







Deep Learning Based Face Recognition System for Automated Identification

Prashant Ahlawat¹ , Navpreet Kaur¹, Charnpreet Kaur¹ , Santosh Kumar¹,
and Hitesh Kumar Sharma²  

¹ Computer Science Engineering, University Institute of Engineering, Chandigarh University,
Gharuan, Mohali, Punjab, India

² School of Computer Science, University of Petroleum and Energy Studies (UPES), Energy
Acres, Bidholi, Dehradun, Uttarakhand 248007, India

hkshitesh@gmail.com

Abstract. Face recognition is a fundamental task in computer vision with numerous applications in various domains, including surveillance, security, access control, and human-computer interaction. In recent years, deep learning has revolutionized the field of face recognition by significantly improving accuracy and robustness. This abstract presents an overview of a deep learning-based face recognition system developed for automated identification purposes. The proposed system leverages convolutional neural networks (CNNs) to extract discriminative features from facial images and a classification model to match and identify individuals. The system comprises three main stages: face detection, feature extraction, and identification. In the face detection stage, a pre-trained CNN model is employed to locate and localize faces within input images or video frames accurately. The detected faces are then aligned and normalized to account for variations in pose, scale, and illumination. Next, a deep CNN-based feature extraction network is utilized to capture high-level representations from the aligned face regions. This network learns hierarchical features that are robust to variations in facial appearance, such as expression, occlusion, and aging. The extracted features are typically represented as a compact and discriminative embedding vector, facilitating efficient and accurate face matching. To enhance the system's performance, various techniques can be employed, such as data augmentation, model fine-tuning, and face verification to handle challenging scenarios, including pose variations, illumination changes, and partial occlusions. The proposed deep learning-based face recognition system has shown remarkable accuracy and robustness in automated identification tasks. It has the potential to be deployed in real-world applications, including surveillance systems, access control in secure facilities, and personalized user experiences in human-computer interaction.

Keywords: Face Recognition · Machine Learning · Attendance System · BPH

1 Introduction

A sheet of paper is typically used to record attendance, along with any other comments that may be required, and on which the student's name and any other pertinent information is written. Each student will receive a copy of this piece of paper, which will serve

as their registration. As an illustration, the lecturer might fill out the date on the registration form before passing the paper around to each student in the group. The lecturer occasionally gave out papers before the session started, to make the most of everyone's time and the most efficient use of the speaker's time. Fig. 1 shows the face recognition system.

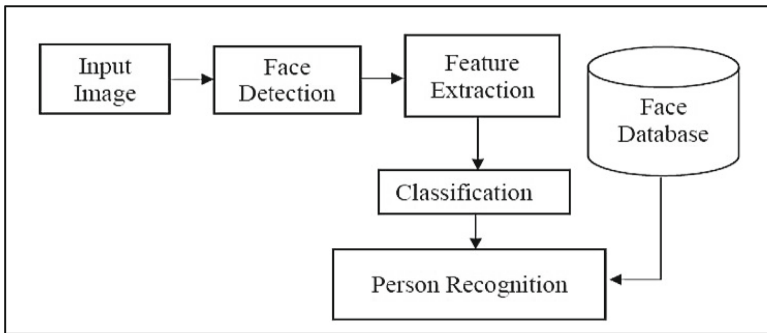


Fig. 1. Face recognition system

The two methods that have been thus far discussed have both been used on a sizable scale for a sizable amount of time. On the other hand, problems frequently arise when the typical or traditional technique of recording attendance is applied. They could take a long time, particularly when pupils are expected to respond when the speaker calls their names. It will be difficult to keep the attendance report up to date until it is time to write the report since it is conceivable that the registration form or the attendance sheet have been misplaced. While using the conventional approach, lecturers or teachers must accurately record the attendance of each student by marking their presence on the registration sheet or attendance form.

2 Literature Review

Authentication is a computer-based type of communication that serves a very important role in preserving system control [1]. Face recognition is currently utilised in a wide range of applications and has grown to be a crucial part of biometric authentication. Examples of these applications include network security, human-computer interfaces, and video monitoring systems. [2] The requirement that students attend a physical educational facility is the element of the virtual platform that presents the greatest challenge for pupils. The technique takes a lot of time and effort to complete by hand. [3] One of the most useful applications of image processing is the recognition of faces, which is crucial in the technology world. Recognizing human faces is a current issue that needs to be addressed for the sake of authentication, particularly in the context of taking student attendance. [4] Educational institutions are concerned about the consistency of their students' performance in the modern environment. One element influencing the general fall in pupils' academic performance is the low attendance rate. Your attendance may

be recorded using a variety of techniques, the two most common being signing in and having pupils raise their hands. The process took longer and presented several limitations and challenges.

3 Methodology

The rising desire for higher levels of security as well as the quick uptake of mobile devices may be to blame for the spike in interest in facial recognition research over the past several years. Whatever the case, field researchers have warmly welcomed this excitement. In any case, the unexpected increase in interest has been a really fortunate development. Face recognition technology has applications in many fields, including but not limited to access control, identity verification, surveillance, security, and social media networks. In the computer vision subfield of face detection, the OpenCV approach is a well-liked option. The AdaBoost approach is then employed as the face detector after having previously been used to extract the feature photos into a sizable sample set by first extracting the face Haar features included within the image [5]. This happens following its use to extract the feature images from a sizable sample set. Here, the goal is to teach a deep neural network how to recognize people by their faces and output their faces. In order for the neural network to automatically recognise the many elements of a face and produce numbers based on those features, it appears that the neural network must be trained. If you feed in multiple photographs of the same person, the neural network's output will be very similar or close, however if you pass in multiple images of different persons, the output will be considerably different. The neural network's output can be thought of as an identifier for a certain person's face. The output of the neural network can be compared to a face-specific identification number. The Local Binary Patterns Histogram method was put out as a potential fix for the issue in 2006. It is constructed on top of the framework offered by the neighbourhood binary operator. Due to how easily it can be calculated and how powerfully it can select, it is used extensively in the field of face recognition. The following actions must be taken in order to reach this objective: (i) the creation of datasets; (ii) the collecting of faces; (iii) the recognition of distinctive traits; and (iv) the classification of the faces.

Figs. 2 and 3 show the algorithm and data flow process.

A dialogue window that asks for the user id and name, in that order, appears on the screen during the initial run of the application. The following step is selecting the "Take Images" option by clicking the pertinent button after entering the necessary data in the "name" and "id" text boxes. The working computer's camera will open when you select the Take Images option from the drop-down menu, and it will begin capturing pictures of the subject. The file with this ID and Name's name is Student Details.csv, and both of them are kept in a folder called Student Details. The Documents area of the navigation bar contains the Student Details folder.

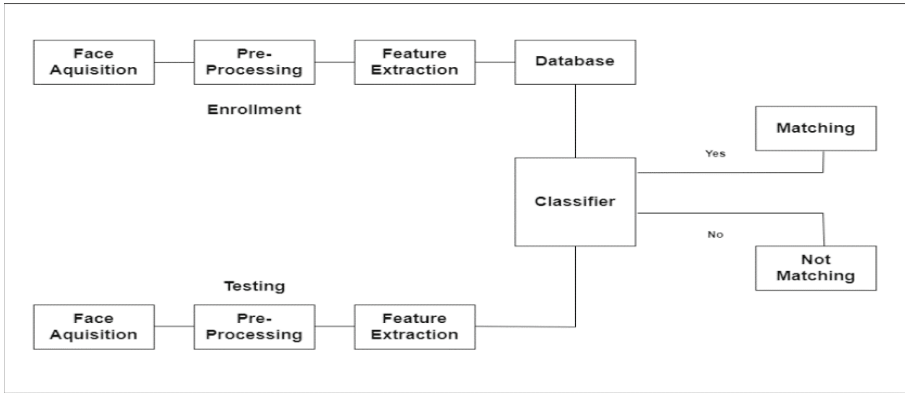


Fig. 2. Algorithm and Data flow process.

These sixty photographs are kept in a folder called Training Image and are used as an example. After the procedure has been properly finished, a notification that the photographs have been saved will be obtained once the process is complete. Once several picture samples are collected, the Train Image button is selected in order to train the images that have been collected. Now it only takes a few seconds to teach the computer to recognise the images, which also results in the creation of a Trainer.yml file and the storage of the images in the Training Image Label folder. Each of the initial parameters has now been fully implemented as of this writing. The following step is to choose the Track images after choosing Take pictures and Train pictures.

4 Dataset Description

For this work, the Facial Images Dataset available at [<https://www.kaggle.com/datasets/apollo2506/facial-recognition-dataset>] is used. It consists of 3950 images with different facial expressions. The dataset consists of basically 5 categories of facial expression [10]. The dataset taken for this research work is publicly available facial expression dataset.

The description of the image dataset and number of images in each class has given in Table 1 (Fig. 4).

4.1 Dataset Standardization

An image generator is created to standardize the input images. It will be used to adjust the images in such a way that the mean of pixel intensity will be zero and standard deviation will become 1. The old pixel value of an image will be replaced by new values calculated using following formula. In this formula each value will subtract mean and divide the result with standard deviation [6].

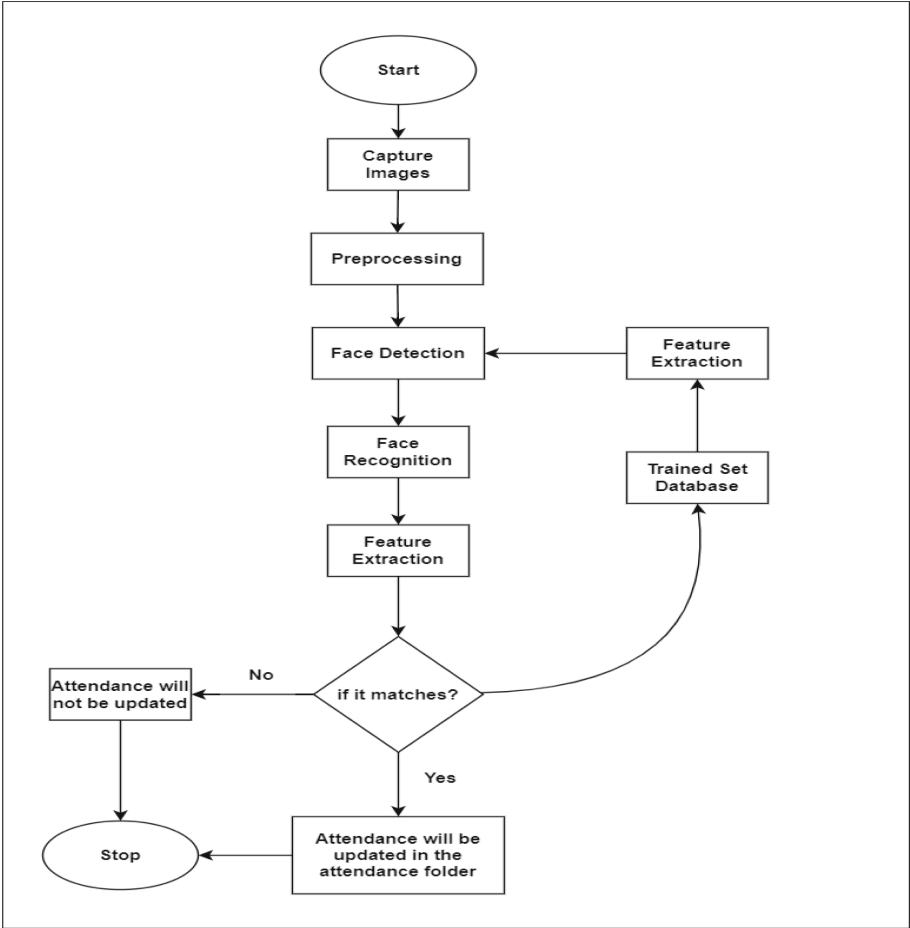


Fig. 3. Flowchart of the proposed system.

Table 1. Facial Image Dataset Description [<https://www.kaggle.com/datasets/apollo2506/facial-recognition-dataset>]

Facial Expression Class	Images
Angry	540
Sad	890
Happy	770
Neutral	760
Surprise	990

4.2 Class Imbalance in Image Dataset M

The major challenge in plant disease or medical image dataset is class imbalance problem. There are not equal numbers of images in all classes of these datasets. In such case, biasness will be for the class which has a greater number of images [7]. The frequency of each class or label has been plotted in Fig. 5.

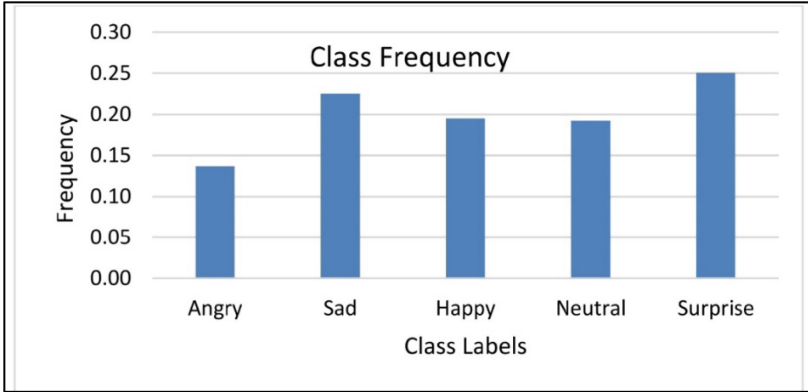


Fig. 5. Class frequency plot

It can be seen in Fig. 5 that, there is a large variation in number of images in each class. This data imbalance generates a biasness in the model and reduce the performance of model. To overcome this data imbalance challenge the following mathematical formulation is applied which is explained in detail in following subsection [11, 13].

4.3 Impact of Class Imbalance on Loss Function

The cross-entropy loss formula is given in Eq. (1).

$$L_{cross\ entropy}(X_i) = -(Y_i \text{Log}(f(X_i)) + (1 - Y_i) \text{Log}(1 - f(X_i))) \quad (1)$$

X_i and Y_i denotes the input feature and corresponding label, $f(X_i)$ represents the output of the model. Which is probability that output is positive. This formula is written for overall average cross entropy loss for complete dataset D of size N and it is given below in Eq. (2).

$$L_{cross\ entropy}(D) = -\frac{1}{N} \left(\sum_{\text{positive}} \text{Log}(f(X_i)) + \sum_{\text{Negative}} \text{Log}(1 - f(X_i)) \right) \quad (2)$$

This formulation shows clearly that the loss will be dominated by negative labels if there is large imbalance in dataset with very few frequencies of positive labels. Fig. 6 shows the positive and negative class frequency of the five given classes before applying weight factor [12].

$$Freq_{\text{positive}} = \frac{\text{No. of Positive Samples}}{N} \quad (3)$$

$$Freq_{negative} = \frac{No. of Negative Samples}{N} \quad (4)$$

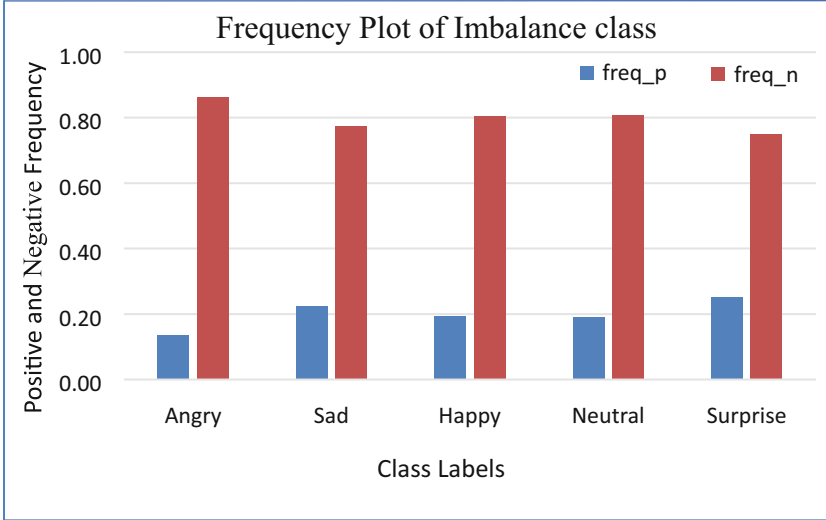


Fig. 6. Positive and Negative Class Frequency of 5 given Classes before Applying Weight Factor.

As shown in Fig. 6, contribution of positive labels is less than the negative labels. However, for accurate results this contribution should be equal. The one possible way to make equal contribution is to multiply each class frequency with a class specific weight factor $W_{positive}$ and $W_{negative}$, so each class will contribute equally in classification model [8, 9].

So, the formulation will be represented as,

$$W_{positive} \times Freq_{positive} = W_{negative} \times Freq_{negative} \quad (5)$$

which can be done simply by taking,

$$W_{positive} = Freq_{negative}$$

$$W_{negative} = Freq_{positive}$$

Using the above formulation, the class imbalance problem is dealt with. To verify the formulation, the frequency graph is plotted and it shows the expected chart. It can be seen in Fig. 5 that both frequencies are balanced. (Table 2)

As shown in Fig. 7, after applying weight factor to positive and negative class the contribution to loss function is equally distributed. The calculation of loss function is given in Eq. (6).

$$L_{cross\ entropy}^W(X) = -(W_{positive} Y \log(f(X)) + W_{negative} (1 - Y) \log(1 - f(X))) \quad (6)$$

This adjustment of loss function will eliminate class imbalance problem and biasness problem with imbalance dataset [14, 15].

Table 2. Computational table for data imbalance.

Class Labels	Total Images	freq_p	freq_n	w _p	w _n	freq_p	freq_n
Angry	540	0.14	0.86	0.86	0.14	0.12	0.12
Sad	890	0.23	0.77	0.77	0.23	0.17	0.17
Happy	770	0.19	0.81	0.81	0.19	0.16	0.16
Neutral	760	0.19	0.81	0.81	0.19	0.16	0.16
Surprise	990	0.25	0.75	0.75	0.25	0.19	0.19

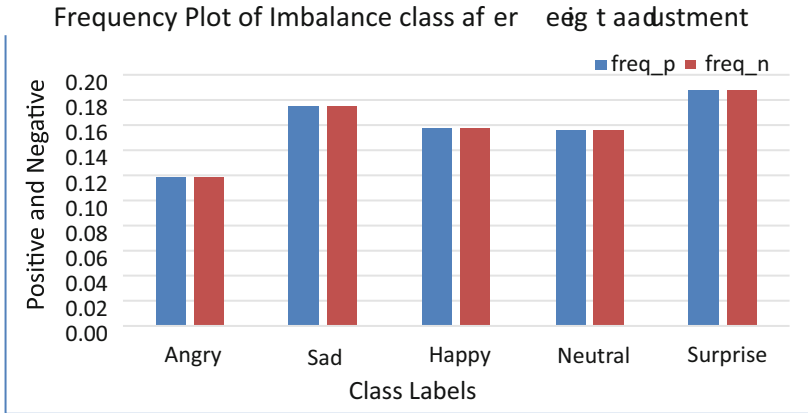


Fig. 7. Positive and negative class frequency plot of 5 given classes after applying weight factor.

5 Model Methodology

The proposed face recognition framework is carried out by building up a convolutional neural organization which is a productive model for picture characterization. CNN models are prepared by first taking care of a bunch of face pictures alongside their marks. These pictures are then gone through a pile of layers including convolutional, ReLU, pooling, and completely associated layers. These pictures are taken as bunches. In the proposed framework, a group size of 64 was given. The model was prepared to utilize 50 epochs. At first, the model concentrates little highlights, and as the preparation interaction advances more definite highlights will be separated. The greater part of the pre-processing is done consequently which is one of the significant benefits of CNN. Notwithstanding that information pictures were resized. Expansion is additionally applied which builds the size of the dataset by applying activities like turn, shear, and so forth. During the preparation interaction, the model finds highlights and designs and learns them. This information is then used later to discover the expression of face when another face expression picture is given as info. Unmitigated cross-entropy is utilized as misfortune work [16]. At first, the misfortune esteems would be extremely high however as the interaction progresses the misfortune work is diminished by changing the weight esteems.

The layer configuration of CNN model is shown in Table 3.

Table 3. Layer specifications of our CNN model

Proposed CNN Model specification	
Layers	Layer specifications
Optimizer	Adam
Loss	categorical_crossentropy
Metrics	['accuracy']
Conv2D Layer (1)	64 filters, 3 × 3 Filter size and ReLU Activation
Max-Pooling2d Layer	2 × 2 kernel size
Dropout	20%
Conv2D Layer (2)	128 filters, 3 × 3 Filter Size and ReLU Activation
Max-Pooling2d Layer	2 × 2 kernel
Dropout	30%
Conv2D Layer (3)	128 filters, 3 × 3 Filter Size and ReLU Activation
Max-Pooling2d Layer	2 × 2 kernel
Dropout	30%
Conv2D Layer (4)	256 filters, 3 × 3 Filter size and ReLU Activation
Max-Pooling2d Layer	2 × 2 kernel size
Dropout	20%
Conv2D Layer (5)	512 filters, 3 × 3 Filter size and ReLU Activation
Max-Pooling2d Layer	2 × 2 kernel size
Dropout	30%
Flatten layer	4608 Neurons
Dense Layer	1024 Neurons
Batch Normalization	Relu Activation
Dropout	50%
Dense Layer	512 Neurons
Batch Normalization	Relu Activation
Dropout	25%
Output layer	Softmax 5 classes

The proposed model contains

Total params: 10,101,765

Trainable params: 10,097,541

Non-trainable params: 4,224

6 Experimental Results

The heat map of confusion matrix is given in Fig. 8.

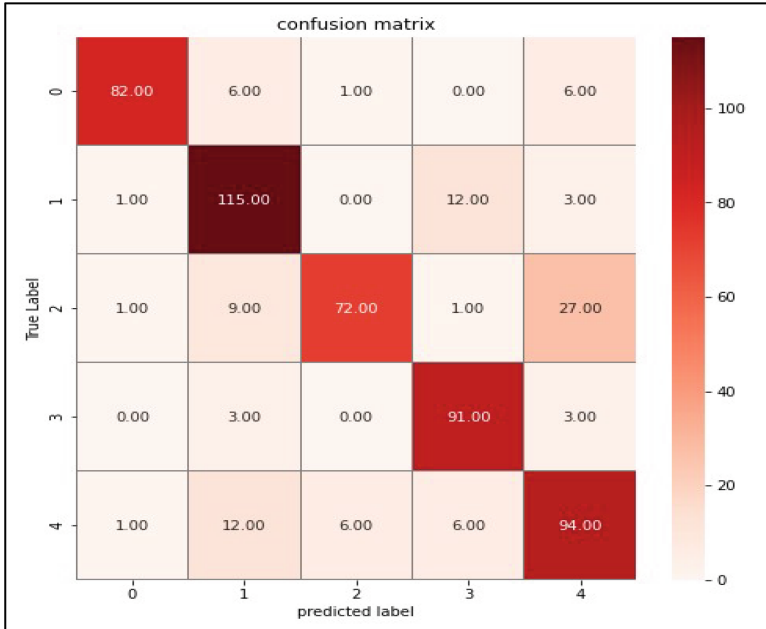
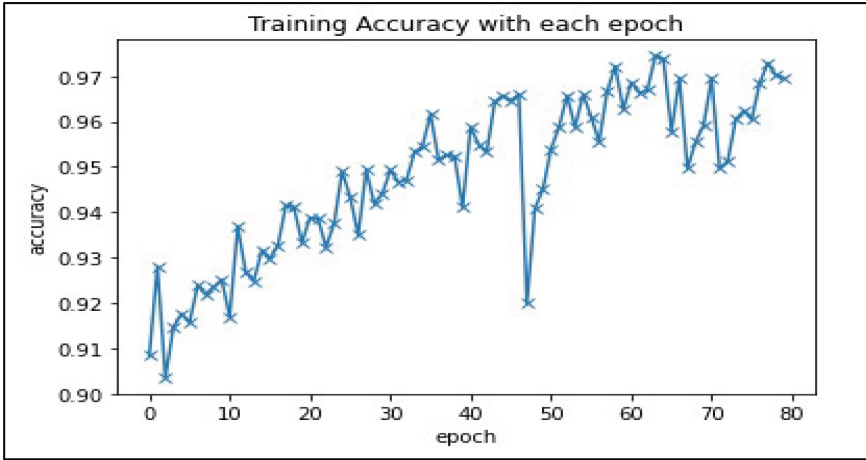
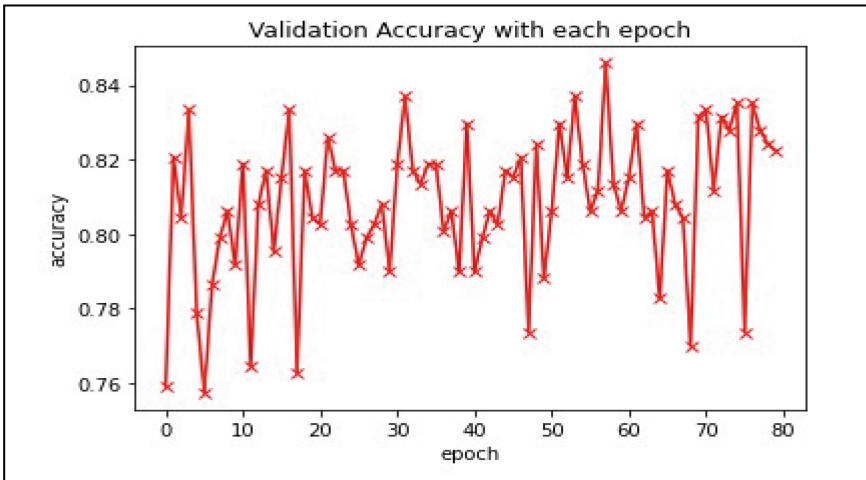


Fig. 8. Heat map of Confusion matrix for our proposed model

The results of training and validation accuracy are shown in Fig. 9(a) and (b). Accuracy rate exceeded 80% limit while in the training phase. Similarly, validation accuracy is around 80%. For this data set the result is appreciable and it can easily exceed more in more epochs.



(a)



(b)

Fig. 9. Training and Validation Accuracy of Model.

7 Conclusion

Hence, in order to track the children's whereabouts, this work created a procedure for the management and administration of attendance. Particularly in situations where a sizable portion of students have previously recorded their attendance, it helps to reduce the amount of time and effort needed. Python is the programming language that is utilised to implement the system from start to finish. Face recognition techniques were employed to accurately record the children's attendance. This list of students' attendance may also

be used for other purposes, such as to identify who is and is not present for exams in order to resolve examination-related issues.

References

1. Chen, Y., Jiang, H., Li, C., Jia, X.: Profound component extraction and characterization of Hyperspectral pictures dependent on Convolutional Neural Network (CNN). *IEEE Trans. Geosci. Remote Sens.* **54**(10), 62326251 (2016)
2. Chuang, M.-C., Hwang, J.-N., Williams, K.: A feature learning and object recognition framework for underwater fish images. *IEEE Trans. Image Process.* **25**(4), 1862–1872 (2016)
3. Chuang, M.-C., Hwang, J.-N., Williams, K.: Regulated and unsupervised highlight extraction methods for underwater fish species recognition. In: *IEEE Conference Distributions*, pp. 33–40 (2014)
4. Kim, H., et al.: Image-based monitoring of jellyfish using deep learning architecture. *IEEE Sensors J.* **16**(8), 2215–2216 (2016)
5. Silva, C., Welfer, D., Gioda, F.P., Dornelles, C.: Cattle brand recognition using convolutional neural network and support vector machines. *IEEE Latin Am. Trans.* **15**(2), 310–316 (2017)
6. Sharma, H.K.: E-COCOMO: the extended cost constructive model for cleanroom software engineering. *Database Syst. J.* **4**(4), 3–11 (2013)
7. Sharma, H.K., Kumar, S., Dubey, S., Gupta, P.: Auto-selection and management of dynamic SGA parameters in RDBMS. In: *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1763–1768. *IEEE* (2015)
8. Khanchi, I., Ahmed, E., Sharma, H.K.: Automated framework for Realtime sentiment analysis. In *5th International Conference on Next Generation Computing Technologies (NGCT-2019)* (2020)
9. Agarwal, A., Gupta, S., Choudhury, T.: Continuous and Integrated Software Development using DevOps. In: *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, pp. 290–293 (2018). <https://doi.org/10.1109/ICACCE.2018.8458052>
10. Chen, C., Zhong, Z., Yang, C., Cheng, Y., Deng, W.: Deep learning for face recognition: a comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* (2022). <https://doi.org/10.1109/TPAMI.2022.3056791>
11. Duan, Z., Zhang, X., Li, M.: FSA-net: a light-weight feature shrinking and aggregating network for real-time face recognition. *Pattern Recogn. Lett.* **154**, 129–136 (2022)
12. Jin, Y., Xu, Y., Dong, S., Yan, J.: DSFD: dual shot face detector. *IEEE Trans. Pattern Anal. Mach. Intell.* (2022). <https://doi.org/10.1109/TPAMI.2021.3082544>
13. Cai, Y., Guo, H., Zhang, Z., Wang, L.: Deep dual attention networks for face recognition. *Pattern Recogn.* **125**, 108308 (2022)
14. Cheng, Y., Chen, C., Deng, W.: Learning spatial-aware and occlusion-robust representations for occluded face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* (2021). <https://doi.org/10.1109/TPAMI.2021.3081622>
15. Li, X., Liu, Z., Chen, J., Lu, H.: Stacked dense network with attentive aggregation for robust face recognition. *IEEE Trans. Cybern.* 1–14 (2021)
16. Zhang, W., Liu, H., Zhu, S., Zhang, S., Zhang, Y.: DPSG-Net: deep pose-sensitive graph convolutional network for face alignment. *Pattern Recogn. Lett.* **153**, 52–59 (2021)