Face Recognition Attendance System Using Deep Learning

Prof. Gajanan Walunjkar[1], Dr. Rahul Desai[2] & Bhavesh Singh Bisht[3]

[1]Assistant Professor, Department of Information Technology, Army Institute of Technology, Pune
[2]Associate Professor, Department of Information Technology, Army Institute of Technology, Pune
[3]Alumni, Army Institute of Technology, Pune

Abstract*: *In this era, with the breakthrough of machine and deep learning, we have developed assorted mechanisms for face detection and recognition using deep learning. However, we have witnessed that taking attendance that is an integral part of every educational institution, and work organization is still done manually. Attendance when taken manually or using biometric is a tedious and cumbersome process. The system described in this paper can effectively replace traditional manual attendance methods and can save time and logistics. The proposed system contains multiple stages for the whole attendance journey where it starts with the enrollment of all the students at once. Then using a pre-trained model embeddings are extracted from all the faces then a fully connected network on top of these embedding with softmax function for multiclass classification into respective classes. The whole system is real-time and is masked behind a simple user interface developed using python. The classification accuracy achieved is ~85% provided with good lighting conditions. The future scope of this system is to be deployed end to end on Amazon Web Services that would eventually eliminate the manual process of managing the databases and code using fully managed services.

Keywords*: *facial recognition, face detection, deep learning, machine learning, OpenCV, Keras, Tensorflow, unstructured data, image data, regularization

I. INTRODUCTION

The entire system is divided into multiple stages. Typically fully connected neural networks are not directly applied to image data; computational cost is one the major reason. Therefore, the image dataset is to be converted into a feature vector that can be handled by the fully connected layers. In this architecture, the face embeddings are converted to 128 feature vectors using a face-net Inception-Resent model trained on the MS-Celeb-1M dataset and then these embeddings are fed into Keras Dense Layers. The model is Attendance marked; student enrollment is stored into a MongoDB database hosted locally. For now, we have created a system for two lectures English and Hindi that can be extended further by making required changes in the UI and the database.

The overall system is broken down into the following stages:

A. Person Enrollment

This is the first phase of the proposed solution where all the students are enrolled for the classes/lectures. This is similar to enrolling the students in a school. In the current scenario, all the students are enrolled at once in all the lectures taking the assumption that all subjects are compulsory for all the students. In this phase, the images are enrolled using the UI. Currently, we are taking 50 images per student that are be reduced or increased as per the requirement. The faces are detected in real-time using Haar feature-based cascade classifiers [1,2] The faces are cropped from the images and stored in the name folders that are created using scripts based on the input name from the UI. [3] The faces are stored incrementally from 1 to 50.

B. Facial features extraction using FaceNet Inception-ResNet

Feature extraction is an integral part of this whole system. If the fully connected network is fed with all the pixels of the input images then, it would be highly unstable, would require a lot of images, and computationally expensive. In this system, we are using the KerasFaceNet model. The face-net system is used to extract high-quality features called embeddings from the faces. The model is an Inception-ResnetV1 model trained on the 1 M celebrity dataset. The total embeddings extracted out the faces are 64 (Px, Py) and a 128 vector is formed after flatten [4].

C. Artificial Neural Networks for Training

The faceNet model outputs a 128 embedding vector for the input image using model predict in Keras [5]. This vector is flattened and fed to a fully connected neural network with 3 hidden layers and a softmax activation function in the output layer. The model predicts the score for all the classes and the greatest score above a threshold of 0.85 is mapped to that respective class(here the output classes are the students)

D. MongoDB database

A MongoDB database hosted locally is updated with the attendance of the students present in the particular lecture. A database named “Student” is created locally during the enrollment and two collections namely “English” and “Hindi” are created inside it. The teacher/user is required to select the lecture for which the attendance needs to be marked. The

collection is mapped to the UI using local variables that are passed as a parameter in the Recognizer file.

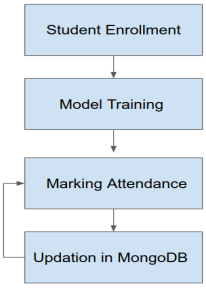

Fig 1. High-level workflow of the proposed system

II. ALGORITHM

The Following are the two major part of the system:

1) Face Detection
2) Face Recognition

A. Face Detection

Detecting the face out of the image is one of the important aspects of the face recognition journey. Cropping out the face helps to get rid of the extra pixels (noise) from the images. Face detection takes place in two stages in our system. The first stage is the enrollment phase, in which the students are enrolled in the database and their cropped images are saved in folders named with their name. These images are the training data-set for our Neural Network. Some parts of these images are also used as cross-validation set for model evaluation. The face localization from the image is done using Haar based cascade classifiers [6-7]. The input for training for these classifiers is positive and negative face images. Following are the three major type of feature kernels (windows) that are used to detect the faces:

1) Edge features
2) Line features
3) Four-Rectangle features

The cascade classifier is an ensemble of weak classifiers. There are well-defined stages that detect the region either positive or negative. If the region is detected as positive it is passed to later stages and if the region is tested negative the window moves to the next position based on a defined stride[8]. The aim is to discard negative positions in the image with the assumption that the majority of the part doesn't contain objects of interest. This is achieved using the Adaboost. In the proposed system, a rectangular bounding box is drawn around the face using the CV2 library. The further process is done on the extracted face coordinates from the bounding box.

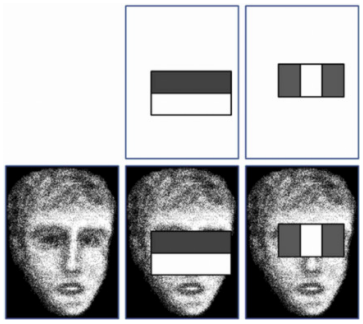

Fig 2. Cascade kernels sliding over images to detect face features

B. Face Recognition

Face Recognition is done in two phases:

1. Embeddings generator
2. Fully Connected Network

1) Embedding Generator

Feeding fully connected layers with all the image pixels is impractical and computationally expensive. Therefore, embeddings (features) are extracted from the faces before feeding to fully connected layers. Using Convolution and Pooling layers not only reduce the image dimensions but also extract required features. These features can be extracted using various CNN architectures like AlexNets, Inception Nets, etc. FaceNet provides a unique architecture for performing tasks like face recognition, verification, and clustering. It uses deep convolutional networks along with triplet loss to achieve a state of the art accuracy. In our system, we are using Inception-ResNet Model. The input dimension is 160*160*3 and the output

dimension is 64*2.
The Inception-ResNet Model is a combination of Google's Inception Net mode with skip connections or Residual blocks. The Inception model uses three stacked convolution filters at a single layer (1*1, 5*5, and 3*3) and the output of these layers are added and fed to the next identity block. The inception model uses a 1*1 filter to reduce the channel size to reduce the number of parameters.

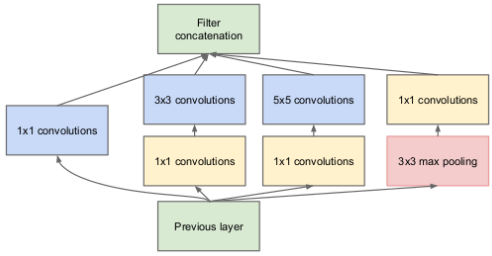

Fig 3. Inception Net Architecture

The Residual blocks use skip connection that helps to learn complex relationships without hampering the performance of the model. Using skip connections also eliminates the problem of vanishing gradient and helps to train deep neural networks[9-10].

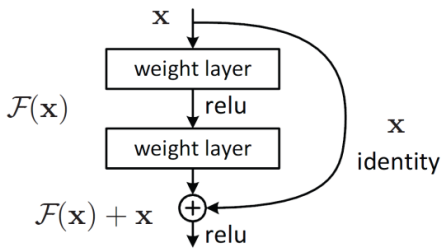

Fig 4.Skip Connection/residual block

In the proposed solution, the facnet model that is used to extract the embeddings using the Inception model with skip connections that can be illustrated in the diagram below:

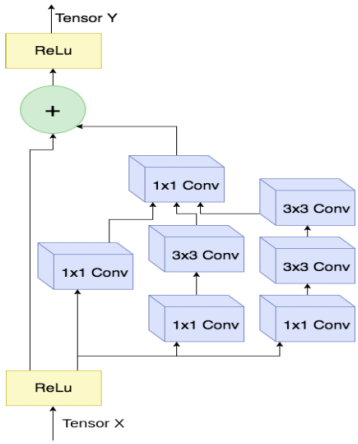

Fig 5.Identity block of Inception-ResNet model

The model inputs 160*160 with a 3 channel dimension input image and output 128 embedding vectors. Several Identity blocks are connected to form deep neural networks of several layers. The activation function used in the Inception-Resnet model is ReLU [11]. In the later layers, several techniques of regularization such as dropout are used to prevent overfitting with Batch normalization that helps to standardize the input to the next layer for every mini-batch and thus reduces the epochs required to train the network.

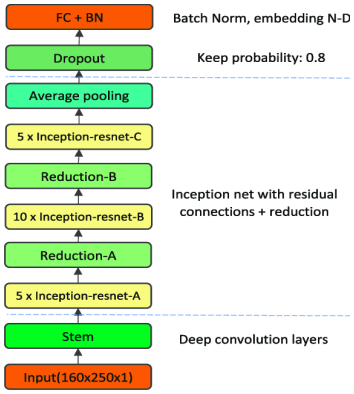

Fig 6.Workflow of the Inception-ResNet model

These 128 embedding are flattened and fed to a fully connected network. The activation function we have used here is leakyReLU with alpha 0.1. The fully connected network contains 3 hidden layers with dimensions 62, 32, and 16 neurons in each layer. The softmax function is used in the output layer with Adam optimizer and loss function as categorical_crossentropy.

```
def arch(self):
    self.model.add(Dense(128,input_dim=128))
    self.model.add(LeakyReLU(alpha=0.1))
    self.model.add(Dense(64))
    self.model.add(LeakyReLU(alpha=0.1))
    self.model.add(Dense(32))
    self.model.add(LeakyReLU(alpha=0.1))
    self.model.add(Dense(16))
    self.model.add(LeakyReLU(alpha=0.1))
    self.model.add(Dense(self.classes))
    self.model.add(Activation('softmax'))
```

Fig 7.Architecture of fully connected layers

III. USER INTERFACE

The user interface is created using the Tkinter library in python. The UI consists of several Labels, buttons, progress bars, radio buttons, tree view, scrollbars, date, etc. The whole attendance system can be divided into the following steps from an end-user perspective.

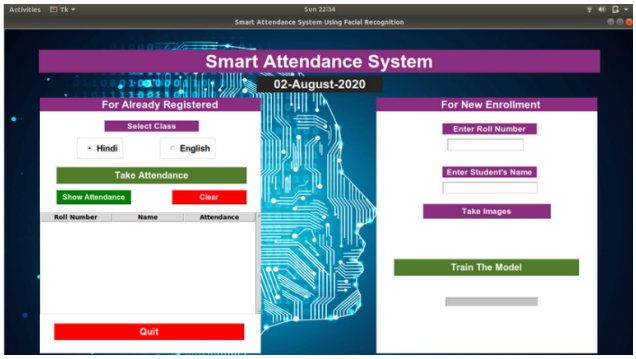

Fig 8. UI of the system

A. Students Enrollment:

This is a one-time process where students are enrolled one by one by capturing their faces in the CV2 window. The python script automatically takes 50 photos of a student in a couple of seconds and stores them in different folders. In this step, the students are also enrolled in the MongoDB database with 0 attendances. For more convenience a CSV file is also created that contains the input information entered in the UI and if the students are enrolled successfully. Two parameters i.e.Roll number, Name are required fields.

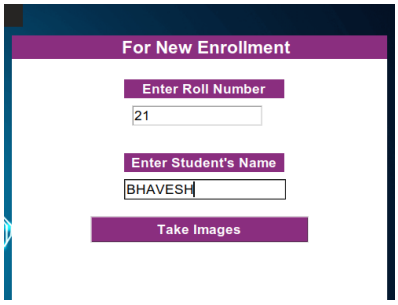

Fig 9.One-time student enrollment

B. Model Training:

Once all the students are enrolled successfully the end-user can train the model. In the model training phase, the defined model is fit using the captured images. "Train the model" on the UI button can be used to begin the training. For running two concurrent functions i.e deterministic progress bar and model_train to train the model, two independent threads are created.

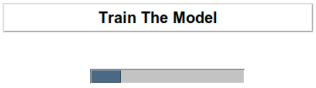

Fig 10.Model training with the deterministic progress bar

C. Taking Attendance:

Once model training and enrollment are done, the end-user can start taking attendance. For the current system, we have created two lectures i.e English and Hindi. The user has to select the lecture and click on the Take attendance button.

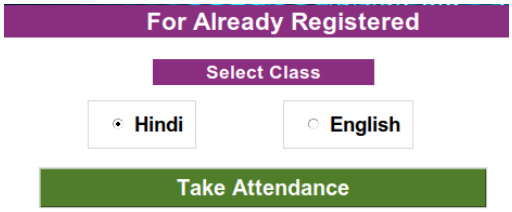

Fig 11. UI for taking attendance

Once the user clicks the button a CV2 window pops out and takes frames and processes it. From each frame, faces are cropped out. The output cropped face is fed as an input to the face-net model to extract the embeddings. These embeddings are predicted using the trained model. For any certain class if the prediction value is more than 0.85 confidence that a particular frame is labeled with the class name. For mapping class with the names of students, a python dictionary is used. Also, for making sure that the student is marked for a single time for a lecture we have created a temporary python dictionary that increments a value if the frame recognizes a particular person for at least 30 frames. Once the student is marked present, "Attendance is marked labeled is displayed on the window in the UI.

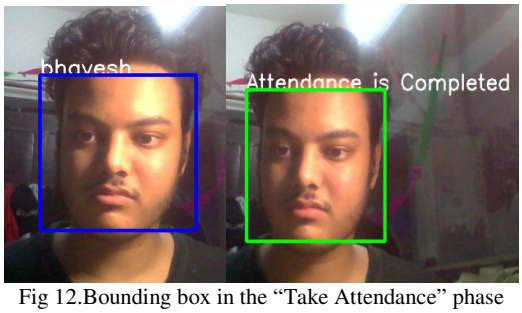

Fig 12.Bounding box in the "Take Attendance" phase

The output attendance is also displayed on the UI that can be displayed using the Show Attendance button

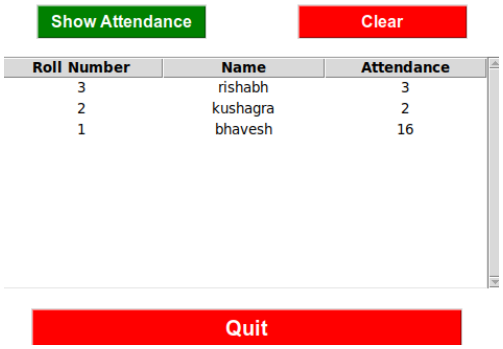

Fig 13.Overall attendance for the selected lecture

IV. DATABASE

A MongoDB database hosted locally is used for storing the student details. The Pymongo library in python is used to interact with the database. During the enrollment phase, the details such as student name, roll number, and attendance as 0 is stored in the MongoDB database for all the collections. We have created two collections “English” and “Hindi” in a database named “Students”. Once, the student details are captured in the database, their attendance is marked. During the Take Attendance phase, the students who are present/recognized by the model are marked present by incrementing the attendance value in the collection for the lecture.



Fig 14.Entries in theMongoDB collection

V. RESULTS

We have tested the system for 7 students/classes. The classification accuracy was improved from 75% to ~89% using regularization, changing the hidden layer size, batch normalization, changing optimizer, and increasing the hidden layers. LeakyReLU with an alpha value of 0.1 and dropout regularization with Adam optimizer provided the best accuracy. The model is highly dependent on the lighting conditions in both the training and inference phases.

Number of hidden layers	Activation function	Classification accuracy (approx.)
6	ReLU	77
3	ReLU	80
4	ReLU	83
3	leakyReLU	87
3	leakyReLU and dropout	89

Table 1. Classification accuracy change by tweaking hyper-parametersand adding regularization

VI. CONCLUSION

Cases such as facial expressions, varying hairstyles, spectacles, etc. have been tested on the system and it has shown full accuracy for most of the cases. Thus, it can be concluded from the above discussion that a reliable, secure, fast, and efficient system has been developed replacing a manual and unreliable system. This system can be implemented for better results regarding the management of attendance. The system will save time, reduce the amount of physical work done by the administration and the electronic apparatus, and reduce the number of human resources required for the purpose. Hence, a system with expected results has been developed but there is still room for improvement.

REFERENCES

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.
[2] Adrian RhesaSeptianSiswanto "Face recognition algorithm for biometrics". ICT For Smart Society (ICISS), 2014 International Conference on 24-25Sept.2014
[3] NusratMubinAra,Nishikanto Sarkar Simul,Md. Saiful Islam, "Convolutional neural network approach for vision-based student recognitionsystem".Computer and Information Technology (ICCIT), 2017 20th International Conference of 2017.
[4] Samuel Lukas ; Aditya Rama Mitra ; RirinIkanaDesanti; Dion Kris-Nadi, "Student attendance system in classroom using face recognition technique"Information and Communication Technology Convergence (ICTC), 2016 International Conference on Oct 2016.
[5] Zhi Li, "A discriminative learning convolutional neural network for facial expression recognition". Computer and Communications (ICCC), 2017 3rdIEEEInternational Conference on December 2017
[6] LoredanaStanciu ;FlorentinaBlidariu "Emotional states recognition by interpreting facial features . EHealth and Bioengineering Conference (EHB),2017June".
[7] Jiachen Chen; W. Kenneth Jenkins "Facial recognition with PCA and machine learning methods. Circuits and Systems (MWSCAS), 2017 IEEE 60th International Midwest Symposium on Date of Conference: 6-9 Aug. 2017"
[8] Neelum Mehta; Sangeeta Jadhav "Facial Emotion recognition using Log Gabor filter and PCA, Published in Computing Communication Control and automation (ICCUBEA), 2016 International Conference on Date of Conference: 12-13 Aug. 2016"
[9] William Gutiérrez Pezoa ; Marcela JamettDomínguez Combined approach using artificial vision and neural networks for facial recognition, Electrical,Electronics Engineering, Information and Communication Technologies (CHILE CON), 2017 CHILEAN Conference, 18-20 Oct. 2017
[10] MashhoodSajid ; Rubab Hussain ; Muhammad Usman. A conceptual model for automated attendance marking system using facial recognition, DigitalInformation Management (ICDIM), 2014 Ninth International Conference on Date of Conference: 29 Sept.-1 Oct. 2014.
[11] NusratMubin Ara ; Nishikanto Sarkar Simul ; Md. Saiful Islam Convolutional neural network approach for vision-based student recognition system,Computer and Information Technology (ICCIT), 2017 20th International Conference of Date of Conference: 22-24 Dec. 2017

AUTHORS PROFILE



Gajanan Walunjkar received his Bachelor of Engineering degree in Computer Engg from Pune University in 2003 and Masters in Computer Engineering degree from Pune University in 2009. He is currently pursuing Ph.D. in Computer Science from Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, India. Presently working as Asst Professor, Dept. of Information Technology in Army Institute of Technology, Dighi, Pune, India. He is having 15 years of experience and has published 20 plus research papers in various international and national referred journals and conferences. His area of interest includes Wireless Communication and ad hoc networks.



Dr. Rahul Desai received his Bachelor of Engineering degree in Electronics Engineering from Pune University in 1998 and Masters in engineering degree from College of Engineering, Pune, Indiain 2007. He also completed Ph.D. in Electronics & telecommunication from PuneUniversity in 2017. Presently he is working as Associate Professor, Dept. of Information Technology in Army Institute of Technology, Pune, India. He has published 25 plus research papers in various international and national referred journalsand conferences. His area of interest includes Wireless Communication, Embedded System andWireless Sensor Network. He is having 20 years of teaching and industry experience.



Bhavesh Singh Bishtis the alumni of Department of Information Technology, Army Institute of Technology,Pune, India. He completed Bachelor of Engineering degree in Information Technology from Army Institute of Technology in 2020 and having research interests in machine learning, artificial intelligence and networking.