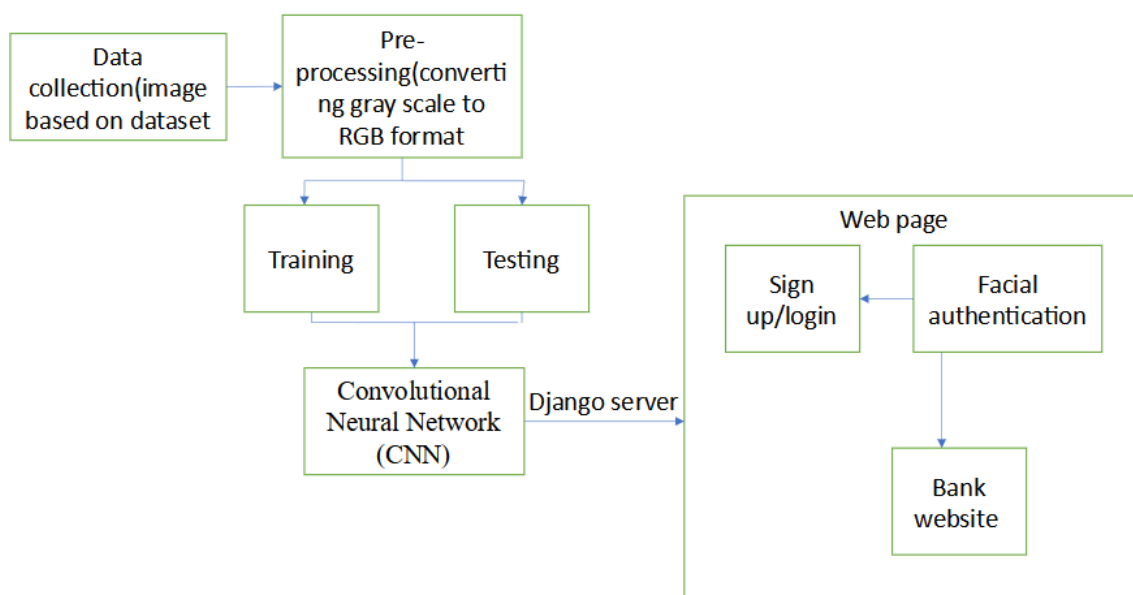


Fig.4.1 architecture of the proposed system

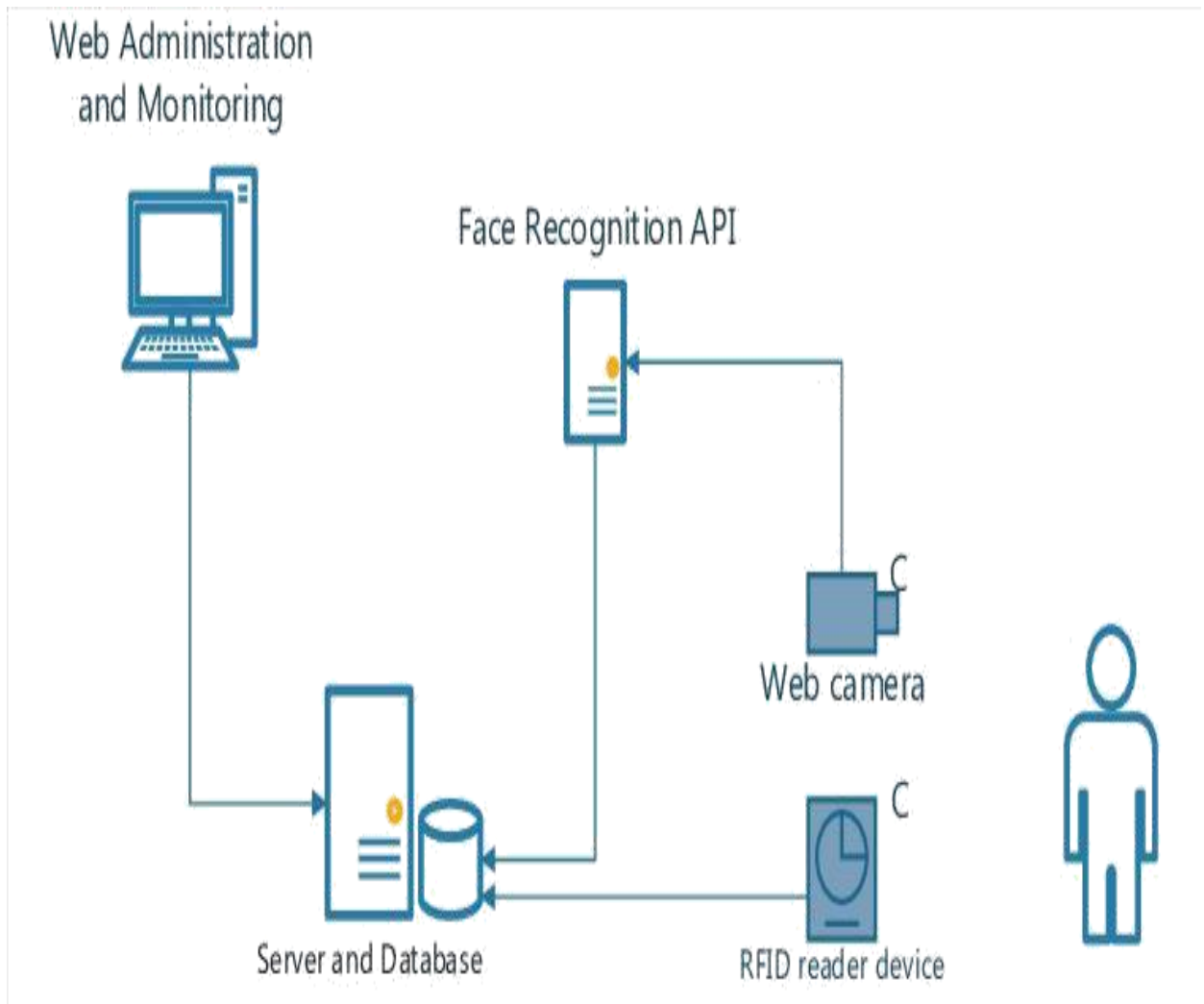


Fig.4.2 architecture of existing system

4.3 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

Software Description

- Python libraries like Scikit-learn, PyTorch, TensorFlow for building and training classification models. Provide ML frameworks.
- OpenCV, Pillow/PIL for image processing and feature extraction from image.
- Docker to containerize and deploy the ML model and web app components.

- ReactJS or Angular for building front-end to display image.
- Django for REST API server to interface ML model with front-end.
- MySQL database to store image, predicted labels and metrics.
- Git version control system for code collaboration and version management.

Testing Plan

- Unit testing of individual components - data processors, feature extractors, ML model APIs.
- Integration testing of end-to-end pipeline - preprocessing, feature extraction, model prediction, output storage.
- Validation testing for model accuracy on unlabeled test datasets with human-verified ground truth.
- Load testing the production system with different user loads, IMAGE volumes to gauge scalability.
- Security testing including checks for OWASP top 10 vulnerabilities in APIs, user authentication.
- User acceptance testing with teachers/evaluators providing feedback on usability.
- Performance testing with different cloud instance types to optimize cost/accuracy trade-offs.

4.4 PROJECT MANAGEMENT PLAN

A Project Management Plan (PMP) for implementing face recognition as a method of authentication in a web-based system is essential for ensuring a successful project.

To create a complete project on Face Recognition, we must work on 3 very distinct phases:

Face Detection and Data Gathering

Train the Recognizer

Face Recognition

Once we gather the data we train our model with this data to see how well it performs. We can do this by splitting the dataset into training and testing datasets. The ideal ratio is 80: 20. This step lets us train the model.

Now we build the recognizer. We feed the new data of the users to the model and assign id to sort them out. Now the model is built and after testing it out by using some examples we try to see its accuracy.

4.5 FINANCIAL REPORT ON ESTIMATED COSTING

API Usage Costs:

- API Licensing and Usage Fees: 1500 rupees
- Development and Integration of Custom APIs: 1300 rupees
- API Documentation and Training: 500 rupees
- Total API Usage Costs: 3300 rupees

Cloud Services Costs:

- Cloud Computing Services (e.g., AWS, Azure): 1500 rupees
- Data Storage and Backup: 800 rupees
- Cloud-Based Tools and Software: 800 rupees
- Total Cloud Services Costs: 3100 rupees

Literature Surveys Costs:

- Access to Academic Journals and Databases: 120 rupees
- Research Materials and Resources: 650 rupees

4.6 TRANSITION / SOFTWARE TO OPERATIONS PLAN

Software and Environment Compatibility:

Cross-Platform Compatibility: Ensure that any software developed (e.g., data analysis scripts or applications) is compatible with both Windows and Mac OS.

IDE Flexibility: Encourage users to choose from popular Integrated Development

Environments (IDEs) such as Jupiter Notebook, VScode, or PyCharm according to their preferences.

CHAPTER 5

IMPLEMENTATION DETAILS

Our Approach on face recognition consists of three stages, that is given an image, we perform face localization, face embeddings generation using Google's FaceNet pretrained model, and classification using the linearSVC classifier.

Data Collection : In a case study on 100 different people, a total of 1424 images from the volunteers. Captured using a 12MP dual-lens camera of the tecno camon 12 air smartphone, the images appeared in different orientations and lightings. The number 1424 is that which was obtained after performing data cleaning

Face Detection :

The Proposal Network (P-Net): This is a fully convolutional network which is used to obtain the candidate windows and their bounding box regression vectors

The Refine Network (R-Net): The output of the P-Net is input into the R-Net which is a Convolution Neural Network and hence denser than the P-Net layer. The R-Net further reduces the number of candidates which further rejects a large number of false candidates, performs calibration with bounding box regression and employs Non-Maximum Suppression to merge overlapping candidate.

The Output Network (O-Net): Similar to the R-Net the O-Net aims to describe the face in detail outputting five facial landmarks' positions for the main face features like the eyes, nose and mouth.

CHAPTER 6

RESULTS AND DISCUSSION

Face localization using the Viola-Jones Haarcascade algorithm produced results. It can be observed that the algorithm eliminated some of the vital facial features like the chin, which in turn affected the accuracy of recognition with a face dataset generated using this algorithm. The algorithm failed to localize faces despite existence in the image on several occasions.

The MTCNN algorithm, on the other hand, could perfectly detect a face whenever one existed in the image. The results of the MTCNN detection on a similar group of images. By observation, the MTCNN outperforms the Viola-Jones algorithm in localizing faces. When the two algorithms were used to generate identical datasets, which were further used to train the linear SVC model, the performance metrics in table II below were obtained. A 74% versus 98% accuracy on a similar dataset reflects how good a choice the MTCNN is compared to the Viola-Jones haarcascade.

COMPARISON OF THE FACE DETECTION ALGORITHMS METRICS

Algorithms	Accuracy	Precision
Haarcascade	0.74	0.74
MTCNN	0.98	0.98

Table -1

Table II shows the results obtained from the comparison of the two face detection algorithms we considered in our research basing on the accuracy and precision as metrics of performance. The performance metrics for classification were as indicated in the table

CLASSIFICATION ALGORITHMS METRICS

Classifier	Precision	Accuracy	Recall	F1 Score
LinearSVC	0.95	0.95	0.95	0.95
OneVsRest	0.94	0.94	0.94	0.93
Random Forest	0.91	0.91	0.91	0.91

Table-2

From the results discussed above, it was determined that the LinearSVC provided a better performance by 1% and 4% compared to the OneVsRest and Random Forest classifiers respectively. Thus the LinearSVC model was exported and utilized in the development of the facerecognition API. Figure 8 below is a normalized confusion matrix for 20 of the 100 classes (people) classified using the LinearSVC classifier. From this

figure, it can be determined that the class labeled 10 was wrongly classified as the class labeled 1 for half of its data items in the test set.

LinearSVC confusion matrix for 20 of the 100 classes

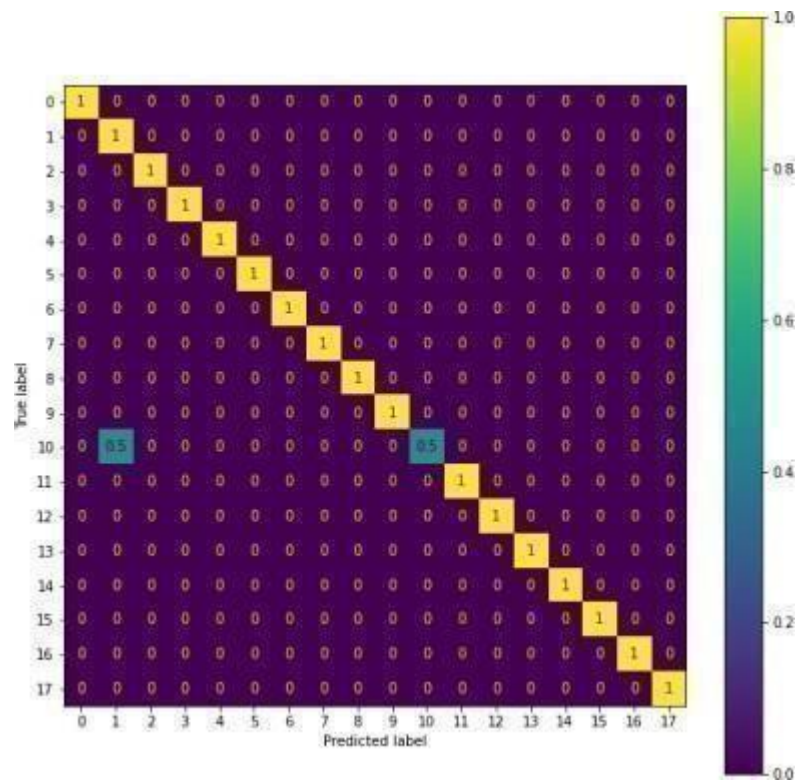


Fig.6.1 Results of linear svc

Table shows the results obtained from the comparison of the performance of the linearSVC when trained on a black faces dataset compared to a white faces dataset.

DATASETS COMPARISON

Dataset	Precision	Accuracy	Recall	F1 Score
Black people	0.94	0.94	0.93	0.92
White people	1.00	1.00	1.00	1.00

Table-3

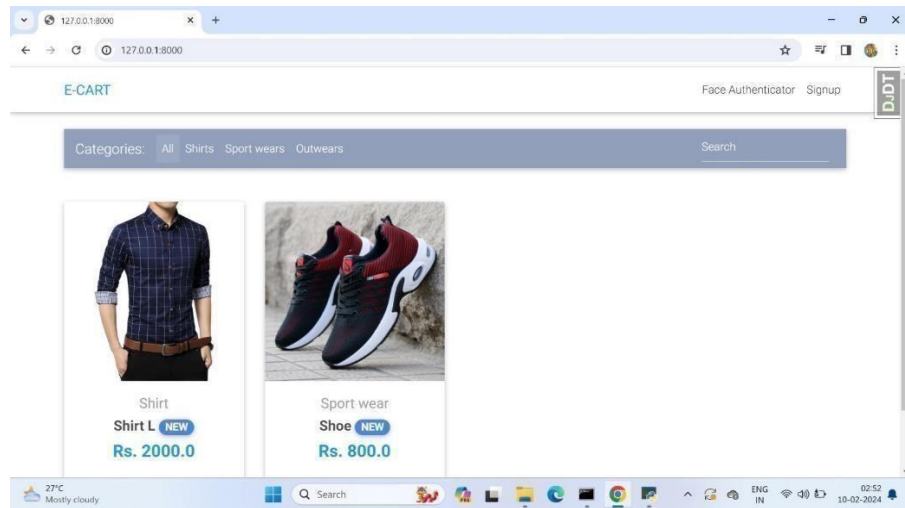


Fig.6.2 login screen

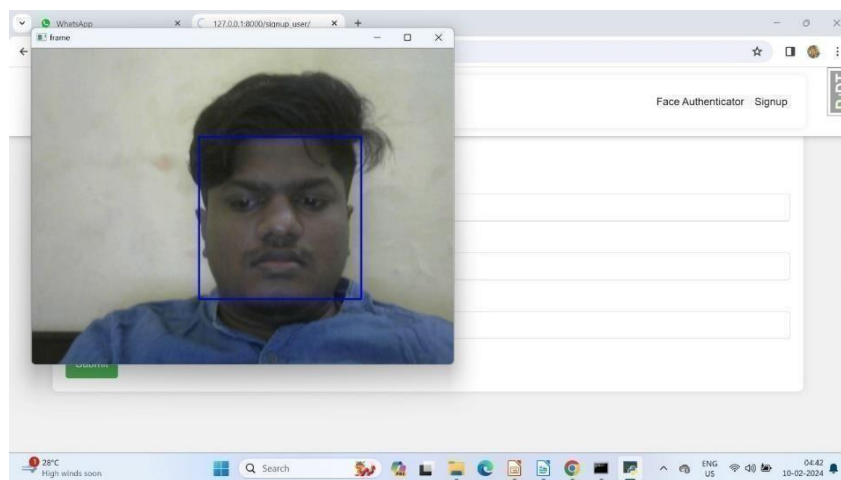


Fig.6.3 during facial recognition

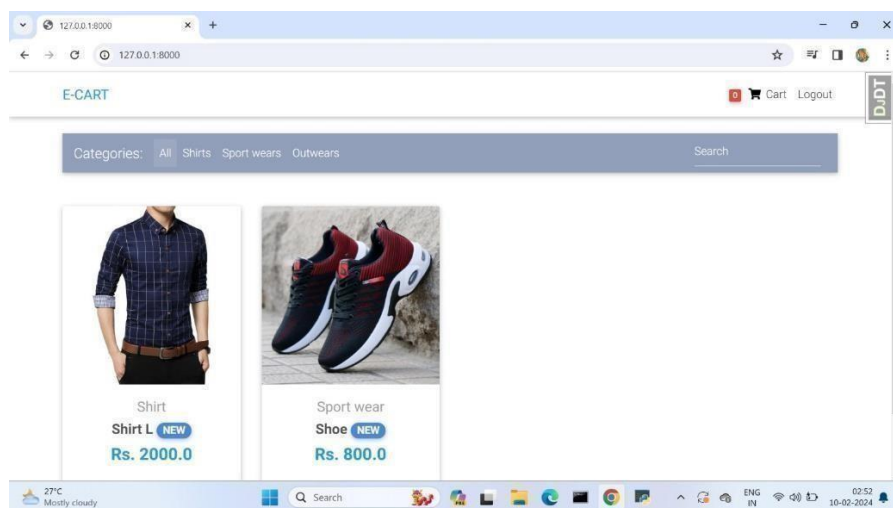


Fig.6.4 dashboard

CHAPTER 7

CONCLUSION

The metrics reflect the fact that the model produced a better performance with a white faces dataset compared to a similar size blackfaces dataset. Thus the pretrained face recognition model used to generate face embeddings that uniquely represent the face had bias toward blackfaces.

The system developed uses the CNN algorithm to detect the face, generates embeddings using a FaceNet pretrained model and a linearSVC classifier to recognize images taken through the system and authenticates users accordingly. The combination of these algorithms resulted in a 95% recognition accuracy. The limitations for this work include the fact that it was determined that the pretrained FaceNet model used had bias towards the black people dataset and the issue of liveness at the login interface, which made it possible for one to login into the system using an image since there is no mechanism for liveness detection. With these limitations, we propose as our future work to retrain the FaceNet model with a black faces dataset to eliminate bias, implement classification with neural networks, integrate liveness detection at the frontend of the system face recognition login, automate the process of classification for digital onboarding of new users.

REFERENCES:

- Hacıoglu, M. Ujgan & Karakas, Geyik, Seda. (2015). An Empirical Research on General Internet Usage Patterns of Undergraduate Students. *Procedia - Social and Behavioral Sciences*. 195. 895-904.10.1016/j.sbspro.2015.06.369.
- van Oorschot, P. C. (2020). User Authentication Passwords, Biometrics and Alternatives. https://doi.org/10.1007/978-3-030-33649-3_3
- Meng, X. (2008). Study on the Model of E-Commerce Identity Authentication based on Multi-biometric Features Identification. *Proceedings-2008 ISECS International Colloquium on Computing, Communication, Control, and Management*, 200.
- A. Alotaibi and A. Mahmmod, "Enhancing OAuth services security by an authentication service with face recognition," 2015 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2015, 2015, doi: 10.1109/LISAT.2015.7160208.
- S. Khokad and V. Kala, "A study of SLIDE Algorithm: Revolutionary AI Algorithm that Speeds Up Deep Learning on CPUs," 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), AURANGABAD, 2020, pp.188-1911
- I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. 2016. [7] SAS. (2020). *Machine Learning*. <https://www.sas.com/en-us/insights/analytics/machinelearning.html#:text=Machine learning is a method,decisions with minimal human intervention.>

APPENDIX

A. SOURCE CODE :

```
import pathlib
import os
import cv2
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import PIL
import warnings
warnings.filterwarnings("ignore")
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential

from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import

ModelCheckpoint, EarlyStopping

from tensorflow.keras.preprocessing.image import load_img

# Defining the path for train and test
images_dir_train =
pathlib.Path("Face/Train")
images_dir_test =
pathlib.Path("Face/Test")

# Count the number of image in Train and Test directory
# Using the glob to retrieve files/pathnames matching a specified pattern.

#Train Image count
image_count_train = len(list(images_dir_train.glob('*/*.jpg'))
print(image_count_train)
```

```

image_dataset =
tf.keras.preprocessing.image_dataset_from_directory(data_dir_train,batch_size=32,
image_size=(180,180),
label_mode='categorical',seed=123)

#all the classes of Skin Cancer
class_names =
image_dataset.class_names

#Dictionary to store the path of image as per the class
files_path_dict = {}

for c in class_names:
files_path_dict[c] = list(map(lambda
x:str(data_dir_train)+'/'+c+'/' +x,os.listdir(str(data_dir_train)+'/'+c)))
#Visualize image
plt.figure(figsize=(15,15))
index = 0
for c in class_names:
path_list =
files_path_dict[c][:1]index +=
1 plt.subplot(3,3,index)
plt.imshow(load_img(path_list[0],target_size=(180,180)))
plt.title(c)
plt.axis("off")

def class_distribution_count(directory):
#count number of image in each
classescount= []
for path in pathlib.Path(directory).iterdir():
if path.is_dir():
count.append(len([name for name in
os.listdir(path)if
os.path.isfile(os.path.join(path, name))]))
#name of the classes
sub_directory = [name for name in
os.listdir(directory)if
os.path.isdir(os.path.join(directory, name))]
#return dataframe with image count and class.
return pd.DataFrame(list(zip(sub_directory,count)),columns =['Class',
'No. of Image'])

```

df

```
#Visualize the Number of image in each
class.import seaborn as sns
plt.figure(figsize=(10, 8))
sns.barplot(x="No. of Image", y="Class", data=df,
label="Class")
```

```
# train dataset
train_ds =
tf.keras.preprocessing.image_dataset_from_directory(data_dir_train,
batch_size=32,
image_size=(180,180),
label_mode='categorical',
seed=123,subset="training",
validation_split=0.2)
```

```
# validation
datasetval_ds
=tf.keras.preprocessing.image_dataset_from_directory(data_dir_train,batch_
size=32,
image_size=(180,180),
label_mode='categorical',
seed=123,subset="validation",
validation_split=0.2)
```

#CNN Model Architecture

```
#Sequential allows you to create models layer-by-
layermodel = Sequential()
```

```
model.add(layers.experimental.preprocessing.Rescaling(1./255,input_shape
=(180,180,3))) #Rescaling Layer
```

#First Convolution layer

```
model.add(layers.Conv2D(32,kernel_size=(3,3),activation='relu'))
model.add(layers.MaxPool2D(pool_size=(2,2)))
```

```
model.add(layers.Conv2D(64,kernel_size=(3,3),activation='relu'))
model.add(layers.MaxPool2D(pool_size=(2,2)))
```

#Third Convulation Layer

```
model.add(layers.Conv2D(128,kernel_size=(3,3),activation='relu'))
model.add(layers.MaxPool2D(pool_size=(2,2)))
```

#Dropout layer with 50% Fraction of the input units to drop.

```
model.add(layers.Dropout(0.5))
```

#Flatten Layer

##Keras.layers.flatten function flattens the multi-dimensional input tensors into a single dimension.

```
model.add(layers.Flatten())
```

#Dense Layer

```
model.add(layers.Dense(128,activation='relu'))
```

#Dropout layer with 25% Fraction of the input units to drop.

```
model.add(layers.Dropout(0.25))
```

#Dense Layer with softmax activation function.

#Softmax is an activation function that scales numbers/logits into probabilities.

```
model.add(layers.Dense(len(class_names),activation='softmax'))
```

```
model.summary()
```

```
epochs_range = range(earlystop.stopped_epoch+1)plt.figure(figsize=(15, 10))
```

```
plt.subplot(1, 2, 1)
```

#Plot Model Accuracy

```
plt.plot(history.history['accuracy'])
```

```
plt.plot(history.history['val_accuracy'])
```

```
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epochs_range')
plt.legend(['train', 'val'], loc='upper left')
```

```
#Plot Model Loss
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epochs_range')
plt.legend(['train', 'val'], loc='upper left')
plt.show()
```

```
class UserProfile(models.Model):
    user = models.OneToOneField(
        settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    stripe_customer_id = models.CharField(max_length=50, blank=True,
null=True)
    one_click_purchasing = models.BooleanField(default=False)

    def __str__(self):
        return self.user.username
```

```
class Item(models.Model):
    title = models.CharField(max_length=100)
    price = models.FloatField()
    discount_price = models.FloatField(blank=True, null=True)
    category = models.CharField(choices=CATEGORY_CHOICES,
max_length=2)
    label = models.CharField(choices=LABEL_CHOICES, max_length=1)
    slug = models.SlugField()
    description = models.TextField()
    image = models.ImageField()

    def __str__(self):
        return self.title

    'slug': self.slug
})
```

```

def get_add_to_cart_url(self):
    return reverse("core:add-to-cart", kwargs={
        'slug': self.slug
    })

def get_remove_from_cart_url(self):
    return reverse("core:remove-from-cart", kwargs={
        'slug': self.slug
    })

```

```

class OrderItem(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
                             on_delete=models.CASCADE)
    ordered = models.BooleanField(default=False)
    item = models.ForeignKey(Item, on_delete=models.CASCADE)
    quantity = models.IntegerField(default=1)

    def __str__(self):
        return f"{self.quantity} of {self.item.title}"

    def get_total_item_price(self):
        return self.quantity * self.item.price

    def get_total_discount_item_price(self):
        return (self.quantity * self.item.price)

    def get_amount_saved(self):
        return self.get_total_item_price() - self.get_total_discount_item_price()

    def get_final_price(self):
        if self.item.discount_price:
            return self.get_total_discount_item_price()
        return self.get_total_item_price()

```

```

class Order(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL,

```

```

start_date = models.DateTimeField(auto_now_add=True)
ordered_date = models.DateTimeField()
ordered = models.BooleanField(default=False)
shipping_address = models.ForeignKey(
    'Address', related_name='shipping_address',
on_delete=models.SET_NULL, blank=True, null=True)
billing_address = models.ForeignKey(
    'Address', related_name='billing_address',
on_delete=models.SET_NULL, blank=True, null=True)
payment = models.ForeignKey(
    'Payment', on_delete=models.SET_NULL, blank=True, null=True)
coupon = models.ForeignKey(
    'Coupon', on_delete=models.SET_NULL, blank=True, null=True)
being_delivered = models.BooleanField(default=False)
received = models.BooleanField(default=False)
refund_requested = models.BooleanField(default=False)
refund_granted = models.BooleanField(default=False)

```

'''

1. Item added to cart
2. Adding a billing address
(Failed checkout)
3. Payment
(Preprocessing, processing, packaging etc.)
4. Being delivered
5. Received
6. Refunds

'''

```

def _str_(self):
    return self.user.username

```

```

def get_total(self):
    total = 0
    for order_item in self.items.all():
        total += order_item.get_final_price()
    if self.coupon:
        total -= self.coupon.amount
    return total

```

```

street_address = models.CharField(max_length=100)
apartment_address = models.CharField(max_length=100)
country = CountryField(multiple=False)
zip = models.CharField(max_length=100)
address_type = models.CharField(max_length=1,
choices=ADDRESS_CHOICES)
default = models.BooleanField(default=False)

def __str__(self):
    return self.user.username

class Meta:
    verbose_name_plural = 'Addresses'

class Payment(models.Model):
    stripe_charge_id = models.CharField(max_length=50)
    user = models.ForeignKey(settings.AUTH_USER_MODEL,
                             on_delete=models.SET_NULL, blank=True, null=True)
    amount = models.FloatField()
    timestamp = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.user.username

class Coupon(models.Model):
    code = models.CharField(max_length=15)
    amount = models.FloatField()

    def __str__(self):
        return self.code

class Refund(models.Model):
    order = models.ForeignKey(Order, on_delete=models.CASCADE)
    reason = models.TextField()
    accepted = models.BooleanField(default=False)
    email = models.EmailField()

    def __str__(self):
        return f'{self.pk}'

```

```
def userprofile_receiver(sender, instance, created, *args, **kwargs):
    if created:
        userprofile = UserProfile.objects.create(user=instance
```

B. SCREENSHOTS :

```
Trainer.yml
C:\Users\> OneDrive > Desktop > PROJECT > project > media_root > TrainingImageLabel > ! Trainer.yml

3 opencv_lbpfaces:
4   threshold: 1.7976931348623157e+308
5   radius: 1
6   neighbors: 8
7   grid_x: 8
8   grid_y: 8
9   histograms:
10    - !opencv-matrix
11      rows: 1
12      cols: 16384
13      dt: f
14      data: [ 1.04166670e-02, 8.68055597e-03, 0., 3.47222225e-03,
15             3.47222225e-03, 5.20833349e-03, 3.47222225e-03,
16             2.77777780e-02, 0., 1.73611112e-03, 0., 0., 3.47222225e-03,
17             0., 8.68055597e-03, 1.02430560e-01, 8.68055597e-03,
18             6.94444450e-03, 0., 1.73611112e-03, 1.73611112e-03,
19             1.73611112e-03, 0., 0., 5.20833349e-03, 1.73611112e-03, 0.,
20             0., 8.68055597e-03, 0., 3.99305560e-02, 7.81250000e-02, 0.,
21             0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
22             0., 1.73611112e-03, 0., 0., 0., 0., 0., 0.,
23             1.73611112e-03, 0., 0., 3.47222225e-03, 0., 6.94444450e-03,
24             5.20833349e-03, 1.73611112e-03, 1.73611112e-03, 0.,
25             1.73611112e-03, 1.73611112e-03, 0., 0., 1.73611112e-03, 0.,
26             0., 0., 1.73611112e-03, 0., 0., 1.73611112e-03, 0., 0.,
27             0., 0., 0., 0., 0., 0., 0., 0., 1.73611112e-03, 0.,
28             1.73611112e-03, 6.94444450e-03, 1.73611112e-03,
29             1.73611112e-03, 0., 0., 0., 0., 0., 0., 0., 0.,
30             0., 0., 0., 1.73611119e-02, 0., 0., 0., 0., 0.,
31             1.73611112e-03, 3.47222225e-03, 0., 0., 0., 6.94444450e-03,
```

Fig.B.1 Image bit

```
rating.js
C:\Users\> OneDrive > Desktop > PROJECT > project > static_in_env > js > addons > js > rating.js > ...

1 var $stars;
2
3 jQuery(document).ready(function ($) {
4
5   // Custom whitelist to allow for using HTML tags in popover content
6   var myDefaultWhitelist = $.fn.tooltip.Constructor.Default.whitelist
7   myDefaultWhitelist.textarea = [];
8   myDefaultWhitelist.button = [];
9
10  $stars = $('rate-popover');
11
12  $stars.on('mouseover', function () {
13    var index = $(this).attr('data-index');
14    markStarsAsActive(index);
15  });
16
17  function markStarsAsActive(index) {
18    unmarkActive();
19    for (var i = 0; i <= index; i++) {
20      $($stars.get(i)).addClass('amber-text');
21    }
22  }
23
24  function unmarkActive() {
25    $stars.removeClass('amber-text');
26  }
27
28  $stars.on('click', function () {
29    $stars.popover('hide');
```

Fig.B.2 Image processing

The screenshot shows a code editor with the file 'bootstrap.js' open. The code is written in JavaScript and defines a factory function for a jQuery plugin. The code includes type checks for 'exports' and 'define', and uses 'define.amd' to support both AMD and CommonJS module systems. It defines a 'global' object and a 'factory' function. The 'factory' function is called with 'global.bootstrap' as the first argument and 'global.jQuery' and 'global.Popper' as the second and third arguments. The code defines a 'Popper' object and a 'defineProperties' function. The 'defineProperties' function is used to define properties on the 'Popper' object. The code also includes a 'createClass' function and a 'defineProperty' function. The code is written in a dark theme with syntax highlighting. The editor's status bar at the bottom shows 'Ln 1, Col 1', 'Spaces: 2', 'UTF-8', 'LF', and 'JavaScript'. The system tray at the bottom shows the date '07-04-2024' and the time '14:51'.

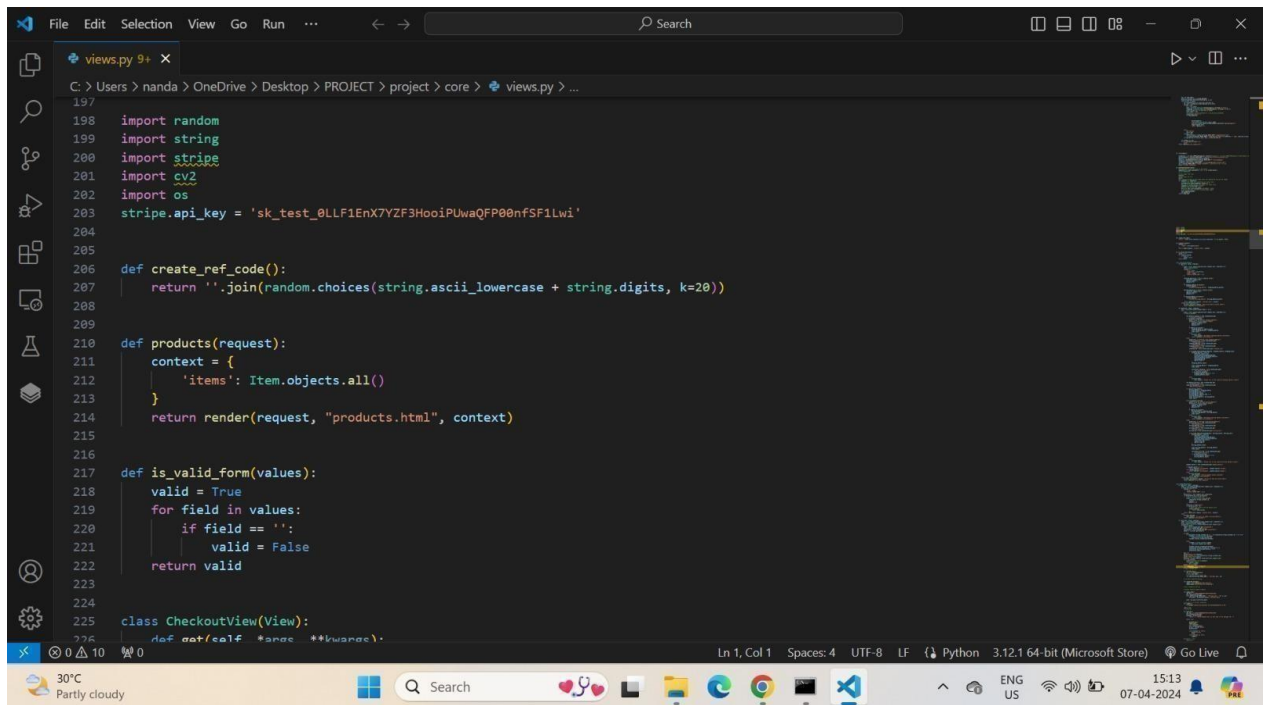
```
1 function (global, factory) {
2   typeof exports === 'object' && typeof module !== 'undefined' ? factory(exports, require('jquery'), require('popper.js')) :
3   typeof define === 'function' && define.amd ? define(['exports', 'jquery', 'popper.js'], factory) :
4   (global = global || self, factory(global.bootstrap = {}, global.jQuery, global.Popper));
5 }(this, function (exports, $, Popper) { 'use strict';
6
7   $ = $ && $.hasOwnProperty('default') ? $['default'] : $;
8   Popper = Popper && Popper.hasOwnProperty('default') ? Popper['default'] : Popper;
9
10  function _defineProperties(target, props) {
11    for (var i = 0; i < props.length; i++) {
12      var descriptor = props[i];
13      descriptor.enumerable = descriptor.enumerable || false;
14      descriptor.configurable = true;
15      if ("value" in descriptor) descriptor.writable = true;
16      Object.defineProperty(target, descriptor.key, descriptor);
17    }
18  }
19
20  function _createClass(Constructor, protoProps, staticProps) {
21    if (protoProps) _defineProperties(Constructor.prototype, protoProps);
22    if (staticProps) _defineProperties(Constructor, staticProps);
23    return Constructor;
24  }
25
26  function _defineProperty(obj, key, value) {
27    if (key in obj) {
28      Object.defineProperty(obj, key, {
29        value: value,
```

Fig.B.3 Django server code

The screenshot shows a code editor with the file 'bootstrap.js' open. The code is written in JavaScript and defines a factory function for a jQuery plugin. The code includes type checks for 'exports' and 'define', and uses 'define.amd' to support both AMD and CommonJS module systems. It defines a 'global' object and a 'factory' function. The 'factory' function is called with 'global.bootstrap' as the first argument and 'global.jQuery' and 'global.Popper' as the second and third arguments. The code defines a 'Popper' object and a 'defineProperties' function. The 'defineProperties' function is used to define properties on the 'Popper' object. The code also includes a 'createClass' function and a 'defineProperty' function. The code is written in a dark theme with syntax highlighting. The editor's status bar at the bottom shows 'Ln 1, Col 1', 'Spaces: 2', 'UTF-8', 'LF', and 'JavaScript'. The system tray at the bottom shows the date '07-04-2024' and the time '15:00'.

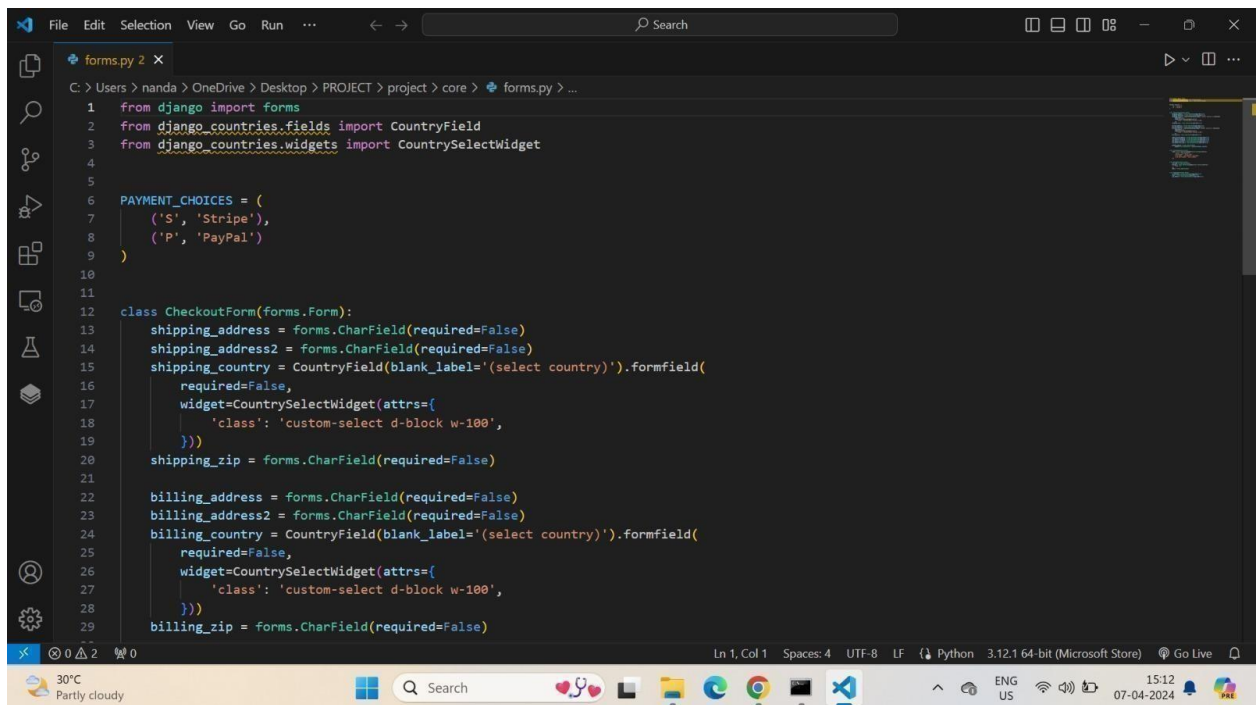
```
1 function (global, factory) {
2   typeof exports === 'object' && typeof module !== 'undefined' ? factory(exports, require('jquery'), require('popper.js')) :
3   typeof define === 'function' && define.amd ? define(['exports', 'jquery', 'popper.js'], factory) :
4   (global = global || self, factory(global.bootstrap = {}, global.jQuery, global.Popper));
5 }(this, function (exports, $, Popper) { 'use strict';
6
7   $ = $ && $.hasOwnProperty('default') ? $['default'] : $;
8   Popper = Popper && Popper.hasOwnProperty('default') ? Popper['default'] : Popper;
9
10  function _defineProperties(target, props) {
11    for (var i = 0; i < props.length; i++) {
12      var descriptor = props[i];
13      descriptor.enumerable = descriptor.enumerable || false;
14      descriptor.configurable = true;
15      if ("value" in descriptor) descriptor.writable = true;
16      Object.defineProperty(target, descriptor.key, descriptor);
17    }
18  }
19
20  function _createClass(Constructor, protoProps, staticProps) {
21    if (protoProps) _defineProperties(Constructor.prototype, protoProps);
22    if (staticProps) _defineProperties(Constructor, staticProps);
23    return Constructor;
24  }
25
26  function _defineProperty(obj, key, value) {
27    if (key in obj) {
28      Object.defineProperty(obj, key, {
29        value: value,
```

Fig.B.4 Facial training



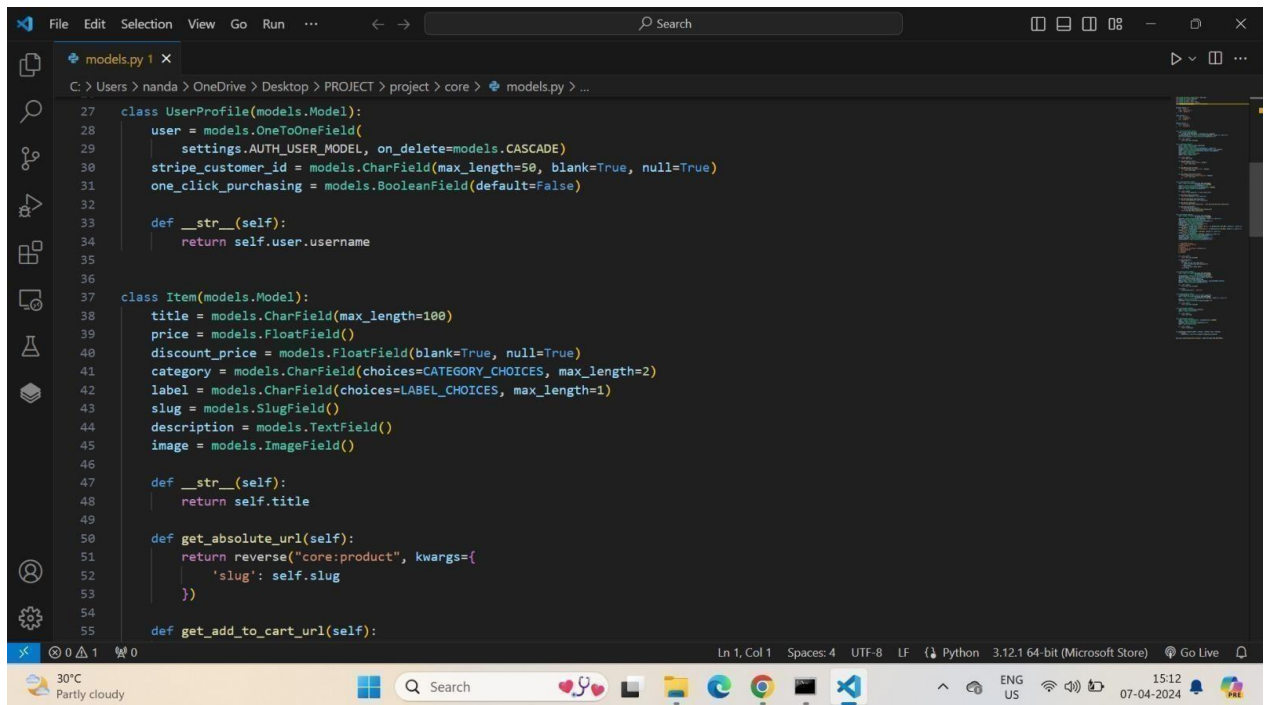
```
197
198 import random
199 import string
200 import stripe
201 import cv2
202 import os
203 stripe.api_key = 'sk_test_0LLF1EnX7YZF3HooiPUwaQFP00nfSF1Lwi'
204
205
206 def create_ref_code():
207     return ''.join(random.choices(string.ascii_lowercase + string.digits, k=20))
208
209
210 def products(request):
211     context = {
212         'items': Item.objects.all()
213     }
214     return render(request, "products.html", context)
215
216
217 def is_valid_form(values):
218     valid = True
219     for field in values:
220         if field == '':
221             valid = False
222     return valid
223
224
225 class CheckoutView(View):
226     def get(self, request):
227         context = {
228             'items': Item.objects.all()
229         }
230         return render(request, "checkout.html", context)
231
232
233 class CheckoutForm(forms.Form):
234     shipping_address = forms.CharField(required=False)
235     shipping_address2 = forms.CharField(required=False)
236     shipping_country = CountryField(blank_label='(select country)').formfield(
237         required=False,
238         widget=CountrySelectWidget(attrs={
239             'class': 'custom-select d-block w-100',
240         }))
241     shipping_zip = forms.CharField(required=False)
242
243     billing_address = forms.CharField(required=False)
244     billing_address2 = forms.CharField(required=False)
245     billing_country = CountryField(blank_label='(select country)').formfield(
246         required=False,
247         widget=CountrySelectWidget(attrs={
248             'class': 'custom-select d-block w-100',
249         }))
250     billing_zip = forms.CharField(required=False)
```

Fig.B.5 Website cart



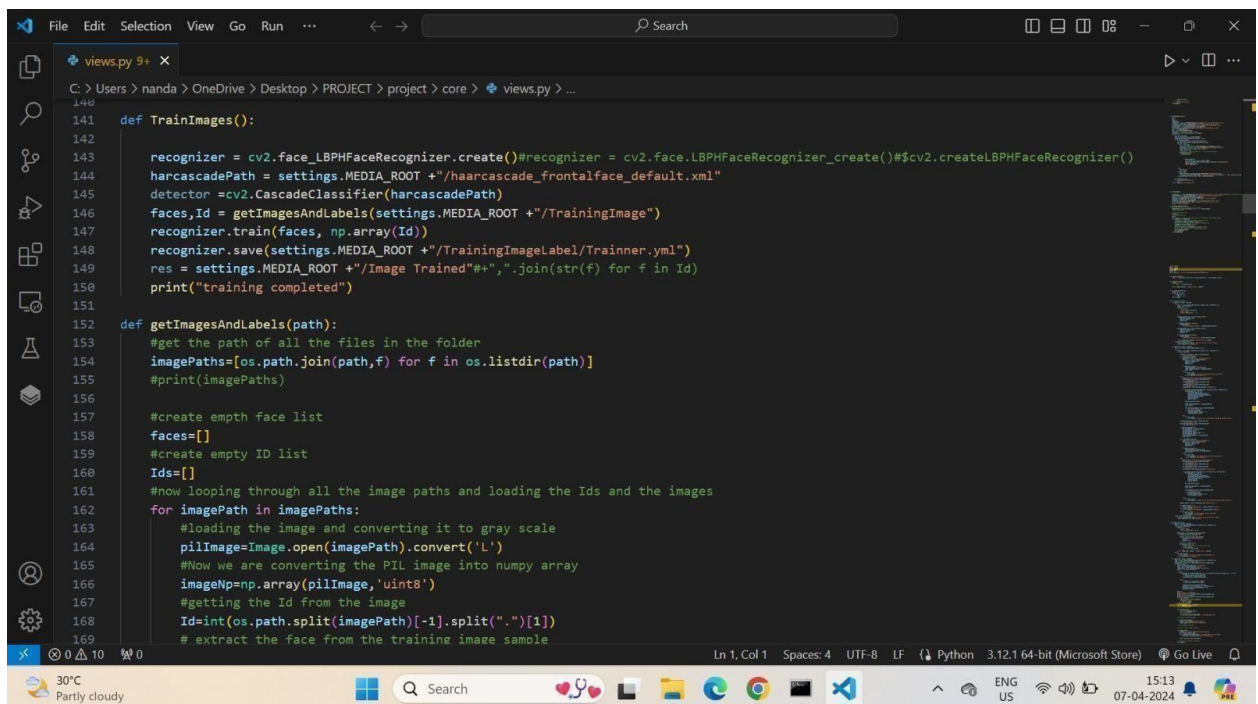
```
1 from django import forms
2 from django_countries.fields import CountryField
3 from django_countries.widgets import CountrySelectWidget
4
5
6 PAYMENT_CHOICES = (
7     ('S', 'Stripe'),
8     ('P', 'PayPal')
9 )
10
11
12 class CheckoutForm(forms.Form):
13     shipping_address = forms.CharField(required=False)
14     shipping_address2 = forms.CharField(required=False)
15     shipping_country = CountryField(blank_label='(select country)').formfield(
16         required=False,
17         widget=CountrySelectWidget(attrs={
18             'class': 'custom-select d-block w-100',
19         }))
20     shipping_zip = forms.CharField(required=False)
21
22     billing_address = forms.CharField(required=False)
23     billing_address2 = forms.CharField(required=False)
24     billing_country = CountryField(blank_label='(select country)').formfield(
25         required=False,
26         widget=CountrySelectWidget(attrs={
27             'class': 'custom-select d-block w-100',
28         }))
29     billing_zip = forms.CharField(required=False)
```

Fig.B.6 Bootstrap javascript



```
27 class UserProfile(models.Model):
28     user = models.OneToOneField(
29         settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
30     stripe_customer_id = models.CharField(max_length=50, blank=True, null=True)
31     one_click_purchasing = models.BooleanField(default=False)
32
33     def __str__(self):
34         return self.user.username
35
36
37 class Item(models.Model):
38     title = models.CharField(max_length=100)
39     price = models.FloatField()
40     discount_price = models.FloatField(blank=True, null=True)
41     category = models.CharField(choices=CATEGORY_CHOICES, max_length=2)
42     label = models.CharField(choices=LABEL_CHOICES, max_length=1)
43     slug = models.SlugField()
44     description = models.TextField()
45     image = models.ImageField()
46
47     def __str__(self):
48         return self.title
49
50     def get_absolute_url(self):
51         return reverse("core:product", kwargs={
52             'slug': self.slug
53         })
54
55     def get_add_to_cart_url(self):
```

Fig.B.7 For training



```
141 def TrainImages():
142
143     recognizer = cv2.face.LBPHFaceRecognizer_create()#recognizer = cv2.face.LBPHFaceRecognizer_create()#$cv2.createLBPHFaceRecognizer()
144     haarcascadePath = settings.MEDIA_ROOT + "/haarcascade_frontalface_default.xml"
145     detector = cv2.CascadeClassifier(haarcascadePath)
146     faces,Id = getImagesAndLabels(settings.MEDIA_ROOT + "/TrainingImage")
147     recognizer.train(faces, np.array(Id))
148     recognizer.save(settings.MEDIA_ROOT + "/TrainingImageLabel/Trainer.yml")
149     res = settings.MEDIA_ROOT + "/Image Trained" + ".join(str(f) for f in Id)
150     print("training completed")
151
152 def getImagesAndLabels(path):
153     #get the path of all the files in the folder
154     imagePath=[os.path.join(path,f) for f in os.listdir(path)]
155     #print(imagePaths)
156
157     #create empty face list
158     faces=[]
159     #create empty ID list
160     Ids=[]
161     #now looping through all the image paths and loading the Ids and the images
162     for imagePath in imagePath:
163         #loading the image and converting it to gray scale
164         pilImage=Image.open(imagePath).convert('L')
165         #Now we are converting the PIL image into numpy array
166         imageNp=np.array(pilImage,'uint8')
167         #getting the Id from the image
168         Id=int(os.path.splitext(imagePath)[-1].split(".")[1])
169         # extract the face from the training image sample
```

Fig.B.8 Image training

C. Research paper

Web based facial recognition and authentication

VINAY MANNAM
FINAL YEAR STUDENT(UG)
DEPARTMENT OF CSE
SATHYABAMA INSTITUTE OF
SCIENCE AND TECHNOLOGY
vinaymannam123@gmail.com

M SIDDHU MAHANANDA
FINAL YEAR STUDENT(UG)
DEPARTMENT OF CSE
SATHYABAMA INSTITUTE OF
SCIENCE AND TECHNOLOGY
nanda63030@gmail.com

MS. BALASAIGAYATHRI
ASSISTANT PROFESSOR
DEPARTMENT OF CSE
SATHYABAMA INSTITUTE OF
SCIENCE AND TECHNOLOGY
balasaigayatri.cse@sathyabama.ac.in

Abstract - Web-based facial recognition and authentication have gained significant attention in recent years due to the growing need for secure and convenient identity verification methods. This technology utilizes advanced algorithms to capture, analyze, and compare facial features for authentication purposes. The primary goal is to accurately verify the identity of individuals accessing online services or digital platforms, mitigating the risk of unauthorized access and fraudulent activities. The web-based approach allows users to conveniently access and utilize the system from any location, using devices such as smartphones, tablets, or computers with internet connectivity. The process involves capturing the user's facial image through the device's camera, then extracting and encrypting facial features to ensure privacy and security. The encrypted data is transmitted securely to a cloud-based server for processing, where it is compared to a stored database of registered users. The matching algorithm evaluates the similarity between the captured image and the stored templates, providing a confidence score for authentication. **Keywords:** facial recognition, authentication, web-based, identity verification, algorithms, secure, convenient, online services, digital platforms, unauthorized access, fraudulent activities, smartphones, tablets, computers, internet connectivity, camera, facial image, encrypting, privacy, security, cloud-based server, processing, matching algorithm, database, registered users, confidence score.

I. INTRODUCTION

Web-based facial recognition and authentication technology has emerged as a powerful tool in enhancing security and user experience on various online platforms. This cutting-edge solution leverages artificial intelligence and computer vision techniques to verify an individual's identity by analyzing their unique facial features. With the proliferation of online transactions, social media engagement, and remote access to sensitive data, the need for secure and convenient authentication methods has become paramount. By harnessing facial recognition technology, businesses and organizations can offer their users a seamless and user-friendly experience while ensuring the utmost security.

At its core, the crux of web-based facial recognition and

authentication systems consist of two major components: face detection and face verification. Face detection algorithms instantly spot and locate faces within an image or video frame, allowing the system to focus its attention on the facial region. This step is crucial as it ensures that the subsequent analysis is performed on the correct area of interest. Once the face is detected, face verification algorithms compare the extracted facial features against a pre-registered template to determine whether the person is who they claim to be. These algorithms assess various facial attributes such as the shape of the eyes, nose, and mouth, along with other unique identifiers like facial landmarks and textures. Through advanced machine learning techniques, the system learns to accurately distinguish between genuine users and impostors, even in challenging scenarios such as different lighting conditions, facial expressions, or minor changes in appearance.

The integration of web-based facial recognition and authentication technology on online platforms brings numerous benefits. Firstly, it provides a highly secure method of user identification. Since each person's facial features are distinct and difficult to replicate, the chances of unauthorized access or identity theft are significantly reduced. Additionally, facial recognition eliminates the need for complicated passwords that can be forgotten or easily hacked, consequently enhancing the overall user experience. Users can effortlessly and securely log in to their accounts by simply presenting their face to the camera, eliminating the hassle of remembering or inputting complex passwords.

Moreover, web-based facial recognition technology offers a high level of versatility and scalability. It can be seamlessly integrated into various applications, ranging from e-commerce platforms and financial institutions to social media networks and government services. This adaptability ensures that the technology can cater to diverse industry needs, delivering a consistent and reliable authentication experience across different domains.

As with any technology, it is essential to address potential privacy concerns associated with facial recognition. Strict regulations and privacy policies need to be in place to ensure that user data is protected and used solely for authentication purposes. Transparency and user consent should be given utmost priority to build trust and foster acceptance of this technology among individuals.

In conclusion, web-based facial recognition and authentication technology presents an innovative solution to enhance security and streamline user experience in the digital world. With its ability to accurately verify identities in real-time, this technology has the potential to revolutionize online authentication, offering an efficient, secure, and user-friendly alternative to traditional methods.

II. RELATED WORKS

1. Facial Authentication System by combining feature extraction using Raspberry Pi [1]: This study focuses on the development of a facial authentication system using Raspberry Pi. The authors propose a method that combines feature extraction techniques to enhance the accuracy and efficiency of the system.

2. Smart Home Security Solutions using Facial Authentication and Speaker Recognition through Artificial Neural Networks [2]: This research article discusses the implementation of smart home security systems that utilize both facial authentication and speaker recognition. The authors employ artificial neural networks to enhance the security and reliability of the system.

3. RFace: Anti-Spoofing Facial Authentication using COTS RFID [3]: In this paper, the authors introduce RFace, a facial authentication system that utilizes commercial off-the-shelf (COTS) RFID technology to counter spoofing attacks. The proposed system ensures secure and reliable user authentication.

4. Facial Image Recognition for Biometric Authentication Systems using Geometrical Feature Points and Low-Level Visual Features [4]: This study presents a facial image recognition technique that combines geometrical feature points and low-level visual features for biometric authentication systems. The authors aim to improve the accuracy and robustness of facial recognition algorithms.

5. Security of Facial Biometric Authentication for Attendance System [5]: This research paper focuses on the security aspects of facial biometric authentication systems used in attendance systems. The authors propose measures to enhance the security and protect against unauthorized access or identity fraud.

6. Robust Human Face Authentication Leveraging Acoustic Sensing on Smartphones [6]: This article introduces a robust human face authentication system that leverages acoustic sensing capabilities on smartphones. The authors demonstrate the potential of utilizing acoustic features to enhance the security and accuracy of facial authentication systems.

7. Multifactor Authentication System based on Color Visual Cryptography, Facial Recognition, and Dragonfly Optimization [7]: This research presents a multifactor authentication system that combines color visual

cryptography, facial recognition, and dragonfly optimization. The authors aim to enhance the security of the authentication process by integrating multiple factors.

8. Real-Time Multimodal Biometric Authentication of Human Using Face Feature Analysis [8]: This paper discusses the real-time multimodal biometric authentication of humans using face feature analysis. The authors propose an approach that analyzes various face features to achieve reliable and efficient authentication.

9. AI-Based Login System using Facial Recognition [9]: This study presents an AI-based login system that utilizes facial recognition. The authors explore the use of artificial intelligence techniques to improve the accuracy and efficiency of the authentication process.

10. Authentication Mechanisms and Classification: A Literature Survey [10]: This survey paper provides an overview of authentication mechanisms and their classification. The authors discuss different authentication techniques used in various applications and provide insights into the current research trends in authentication systems.

Note: The numbers in square brackets correspond to the references provided at the beginning of each description.

III. EXISTING SYSTEM

The existing system for web-based facial recognition and authentication comes with a range of disadvantages that impede its effectiveness and widespread adoption. Firstly, one of the main issues lies in the accuracy and reliability of the technology itself. Facial recognition systems continue to struggle with accuracy, especially when it comes to diverse populations. Studies have shown that these systems often produce higher false positive rates for people with darker skin tones or other distinctive facial features, leading to potential discrimination and exclusion. Furthermore, facial recognition can also be easily fooled by simple alterations in appearance, such as changes in lighting conditions, facial expressions, or the presence of accessories, making it vulnerable to spoofing attacks.

Another significant disadvantage of the existing system is the concern over privacy and security. Facial recognition technology raises serious questions about the collection, storage, and usage of individuals' biometric information. The potential for misuse and abuse of this sensitive data is alarming, as it could be accessed by unauthorized entities or even used for surveillance purposes without consent. Additionally, the centralized storage of biometric information poses a significant risk of data breaches, potentially leading to the compromise of vast amounts of personal information.

Furthermore, the computational requirements of facial recognition systems can be a practical challenge. The processing power needed to analyze and match facial

features in real-time can be resource-intensive, leading to increased costs for deploying and maintaining these systems. Smaller organizations or individuals may face limitations in adopting facial recognition due to these technical requirements.

Moreover, the existing system lacks standardization and interoperability. Many facial recognition solutions are proprietary and lack compatibility with each other, creating significant barriers to integration and collaboration across different platforms and systems. This lack of interoperability hinders the scalability and effectiveness of facial recognition technology, limiting its potential applications and usefulness in various industries.

Lastly, ethical concerns surrounding facial recognition technology cannot be overlooked. There are valid concerns about invasions of privacy, potential biases, and the lack of transparency in how these systems operate. As the technology becomes more widely adopted, it is crucial to establish clear guidelines and regulations to address these ethical dilemmas and ensure responsible usage.

In conclusion, the existing system for web-based facial recognition and authentication is plagued by accuracy and reliability issues, privacy and security concerns, high computational requirements, lack of standardization, and ethical considerations. Overcoming these disadvantages will be crucial in improving the functionality, fairness, and acceptance of facial recognition technology in the future.

IV. PROPOSED SYSTEM

The proposed work aims to develop a web-based facial recognition and authentication system that offers enhanced security and convenience in various applications. The system will leverage the power of facial recognition technology to accurately identify individuals based on their unique facial features, ensuring reliable authentication for access to sensitive data, secure login to online accounts, and other identity verification purposes.

The primary objective of this work is to enhance the overall user experience by eliminating the need for traditional authentication methods like passwords, PINs, or physical tokens. This will not only reduce the hassle of remembering and managing multiple credentials but also mitigate the risks associated with password-related security breaches. The facial recognition system will provide a seamless and user-friendly experience, allowing individuals to log in or gain access to secure systems by simply looking at a camera.

Moreover, the proposed system will prioritize security to prevent unauthorized access attempts. Advanced algorithms and techniques will be implemented to ensure accurate facial recognition, making it difficult for impostors to bypass the system. Additionally, anti-spoofing measures will be integrated to detect and reject fraudulent attempts involving artificial or replicated

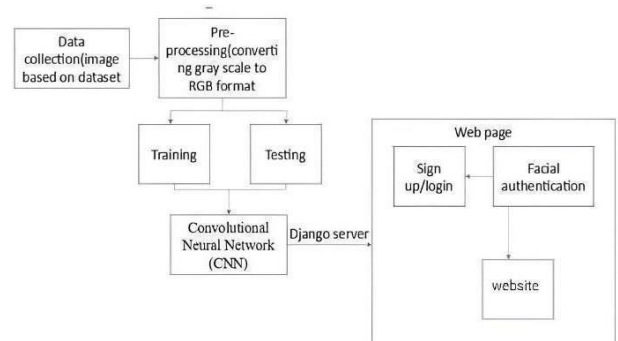
facial features.

To make the system web-based, it will be designed to be easily integrated into existing web applications, websites, or online platforms. This will allow organizations across various industries to implement facial recognition and authentication seamlessly, irrespective of their operational context. The system will support multiple web browsers, ensuring compatibility and accessibility for a wide range of users.

Furthermore, the proposed work will also include research and development of machine learning models to continuously improve the accuracy and efficiency of the facial recognition system. The models will be trained using large datasets to enhance recognition performance and adapt to different facial profiles, lighting conditions, or facial expressions.

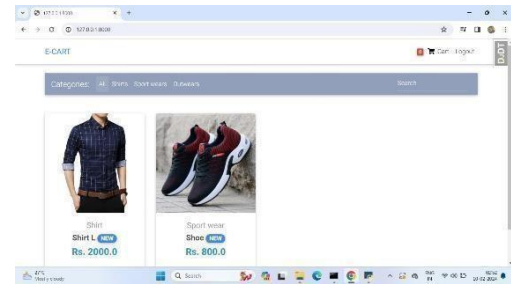
In conclusion, the proposed work aims to develop a robust web-based facial recognition and authentication system that offers heightened security and convenience. By leveraging advanced technologies and algorithms, the system will provide accurate and reliable identification, reducing the reliance on passwords and other traditional authentication methods. Its easy integration and compatibility with web applications will enable widespread adoption across various industries, contributing to a safer and more user-friendly digital environment.

V. SYSTEM ARCHITECTURE

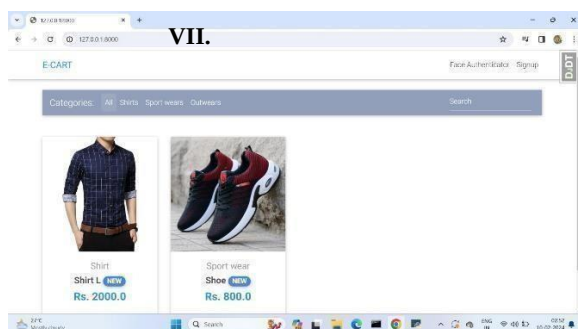
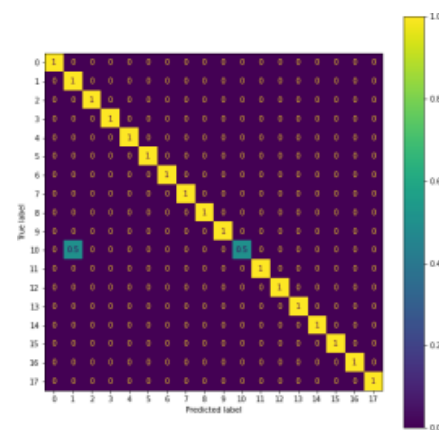


VI. RESULTANDDISCUSSION

Web-based facial recognition and authentication systems have gained significant popularity in recent years due to their ability to provide a secure and convenient method of user authentication. Such systems utilize advanced algorithms and machine learning techniques to analyze and verify the identity of individuals based on their facial features. The process typically involves capturing an image or video of the user's face through a web camera or mobile device, which is then compared to a pre-existing database of facial images. The system matches key facial landmarks and patterns to determine the level of similarity, and if the match exceeds a certain threshold, the user is granted access. These systems offer several advantages over traditional methods of authentication. Firstly, they provide a non-intrusive and user-friendly experience as users do not need to remember and enter complex passwords or PINs. Secondly, they offer a higher level of security as facial features are difficult to replicate or forge. Additionally, web-based facial recognition and authentication systems can be seamlessly integrated into various web applications, such as online banking, e-commerce, and social media platforms, enhancing their overall security and user experience. However, it is important to address potential concerns related to privacy and data protection in deploying these systems. By ensuring proper consent and protecting user data, organizations can leverage the power of web-based facial recognition and authentication systems to deliver a seamless, secure, and user-friendly experience for their customers. Below are the before, during and after for the webpage when a person logs in.



The accuracy of the model we built can be tested using a linear SVC. We now build our model and adjust it to improve our accuracy until we are satisfied. So in case of there being inaccuracies we check the problems and improve upon it. Below is a linear SVC of the model.

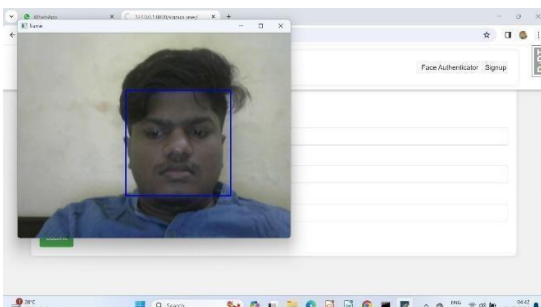


We use various algorithms to measure the accuracy and precision of the model. To do this we use various classification algorithms like random forest and OneVsRest. Below are the findings from these algorithms.

Classifier	Precision	Accuracy	Recall	F1 Score
LinearSVC	0.95	0.95	0.95	0.95
OneVsRest	0.94	0.94	0.94	0.93
Random Forest	0.91	0.91	0.91	0.91

The algorithms we use are haarcascade classifier and MTCNN. Below are the accuracy and precision scores for the algorithms.

Algorithms	Accuracy	Precision
Haarcascade	0.74	0.74
MTCNN	0.98	0.98



CONCLUSION

In conclusion, the system for web-based facial recognition and authentication is a highly efficient and secure method of verifying user identity. Through the use of advanced algorithms and biometric data, it accurately identifies individuals based on their unique facial features. The system offers numerous benefits, such as enhanced user experience, reduced risk of identity fraud, and increased security measures. With its ability to seamlessly integrate into various web-based applications and services, this technology provides a convenient and reliable authentication solution for a wide range of industries, including online banking, e-commerce, and social media platforms. Overall, web-based facial recognition and authentication system represents an innovative and robust approach to verifying user identity, contributing to a safer and more streamlined online experience.

VIII. FUTURE WORK

In the future, one potential area of focus for the advancement of technology could be on the development of systems for web-based facial recognition and authentication. This system would aim to enhance security and convenience by allowing individuals to use their facial biometrics as a means of secure authentication for various online services. The system would involve the integration of facial recognition algorithms into web platforms, enabling users to quickly and accurately verify their identities without the need for passwords or other traditional authentication methods. Research in this field may revolve around improving the accuracy and robustness of facial recognition algorithms, addressing challenges such as varying lighting conditions, pose variations, and potential security vulnerabilities. Additionally, user privacy and data protection would be fundamental concerns, necessitating the development of secure protocols that ensure the confidentiality and integrity of facial biometric data. Future work could also explore the integration of multi-factor authentication, combining facial recognition with other biometric or behavioral authentication methods, to further increase the security of web-based systems. Overall, further advancements in web-based facial recognition and authentication have the potential to significantly enhance security, convenience, and user experience in the digital realm.

IX. REFERENCES

- [1] Gabriel, J. Z., Kabilan, R., Devaraj, G. P., & Muthuraman, U. (2021, February). Facial authentication system by combining of feature extraction using raspberry Pi. In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) (pp. 1142-1145). IEEE.
- [2] Saxena, N., & Varshney, D. (2021). Smart home security solutions using facial authentication and speaker recognition through artificial neural networks. *International Journal of Cognitive Computing in Engineering*, 2, 154-164.
- [3] Xu, W., Liu, J., Zhang, S., Zheng, Y., Lin, F., Han, J., ... & Ren, K. (2021, May). RFace: anti-spoofing facial authentication using cotsrfid. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications* (pp. 1-10). IEEE.
- [4] Vasanthi, M., & Seetharaman, K. (2022). Facial image recognition for biometric authentication systems using a combination of geometrical feature points and low-level visual features. *Journal of King Saud University-Computer and Information Sciences*, 34(7), 4109-4121.
- [5] Wati, V., Kusrini, K., Al Fatta, H., & Kapoor, N. (2021). Security of facial biometric authentication for attendance system. *Multimedia Tools and Applications*, 80, 23625-23646.
- [6] Zhou, B., Xie, Z., Zhang, Y., Lohokare, J., Gao, R., & Ye, F. (2021). Robust human face authentication leveraging acoustic sensing on smartphones. *IEEE Transactions on Mobile Computing*, 21(8), 3009-3023.
- [7] Ibrahim, D. R., Teh, J. S., & Abdullah, R. (2021). Multifactor authentication system based on color visual cryptography, facial recognition, and dragonfly optimization. *Information Security Journal: A Global Perspective*, 30(3), 149-159.
- [8] Srivastava, R., Tomar, R., Sharma, A., Dhiman, G., Chilamkurti, N., & Kim, B. G. (2021). Real-Time Multimodal Biometric Authentication of Human Using Face Feature Analysis. *Computers, Materials & Continua*, 69(1).
- [9] Girmay, S., Samsom, F., & Khattak, A. M. (2021, October). Ai based login system using facial recognition. In *2021 5th Cyber Security in Networking Conference (CSNet)* (pp. 107-109). IEEE.
- [10] Chenchchev, I., Aleksieva-Petrova, A., & Petrov, M. (2021). Authentication Mechanisms and Classification: A Literature Survey. In *Intelligent Computing: Proceedings of the 2021 Computing Conference*, Volume 3.