An Expert model for Deep Face Recognition

Subhasish Mohapatra 1,\* Biplab Biswas2 Sanchari Dey3

*1* Dept. of Computer Science & Engineering, Adamas University, Kolkata, West Bengal, India

2Dept of Master of Computer application, Adamas University, Kolkata, West Bengal, India

3 Dept of Master of Computer application, Adamas University, Kolkata, West Bengal, India

1,\*mohapatra.subhasish@gmail.com, 2biswasbiplab98@gmail.com,3deysanchari8@gmail.com

**ABSTRACT**

The simplest approach to determine each person's unique identity is to recognize their face. The two main phases of a human face recognition system are face detection and face recognition. The terms "face detection" and "face recognition" refer to the process of determining whether or not a picture contains a face. Convolutional Neural Network, or CNN, is important in this. There are several neural network libraries, including TFlearn and Keras. This report is mostly based on TensorFlow and TFlearn, meaning that a model was developed using these libraries. For visualisation of the result, another dataset (without labelling) is prepared and this time, a model predicted the label (i.e. name) of those photos. A dataset (labelled data) is required for model creation and is generated using OpenCV. The accuracy was found to be quite high—nearly all images were predicted accurately. This library stores the information we wish to keep (such as the names of authorized individuals). The face recognition system project's focus area is image processing.

Keyword: - Face Recognition System, OpenCV, CNN, TensorFlow, TFlearn are some of the terms used.

1. **INTRODUCTION**

Finding instances of actual objects, such as a vehicle, television, or a person, in pictures or videos is a technique called object detection. It makes object location, detection, and recognition possible in images. It is frequently utilized in applications like image retrieval, security systems, and even crowd counting. Feature-based and deep learning methods of object detection are only two examples of the many methods available.

A previously detected item must be recognized as a known or unknown face in order to be classified as face recognition. Tensor Flow and TFlearn are two of the most important libraries utilized for this project. For dataflow programming across a variety of activities, Google offers TensorFlow, an open-source machine learning framework. TensorFlow’s core is the foundation of the Python package TFlearn, which is also built on TensorFlow. Predicting photos based on their names is the major goal of this study. It is crucial to create a dataset for every face recognition project. Thus, OpenCV library is created in order to address this. Computer vision and machine learning software library called OpenCV is free and open source. For the purpose of generating datasets and detecting and identifying faces, the library offers more than 2500 optimized algorithms. Perhaps not as accurately as when utilizing deep learning, facial detection and recognition. Therefore, OpenCV is only utilized in this project to generate datasets because doing so with this library is very simple and effective. Systems for facial recognition rely on computer programmers that examine pictures of human faces in order to identify them. The software creates a unique file from a facial image after measuring features like the distance between the eyes, the length of the nose, and the angle of the jaw. It then uses that file to compare the image with another image and generate a score that indicates how similar the two images are to one another.

The goal of the project on Deep Face Recognition System using Deep Learning, TensorFlow, and Tflearn is to create a highly accurate and effective facial recognition system by using the TensorFlow and Tflearn frameworks and deep learning techniques. The project wants to accomplish the following goals:

The main objective is to develop an advanced facial recognition system that outperforms current approaches by utilizing deep learning's capabilities. Convolutional Neural Networks (CNNs), a type of deep learning algorithm, have proven to perform exceptionally well in a variety of computer vision applications, including face recognition. Deep learning techniques may be used to improve the system's face recognition performance by making it more accurate and reliable.

Using TensorFlow and Tflearn: TensorFlow is a well-liked, open-source deep learning framework that offers a customizable and scalable platform for creating and training neural networks. Building deep learning models is made easier by the use of Tflearn, a high-level library built on top of TensorFlow. The project makes use of these frameworks in order to benefit from their broad functionality, optimized implementations, and effective GPU accelerated training and inference.

Enhanced Accuracy: The research uses the capacity of deep learning models to learn discriminative characteristics from huge datasets to enhance the accuracy of face recognition. Deep learning models can automatically pick up on intricate representations of face traits, enabling them to pick up on minor changes and patterns that conventional approaches can find challenging to identify.

By the use of deep learning models for Real-Time Implementation: The project attempts to optimize the deep facial recognition system for real-time operation, making it appropriate for uses where prompt identification is required. Accelerated inference and real-time performance are made possible by TensorFlow and Tflearn's fast GPU compute and parallel processing capabilities. The research attempts to achieve quick and effective face recognition by using these frameworks and optimizing the model design using relevant assessment criteria, such as accuracy, precision, recall, and F1-score, the project compares the performance of the deep facial recognition system. Additionally, the system's effectiveness will be evaluated against other deep learning-based methods as well as conventional ways for face identification. In terms of accuracy, speed, and efficacy, this comparison study aids in evaluating the superiority and effectiveness of the established system.

* 1. **Contribution**

The following objectives might be set in the context of deep face recognition utilizing TensorFlow and Tflearn deep learning technology:

To develop and put into use a TensorFlow and Tflearn-based deep learning-based facial recognition system.

To research and assess several deep learning architectures that are appropriate for face recognition applications. Gathering and pre-processing a large dataset of facial photos while taking position, lighting, and occlusions into account. To improve the performance of the deep learning model by training it with the gathered dataset. To evaluate the effectiveness of the proposed deep learning-based methodology against the currently used facial recognition methods

The outline of this chapter is shown as follows, In Section 2, Literature Review of some of the previous related studies are provided. In Section 3, Technology used. In Section 4, the methodology of the proposed System will be provided. In Section 5, the Hard ware and Software requirements will be provided. In Section 6, the implementation result will be provided.

**2. RELATED WORK**

By examining a captured face's primary features and contrasting them with those on other faces that have been saved in the database, face recognition may be performed naturally. Chronological sequence has been used to offer a concise summary of the relevant works and history. A semi-automated method was used for the first facial recognition effort in the 1960s. Marks were then drawn to indicate where the key characteristics were on the pictures. Mouths, ears, noses, and eyes were among the traits it made use of. Following that, the distances and ratios were calculated between these markers and a common reference point and then compared to reference data.

In their proposed study, (Ilyas et al. 2019) described a facial recognition system based on deep learning neural networks. The face was extracted from the input image using the Viola-Jones face detection method, and then the histogram equalisation algorithm (AHE) was used to modify and enhance the grey level of the image to a homogeneous distribution. Additionally, they used VGG16 and ResNet50-based convolutional neural networks to extract facial characteristics and categorise human faces after preprocessing the input region of interest.

They used two datasets to evaluate their neural networks ResNet50 and VGG16: First, the accuracy of the Extended Yale B Face database was fine-tuned to 96.12% and 97.23%, respectively.

Second, the accuracy scores for the CMU PIE database were 96.55% overall and 98.38% for each. Modern face recognition systems are powered by deep learning-based neural networks; however, training a state-of-the-art convolution neural network requires enormous computational resources since the model must be trained on millions of photos over the course of hundreds of hours. The idea of transfer learning has lately acquired popularity in the face recognition field as a means of getting around this limitation. Transfer learning is a technique used in machine learning where data from one task is utilised to enhance learning in another task. Perdana and Prahara (2019) demonstrate in their paper how this innovative method has made it possible for professionals and academics to break the link between neural networks and computational complexity.

Both studies use the cutting-edge VGG16 architecture to train the input face photos, which have already been pre-trained on the enormous ImageNet database, which has over 1 million images divided into 1000 different categories. Working with various datasets, selects Yale and AT&T face databases with input image sizes of 224 224 x 3, selects resized 120 120 images from ROSE-Youtu Face Liveness Detection Database, and retrains their respective locally altered VGG16 models, each of which consists of the modified input layer, multiple hidden layers, and output layers. As shown by the aforementioned research, Light convolutional neural networks with transfer learning produced accuracy of over 95% in confined and unconstrained situations after retraining for a much shorter period of time on fewer datasets.

Furthermore, (Shakeel and Lam 2019) created an age-invariant facial recognition method based on a discriminative model with deep feature training. In their study, high-level deep features were learned using AlexNet as a CNN model for transfer learning. The traits were then converted into a special code phrase to represent the image in a codebook. The encoding process made guaranteed that pictures of the same person taken at different times all had the same distinctive code words. For face identification, a linear regression-based classifier was employed. Three datasets, including the freely accessible FGNET, were utilised for testing.

(Peng et al. 2019) in their research showed how to use a deep local descriptor learning framework for cross-modality face recognition, in which both compact local information and discriminant features are learnt directly from raw facial patches. The approach was evaluated on six widely used face recognition datasets of various types, with an overall accuracy of 98.68%. CNN is utilised to produce deep local descriptors.

Additionally, a number of commentators have argued in favour of the automation of a number of tedious chores using facial recognition technology, one of which is an attendance system. They explained how Artificial Neural Networks can be trained over billions of images and then used to detect and recognise faces with a fair amount of simplicity and flexibility in this study (Harikrishnan et al. 2019). They divided their study into four primary categories: data administration, detection, training, and recognition. Resizing the original 8 bits grayscale image, enhancing the face characteristics, and reducing noise in the input images were all part of the first stage of image preparation.

The face area was extracted from the input photos using OpenCV's Haar Cascade Method, and the LBPH recognizer was trained on the resulting dataset. Additionally, during facial identification, live video was passed via the LBPH recognizer, which produced LBPs and histograms for each identified face. The greatest match between each newly formed histogram and those already in the training file was noted as being present in the excel report that contained the students' attendance records for a given day. They constructed a prototype with an accuracy of over 70% that was small but sturdy and simple to customise for use with a portable computer.

Additionally, sentiment analysis based on human facial expressions has also been a significant topic of research, much as gender and age estimation algorithms. Happiness, sorrow, fear, wrath, disgust, and surprise are six fundamental human emotions and neutral expressions that Kukla and Nowak (2015) examine in-depth using a cascade of neural networks. In their multistage inquiry, they used single or multilayer configurations of neural networks to identify each separate expression, and at subsequent stages, classifications were produced by combining the data from these different neural networks.

The authors' collection of photos, the John Kanade database, the Karolinska Directed Emotional Faces (KDEF) database, and these sources served as the input to the cascade of classifiers. Different results were achieved for specific emotions, different outcomes were achieved. With an overall accuracy of 76%, surprise and happiness were the most accurate emotions to identify, while fear was the least accurate.

**PROPOSED FRAMEWORK**

**3.1 Requirement analysis**

The process of carefully identifying, specifying, and documenting the numerous requirements connected to a certain business purpose is known as requirement analysis. The process of gathering requirements aids in defining the project's scope and the resources needed to carry it out, as well as in understanding the demands of the client clearly. The project's technical, non-technical, and functional needs are:

**3.1.1 Functional prerequisites:**

The phrase "any requirement which specifies what the system should do"[8] is used to describe a functional requirement. The following is a list of the project's functional requirements:

* It needs to be capable of handling 'png' and 'jpeg' pictures.
* It ought to correctly produce the dataset.
* It needs to be highly accurate in predicting the authorised users.

**3.1.2 Non-functional requirements:**

"Any requirement that specifies how the system performs a certain function"[8] is referred to as a non-functional requirement. They are the features or qualities of the system that allow us to assess how well it functions. The following is a list of the project's non-functional requirements:

* The system's graphical user interface (GUI) will be user-friendly.
* The system will be adaptable to changes, allowing for the addition of authorized users as necessary.
* The system's efficacy and efficiency will be guaranteed.

**3.1.3. Technical requirements:**

The following is a list of the project's technical requirements:

1. **Hardware requirements are as follows:**

* system with integrated cameras

**b) Software Prerequisites:**

* Windows-based operating system
* The 1.14 version of TensorFlow
* Using OpenCV

**3.2 Feasibility study**

A feasibility study examines the likelihood of a project's success. It also provides a strong framework for creating our business strategy. The three following elements are part of this study:

**3.2.1 Technical feasibility**

The feasibility that is concerned with specifying hardware and software that will effectively meet the user demand is known as technical feasibility. This idea is theoretically viable since the necessary hardware and software for the creation of this system are currently freely and openly available.

**3.2.2 Economic feasibility**

A projects or company' economic viability is determined by its financial and logistical viability. This idea is also economically possible because all the needs are obtained from the internet and there are no fees of any kind related to this system.

**3.3.3 Operational feasibility**

Operational feasibility is a measure of how well a new system is able to solve challenges. This project is also operationally possible since the system addresses the problem of lost keys or password/code breaches as stated in the problem description i.e given in Figure-1.

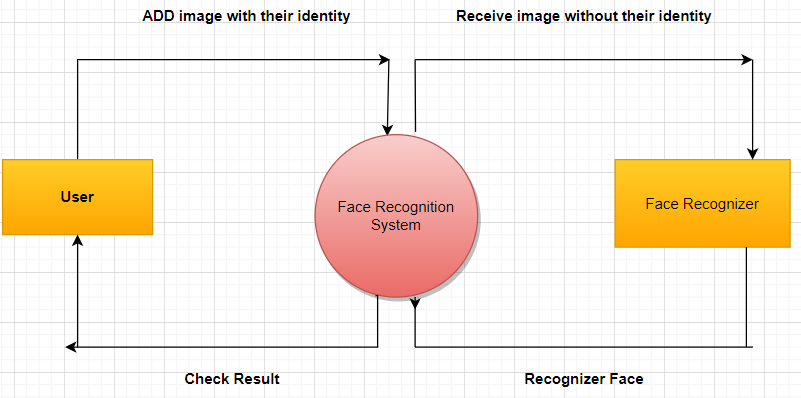


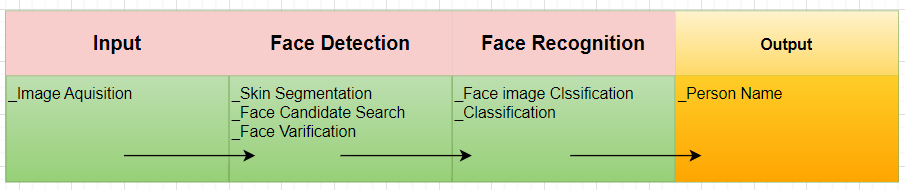
Fig-1Proposed Model for Image analysis

**4 .METHODOLOGY**

The proposed methodology of our model Gather a labelled picture collection with samples of both faces and objects without faces. There should be a range of backdrops, lighting setups, and face expressions in this dataset. Mark up the pictures to say if there are faces there or not. Data preprocessing: Improve the dataset's quality and performance by preprocessing it. Use methods like grayscale conversion, normalization, and scaling to guarantee uniformity in the supplied pictures. Using the OpenCV package, we can train a HaarCascade classifier. In order to do this, positive samples (faces) and negative samples (non-faces) must be defined. Then, pertinent characteristics must be extracted using Haar-like features. To create a powerful face classifier, train the classifier using machine learning techniques like Adaboost or Support Vector Machines (SVM). HaarCascade Face Detection: Use the learned HaarCascade classifier to find faces in the pictures. Use sliding windows of various widths to apply the classifier to the image, then look for areas that match the faces previously discovered patterns. Create bounding boxes all the way around the faces that were found. Create a CNN architecture with dense layers and maximum pooling layers for face categorization. Through convolutional and pooling layers, the CNN should learn hierarchical representations of facial attributes from input pictures. To improve performance, test out various network designs, layer setups, and hyperparameters**.**

**4.1 Face Recognition Approach**

The face recognition system approach is illustrated in below figure-2:

****

**Figure-2Face recognition Approach**

**4.2** **Design of Face Recognition System**

Facial detection and facial recognition are at the core of this technology. We chose to employ the Haar-Like features technique for the face detection component and the neural network approach for the face identification part out of all the potential options.

The button to take pictures appears when the GUI is opened, allowing the user to take pictures with their camera. Following picture capture, pre-processing is carried out, followed by face detection and identification, with the end result being displayed to the user.

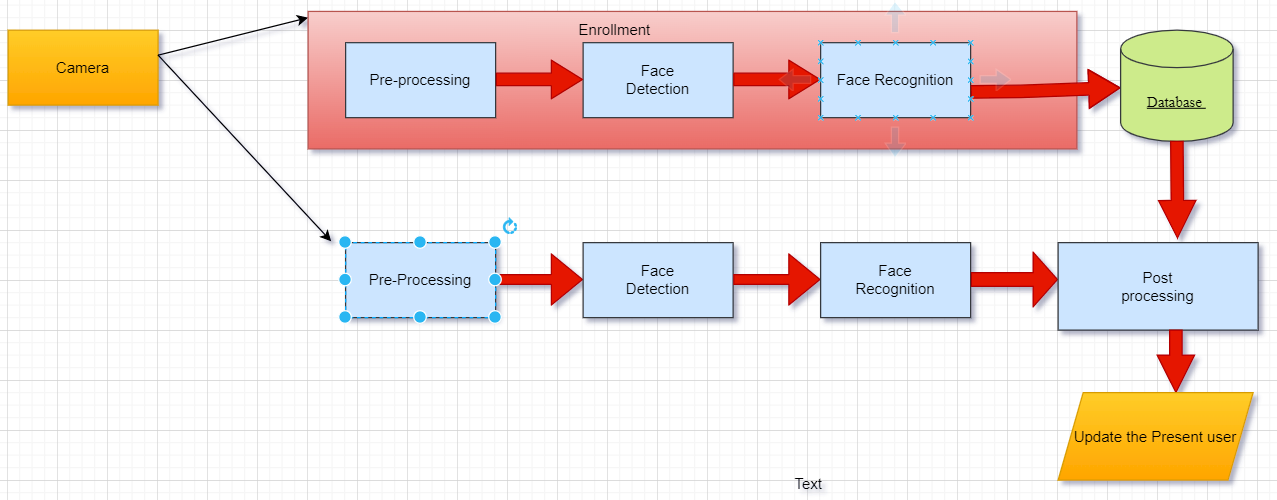
**Figure 4.2: System architecture for Face Recognition**

**4.3 Input part**

**For a facial recognition system to function, input is necessary. In this section, an image acquisition procedure is carried out. For the sake of completing image-processing calculations, live collected pictures are transformed to digital data. The face detection algorithm is provided these collected images**

**4.4 Face Detection Part.**

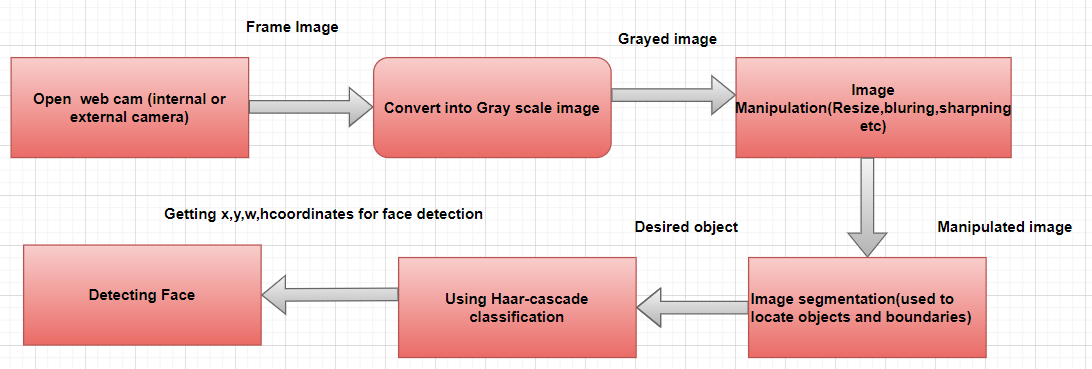
The method of locating a human face in a picture is called face detection. The face detection procedure is described below in Figure-3:

****

**Figure 3: System architecture for Face Recognition**

By specifying the picture's location, the image is first imported. The image is then converted from RGB to grayscale since faces may be easily identified in grayscale. The photos were then subjected to image alteration, including scaling, cropping, blurring, and sharpening where necessary. The next stage is image segmentation, which divides the many items present in an image into smaller groups so that the classifier can easily identify the objects and faces in the image. Image segmentation is also used to detect contours in images. The Haar-Like features method, suggested by Voila and Jones, is the next stage in the process of identifying faces. This technique is employed to locate human faces inside a frame or picture. Some characteristics of the human face are universal, such as the nose area being brighter than the eye region and the eye region being darker than its neighbor pixels.

The haar-like approach is also used for feature selection or feature extraction i.e, given in Figure-4 for an item in an image, with the aid of edge detection, line detection, center detection, and other techniques for identifying eyes, noses, mouths, and other facial features in the image. For face detection, it is utilized to pick out the most important characteristics in a picture and extract them. Giving the coordinates of x, y, w, and h, which create a rectangular box in the image to indicate the location of the face or, as we may say, to indicate the area of interest in the picture, is the next step. In the region of interest where the face is detected, it can then draw a rectangular box.



**Figure 4: The framework of face detection process**

**4.5 Proposed Flow Chart (Facial Recognition part)**

Face recognition is the process of identifying a face that has been detected. Below is an illustration of how facial recognition works i.e. given in Figure-5:

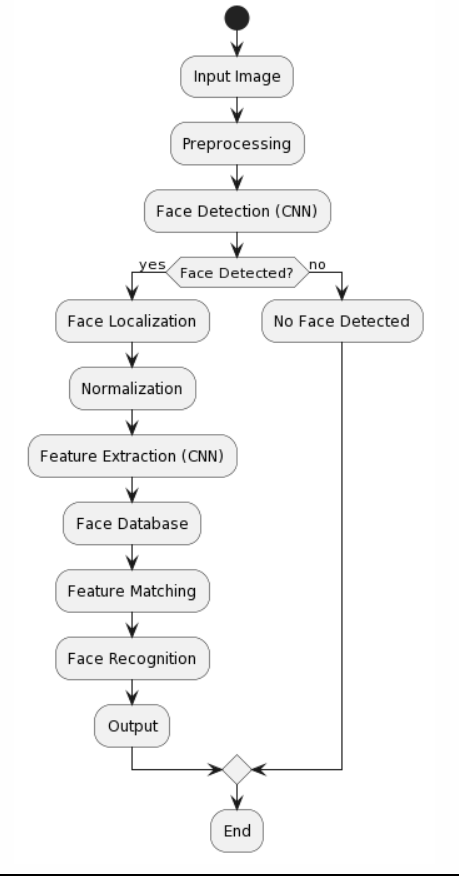


Figure 5: Flow Chart

Start with an input picture that has one or more faces in it.

Pre-processing: Take the necessary measures to improve the image's quality and get it ready for face recognition. These procedures could involve scaling the image, making it grayscale, and using normalization methods. Apply the Haar cascade classifier to the pre-processed picture to find faces using face detection. Based on Haar-like properties, this classifier is taught to recognize face features.

Face localization: Once faces have been identified, pinpoint each face's exact placement and limits in the picture. The facial region may now be isolated for additional processing thanks to this step.

Normalizations: Standardize the size and orientation of the identified facial areas. Through this method, resilience against changes in stance, size, and illumination may be attained. Feature matching involves comparing the features that were retrieved from the identified face to those that were stored in the face database. To determine how similar, the characteristics are, use the appropriate matching methods (such as cosine similarity or Euclidean distance).

Face recognition: Use a comparison of the similarity scores generated from feature matching to identify the face that was detected. Matches that are greater than a threshold is classified as known people; otherwise, they are classified as unknown people.

Output: Display the identified face along with its associated identification, or flag the face as unknown. If there are other faces in the image, repeat steps 4 and 5 for each additional face. Finish: Finish the facial recognition procedure.

**4.6 Data Collection**

Individual images are used to collect the data. We used the Python openCV for data collection. It combines the greatest elements of Python, C++, OpenCV, and API. Numerous techniques for computer vision and machine learning are supported by OpenCV. For locating the frontal faces, we used the haarcascade\_frontalface\_default.xml file. We cropped it to the appropriate size, stored it in the folder, and then used it to train the model.

Detection = cv2.CascadeClassifier(“haarcascade\_frontalface\_default.xml”)

The dataset includes images of three different individuals. Each individual is identified by a unique label or name of the individual. The dataset comprises face images of the individuals. These images are typically in a digital format, such as JPEG, and have a consistent size and resolution. The dataset may include face images captured under different conditions to account for variations in lighting, pose, expression, and occlusions (e.g., wearing glasses or hats). This variation helps make the face recognition model more robust. Our dataset is divided into training and testing sets. The training set is used to train the face recognition model, while the testing set is used to evaluate its performance. The images of each individual are distributed between the training and testing sets to ensure a fair evaluation. Each face image in the dataset is associated with a label or ID that identifies the individual it belongs to. This labelling i.e given in Figure-6 is necessary for supervised learning, where the model learns to recognize different individuals based on their labels. The dataset size can vary depending on the number of images available for each individual. Typically, for face recognition tasks, a larger dataset with a sufficient number of images per individual is desirable to train a more robust and accurate model. When using a face recognition dataset, it is important to ensure ethical considerations are addressed. This includes obtaining proper consent from individuals for using their face images, maintaining data privacy and security, and adhering to legal and ethical guidelines regarding the collection and usage of personal data.

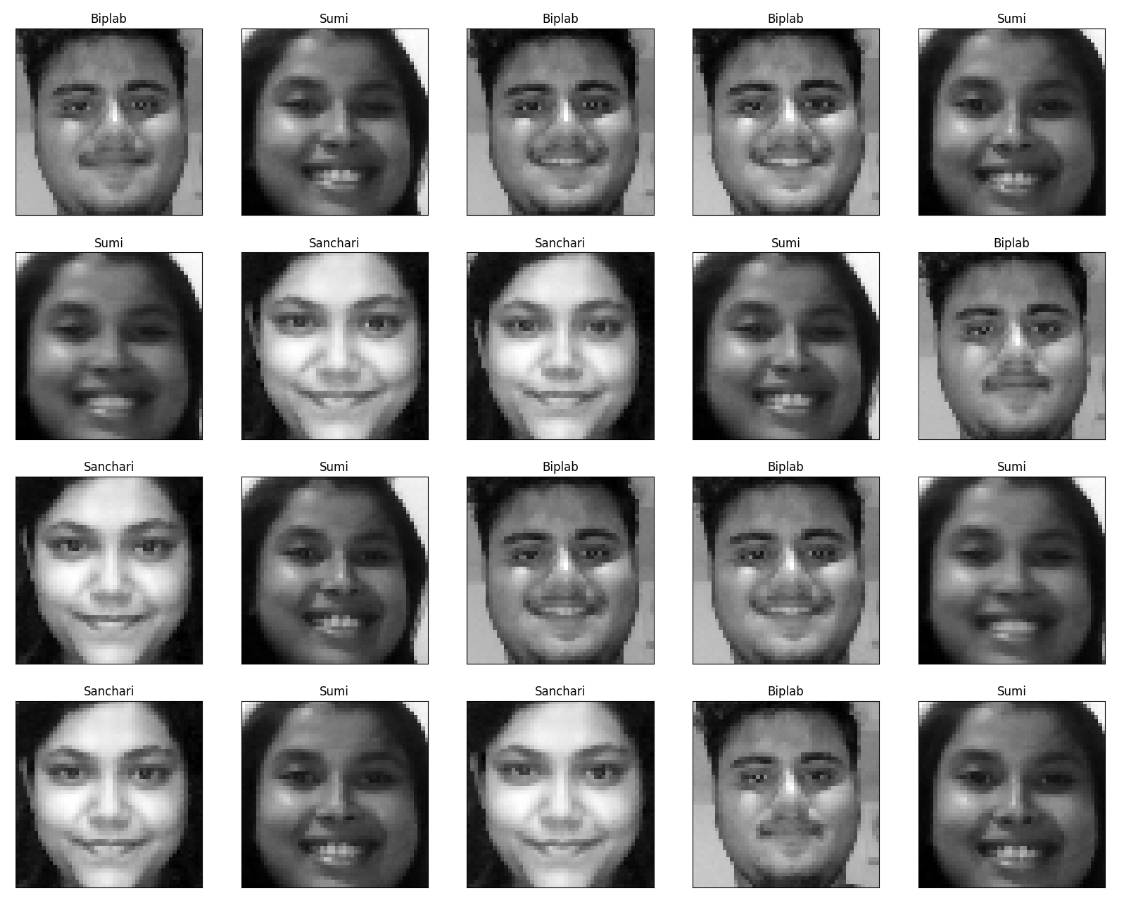


Figure 6: Building Data Set

**4.7 Data Processing**

Normalize or standardize the pixel values of the captured face images from the dataset. This process makes sure that every pixel's value falls within a defined range, which might enhance the model's convergence and performance. Typical methods include min-max scaling, which reduces pixel values to a range of 0 to 1, or z-score normalisation, which involves taking the mean and dividing it by the standard deviation. The input shape is [None, 50, 50, 1], indicating a 2D tensor with batch size, height, width, and number of channels. Which will ensure that the face images are reshaped accordingly before feeding them into the model. The dataset consist of 3000 image of 3 different persons, 1000 for each person. 80% of the dataset is extracted for training dataset to train the model and 20% of the dataset is extracted for test dataset to test the model.

**4.8 Libraries used**

**OpenCV**:

Open (Open-Source Computer Vision Library) is library used for real-time computer vision and image processing. OpenCV is written in C++ language. It was originally developed by Intel, The library is cross-platform and it has a free license which is a open-source software under Apache License 2. From 2011. GPU acceleration was featured by OpenCV for real-time operations.

**TensorFlow**:

TensorFlow is free to use and open-source software library used for machine learning and artificial intelligence. I mainly focuses on training and inference of deep neural networks. TensorFlow was developed by Google of internal use for research and production. TensorFlow is used in a wide variety of programming language, which mainly include Python, JavaScript, C++ and Java.

**TFlearn**:

TFlearn is used as a modular and transparent deep learning aspect assisting TensorFlow framework. The main task of TFlearn is to provide higher level API to TensorFlow. TFlearn is easy to understand and use. Graph visualization in TFlearn includes a various information of weights, gradients and activations.

**4.9 CNN Recognition Algorithm**

CNN (convolutional neural network) is a type of feed-forward neural network with numerous layers. CNNs are made up of filters, kernels, or neurons with programmable weights, parameters, and biases. Every filter receives certain inputs, conducts convolution, and then may or may not perform nonlinearity. Rectified Linear Unit (ReLU), Fully Connected Layers, Convolutional, Pooling, and Layers are all components of CNN's structure.

Convolutional neural networks correspond of multiple layers of artificial neurons. Artificial neurons, a rough reproduction of their natural counterparts, are mathematical functions that calculate a weighted sum of multiple inputs and give an activation value. The geste of each neuron is determined by its weight. When fed with pixel values, CNN artificial neurons prize different visual features.

When you feed an image into ConvNet, each of its layers creates multiple activation charts. Activation maps highlight important features of an image. Each neuron takes pixels as input, multiplies their color values by a weight, sums them, and runs them through an activation function.

The first( or bottom) layer of a CNN generally detects introductory features similar as vertical, perpendicular, and slant edges. The affair of the first layer is handed as input to the coming layer, which excerpts more complex features similar as corners and edge combinations. As you move deeper into a convolutional neural network, the layers begin to detect more advanced features like objects, faces, and more. Multiplying pixel values by weights and adding them together is called" convolution"( hence the name convolutional neural network). A CNN generally consists of several convolutional layers, but also includes other factors. The last layer of the CNN is the classification layer, which takes as input the output of the last convolutional layer( flash back that layers with advanced convolutions detect complex objects).

Grounded on the activation chart of the final convolutional layer, the classification layer provides a set of confidence scores( values between 0 and 1) that determine how likely an image belongs to a" class". For illustration, if you have a ConvNet that detects dogs, cats and horses, the output of the last layer is the probability that the input image contains one of these creatures.

In this project we have mainly carried out three steps of the algorithm:

1. We resized the input image at first and then the images are being passed to different layers of conv layers and pooling layers.
2. 5x5 filter is being used and the number of filter used can differ from one layer to another layer of convolution.
3. At last, we received all flattened images which is used later to recognize the face.

**5.Result Analysis**

The neural network is trained on a dataset of face images, and during inference, it generates an output vector that represents various facial features and characteristics of the input image. By comparing this output vector with a database of known individuals and their corresponding output vectors, the neural network can identify the person's name associated with the input image. To assess the accuracy and performance of the face recognition system, a validation process is typically conducted. During validation, a separate set of face images, distinct from the training data, is used to evaluate the model's performance. To evaluate the performance of the face recognition system, a validation process is conducted using a separate set of face images. The validation accuracy measures how accurately the system identifies the person's name for the validation images. In this case, the validation accuracy of the face recognition system was measured to be 1.000, indicating that it correctly identified the person's name for all the validation images. The validation loss is a metric that quantifies the difference between the predicted outputs and the true outputs during validation. A lower validation loss indicates that the model's predictions are closer to the expected outputs. The validation loss, a metric that quantifies the discrepancy between the predicted and true values, was approximately 0.00271, suggesting that the model's predictions were very close to the expected outputs during validation. These results indicate a highly accurate and reliable face recognition system. The confusion matrix helps you evaluate the performance of your face recognition model by providing insights into how well it is predicting each class. It allows you to analyse the types of errors made by the model, such as false positives and false negatives, and helps you identify areas for improvement in your classification model i.e given in Figure-7. For the purpose of real-time facial recognition, researchers will investigate methods for optimizing deep learning models. In order to do this, it is necessary to create accurate, lightweight designs with fewer parameters and less computational overhead. Real-time processing on devices with limited resources will be made possible by investigational Real-time facial recognition requires hardware acceleration, which is why it is so important. Future work will focus on creating and applying specialized hardware accelerators, such as GPUs and TPUs (Tensor Processing Units), to accelerate computing. Faster inference times and real-time performance will be made possible by this, especially in applications where latency is critical. Optimized Face Recognition Preprocessing: Face recognition relies heavily on preprocessing. Future research will concentrate on creating preprocessing methods that are optimized in order to increase the speed and precision of real-time face recognition. Real-time face identification frequently necessitates analyzing a large number of face photos or video frames in simultaneously. The potential of numerous processors or distributed systems will be explored in future study through the use of parallel and distributed computing techniques. To increase throughput and attain real-time speed, this may include parallelizing the computation over numerous GPUs or using distributed frameworks like TensorFlow or PyTorch. Streamlined Feature Extraction: Deep learning models encode facial traits into a small representation during feature extraction, a critical stage in face recognition. Future research will focus on creating simplified and effective feature extraction techniques that can instantly extract distinguishing traits from faces. To enhance the caliber of extracted data, strategies like deep metric learning and attention techniques might be investigated-design of the hardware and software is essential for achieving the best real-time performance. In the future, hardware and software developers will work together to create integrated solutions that take use of hardware optimizations, effective algorithms, and simplified implementations. To cut down on processing time, this involves maximizing memory access, using parallelism, and minimizing data migration. The development of end-to-end systems for real-time face recognition, covering all phases from face detection and feature extraction to matching and identification, will be the main emphasis of next work. In order to do this, optimized parts must be can handle input datetime and produce accurate face recognition results within a set amount of time.

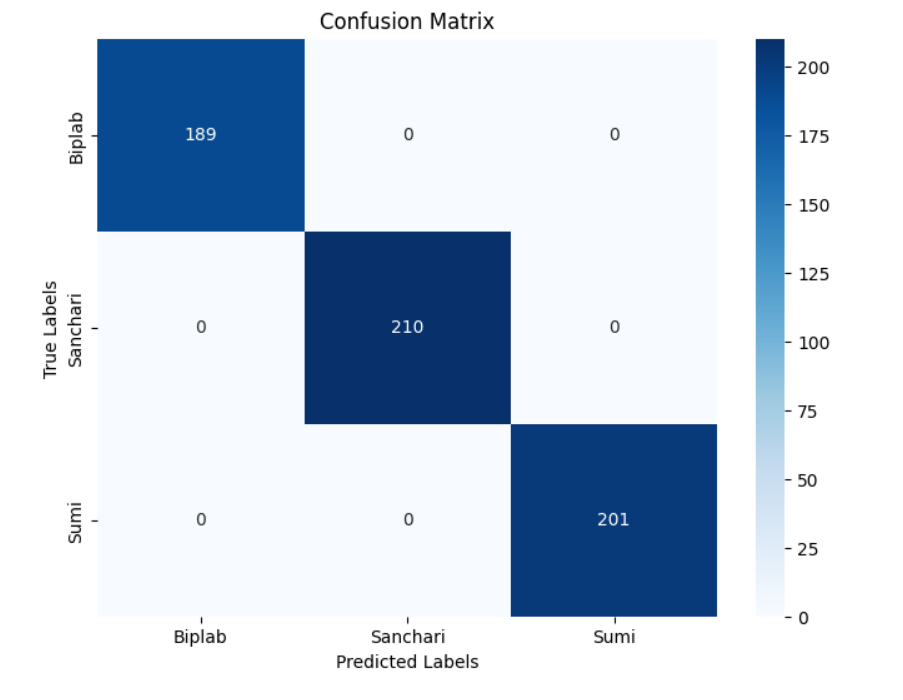


Figure 7: Confusion Matrix of Model

**6.Conclusion and Future Work**

In conclusion, the fields of computer vision and biometrics have been completely transformed by face recognition and identification utilizing deep learning techniques. Convolutional neural networks (CNNs), in particular, have demonstrated exceptional ability in reliably and robustly recognizing and identifying people from face photos or video streams. Significant advancements have been achieved in face recognition system accuracy, effectiveness, and practical application via considerable research and development. The use of deep learning in facial recognition has considerable potential. To further improve the accuracy and discriminative powers of face recognition systems, researchers will continue to investigate innovative model designs, such as advanced CNNs, attention mechanisms, and recurrent neural networks (RNNs). Innovative data augmentation approaches and big annotated datasets will be essential for training more Using deep learning algorithms for face recognition and identification, the disciplines of computer vision and biometrics have undergone a complete transformation. Convolutional neural networks (CNNs) in particular have shown extraordinary capacity in consistently and stably detecting and identifying persons from face pictures or video streams. The accuracy, usefulness, and practical applicability of face recognition systems have significantly improved thanks to extensive research and development. There is significant promise for using deep learning in facial identification. Researchers will continue to look at novel model designs, such enhanced CNNs, attention mechanisms, and recurrent neural networks (RNNs), in order to increase the precision and discriminative abilities of face recognition systems. It will be necessary to teach more people using cutting-edge data augmentation techniques and large annotated datasets. The creation of end-to-end systems and the integration of hardware-software co-design will ultimately be essential for getting the best performance in actual deployments. In order to build smooth and effective pipelines that can offer correct face recognition results under stringent time restrictions, researchers attempt to combine optimized components, from face detection and feature extraction through matching and identification. Deep learning techniques for face identification and recognition have great promise for use in a variety of fields, including surveillance, access control, authentication, and personalized services. We may anticipate that facial recognition systems will become much more precise, effective, and reliable as deep learning techniques progress, opening up a wide range of useful applications that are advantageous to society as a whole.

**5.REFERENCES**

[1] Merched, A., Xia, Y., Visvikis, S., Serot, J. M., & Siest, G. (2000). Decreased high-density lipoprotein cholesterol and serum apolipoprotein AI concentrations are highly correlated with the severity of Alzheimer’s disease☆. *Neurobiology of aging*, *21*(1), 27-30.

[2] Zuin, M., Cervellati, C., Trentini, A., Passaro, A., Rosta, V., Zimetti, F., & Zuliani, G. (2021). Association between Serum Concentrations of Apolipoprotein AI (ApoA-I) and Alzheimer’s Disease: Systematic Review and Meta-Analysis. *Diagnostics*, *11*(6), 984.

[3] Kruthika, K. R., Maheshappa, H. D., & Alzheimer's Disease Neuroimaging Initiative. (2019). CBIR system using Capsule Networks and 3D CNN for Alzheimer's disease diagnosis. *Informatics in Medicine Unlocked*, *14*, 59-68.

[4] Mohapatra, S., & Anand, K. (2021). An Expert System to Implement Symptom Analysis in Healthcare. *Integration of Cloud Computing with Internet of Things: Foundations, Analytics, and Applications*, 57-69.

[5] Feng, C., Elazab, A., Yang, P., Wang, T., Zhou, F., Hu, H., ... & Lei, B. (2019). Deep learning framework for Alzheimer’s disease diagnosis via 3D-CNN and FSBi-LSTM. *IEEE Access*, *7*, 63605-63618.

[6] AbdulAzeem, Y., Bahgat, W. M., & Badawy, M. (2021). A CNN based framework for classification of Alzheimer’s disease. *Neural Computing and Applications*, *33*(16), 10415-10428.

[7] Farooq, A., Anwar, S., Awais, M., & Rehman, S. (2017, October). A deep CNN based multi-class classification of Alzheimer's disease using MRI. In *2017 IEEE International Conference on Imaging systems and techniques (IST)* (pp. 1-6). IEEE.

[8] Islam, J., & Zhang, Y. (2019). Understanding 3D CNN behavior for Alzheimer's disease diagnosis from brain PET scan. *arXiv preprint arXiv:1912.04563*.

[9] Khagi, B., & Kwon, G. R. (2020). 3D CNN design for the classification of Alzheimer’s disease using brain MRI and PET. *IEEE Access*, *8*, 217830-217847.

[10] Mohapatra, Subhasish, Komal Paul, and Abhishek Roy. "Object-oriented modeling of cloud healthcare system through connected environment." *Applications of Internet of Things: Proceedings of ICCCIOT 2020*. Springer Singapore, 2021.

[11] Mohapatra, S., Satpathy, S., & Paul, D. (2021). Data-Driven Symptom Analysis and Location Prediction Model for Clinical Health Data Processing and Knowledgebase Development for COVID-19. In *Applications of Artificial Intelligence in COVID-19* (pp. 99-117). Springer, Singapore.

[12] Lehrner, J., Kogler, S., Lamm, C., Moser, D., Klug, S., Pusswald, G., ... & Auff, E. (2015). Awareness of memory deficits in subjective cognitive decline, mild cognitive impairment, Alzheimer's disease and Parkinson's disease. *International Psychogeriatrics*, *27*(3), 357-366.

[13] Essemlali, A., St-Onge, E., Descoteaux, M., & Jodoin, P. M. (2020, September). Understanding Alzheimer disease’s structural connectivity through explainable AI. In *Medical Imaging with Deep Learning* (pp. 217-229). PMLR.

[14] Mohapatra, Subhasish, Suneeta Satpathy, and Sachi Nandan Mohanty. "A comparative knowledge base development for cancerous cell detection based on deep learning and fuzzy computer vision approach." *Multimedia Tools and Applications* 81.17 (2022): 24799-24814.

[15] Ahmad, I., & Pothuganti, K. (2020). Analysis of different convolution neural network models to diagnose Alzheimer’s disease. *Materials Today: Proceedings*.

[16] Mohapatra, S., Satpathy, S., & Mohanty, S. N. (2022). A comparative knowledge base development for cancerous cell detection based on deep learning and fuzzy computer vision approach. *Multimedia Tools and Applications*, 1-16..

[17] Liu, L., Zhao, S., Chen, H., & Wang, A. (2020). A new machine learning method for identifying Alzheimer's disease. *Simulation Modelling Practice and Theory*, *99*, 102023.

[18] Mohapatra, Subhasish, and Smita Parija. "A brief understanding of IOT health care service model over remotely cloud connected environment." *Advances in Intelligent Computing and Communication: Proceedings of ICAC 2019*. Springer Singapore, 2020.

[19] Umeda-Kameyama, Y., Kameyama, M., Tanaka, T., Son, B. K., Kojima, T., Fukasawa, M., ... & Akishita, M. (2021). Screening of Alzheimer’s disease by facial complexion using artificial intelligence. *Aging (Albany NY)*, *13*(2), 1765.

[20] Kavitha, C., Mani, V., Srividhya, S. R., Khalaf, O. I., & Romero, C. A. T. (2022). Early-Stage Alzheimer's Disease Prediction Using Machine Learning Models. *Frontiers in Public Health*, *10*.

[21] Mohapatra, S., & Anand, K. (2020). A Brief Model Overview of Personalized Recommendation to Citizens in the Health‐Care Industry. *Recommender System with Machine Learning and Artificial Intelligence: Practical Tools and Applications in Medical, Agricultural and Other Industries*, 27-44.