Implementation of Automated Attendance System using Facial Identification from Deep Learning Convolutional Neural Networks

Shyam Sunder Bahety
8th Sem ISE, Dept. of ISE JSSATE
Bangalore, India

Kishan Kumar
8th Sem ISE, Dept. of ISE JSSATE
Bangalore, India

Vishwadeep Tejaswi
8th Sem ISE, Dept. of ISE JSSATE
Bangalore, India

Sharad R Balagar
8th Sem ISE, Dept. of ISE JSSATE
Bangalore, India

Anil B C
Assistant Professor, Dept. of ISE JSSATE
Bangalore, India

***Abstract*— Due to the shortcomings of the existing attendance system which has always been time consuming and poses problems to staff of institutions and organizations, we would like to introduce an attendance monitoring system by using deep learning techniques and thereby tap in the potential it promises in the field of face detection and identification. We first capture an image of students in a classroom and utilize the OpenCV module to detect and frame the faces in that image. In the next stage we enhance these frames using an image enhancing model. In the final stage of the project, we build a Convolution Neural Network (CNN) to train these facial images and compare them with the student records that are stored in the database and hence update the attendance status of the students. Our system promises easy to maintain hassle free attendance system and with other integrations it can be used for other necessities of industries and is not limited to educational institutions.**

***Keywords*— OpenCV; CNN.**

I. INTRODUCTION

Digital Image Processing:

Digital image processing refers to the processing of images with the help of computational algorithms. As image processing is a subcategory of digital processing It is considered and known to have multiple advantages over its predecessor analog processing. It typically has a wide range of algorithms that can be applied with reduction in issues like noise buildup or distortion when images are processed. Images typically are known to have multidimensions and hence image processing models or modules too have multiple attributes.

Generally, image processing refers to the process of making an image better with the help of different techniques. A facial recognition system is when a system of modules or software is able to identify and recognize a person.

Facial Recognition modules tend to work in different forms, but the one concept that holds true for any module is the comparison of facial features that are selected uniquely. It can be called as an application that works with artificial intelligence that identifies people based on their unique features like shape, color, or any other unique feature. This is also implementable in various platforms. This technology is widely used in mobile phones in this era by tech giants as it is extremely efficient and hardly any work is to be done by the user. Security of this system must be made better as it does have flaws of its own. With better modules being built everyday it promises to be the most efficient way of security accessing.

Convolution Neural Networks:

Neural networks are a subset of machine learning. They are like the neural network of a human body and hence the name. Since the biological network is intricate and vast, they are also called deep neural networks and are extremely efficient. They are functionally just approximations and hence can be used on almost all machine learning problems dealing with connectivity of inputs and outputs in space. Convolution neural network is known for its effectiveness in recognizing and classifying images. Object detections, face recognition etc., are some of the areas where CNNs have a wide scope and hence rightly used. A CNN takes in an input image, assigns weights to different objects and features in an image and thereby recognizes in this manner. CNN is a classification algorithm and the processing needed before execution is extremely low. CNN has the ability to learn features on its own after training the model for a while.

The main goal of a CNN is to be able to extract advanced features from an image. It has multiple layers, each layer having functions of its own. The user can also add a few hidden layers with functions of their own. The different layers of a CNN are mentioned in the Figure below.

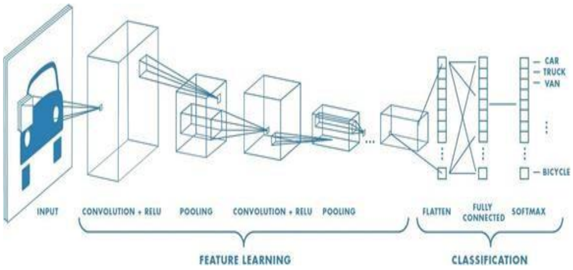


Figure 1: CNN Layers

II. RELATED WORK

Face Detection Method Based on Cascaded Convolutional Networks was a paper written by RONG Q, RUI-SHENG JIA, QI-CHAO MAO, HONG-MEI SUN, LING-QUN ZUO. This
paper uses a cascaded architecture with 3 different stages to improve the performance detection. In this paper they replace the standard CNN with a combination of residual structure in the network. The only drawback this paper has is that the input size isn't fixed.

A Face Detection Framework Based on Deep Cascaded Full Convolutional Neural Networks written by Bikang Peng, Anilkumar Kothalil Gopalakrishnan. This paper presents a face detection technique based on cascaded full CNN to solve the current issues faced by face detection techniques. This technique supports face detection and also detects key facial features by identifying their positioning with the help of third order cascaded CNN.

Face Recognition Based Attendance System, Proc of IJITEE was written by Kalachugari Rohini, Sivaskandha Sanagala, Ravella Venkata Rathnam, Ch.Rajakishore Babu in 2019.
In this paper the solution has been found using both photos and videos. This was used by them to improve security and saving time. This paper also tells if the person is unidentified or not. They have to improve on the future detection and recognition with students and people who develop beard later. The drawback of this paper is that it needed more accuracy.

Multi-scale face detection based on Convolution Neural Network is another paper written by Mingzhu Luo, Yewei Xiao, Yan Zhou in the year 2018. In this paper the model can take in arbitrary sizes as input using spatial pyramid pooling and the detection speed is increased by the division of the mesh extraction from the candidate face. This is a very simple model in which when an image is inputted it directly detects the face. The drawback is that it needs extremely high computation power to train the model.

FDAR-Net: Joint Convolutional Neural Networks for Face Detection and Attribute Recognition written by Hongxin Liu, Xiaorong Shen, Haibing Ren. This paper makes use of CNN named FDAR-Net, which deals with both facial recognition and attribute recognition. For face extraction they use cascade Real Adaboost before applying the FDAR-Net which helps the system run at 71 frames per second and keeps high accuracy rates.

An Approach to Face Detection and Alignment Using Hough Transformation with Convolution Neural Network written by Oshin Misra, Ajit Singh. They make use of Hough Transform and DCNN which is an extremely efficient technique for extracting key features and face detection.It shows high recognition rates and thereby has an extremely high accuracy rate.

III. BLOCK DIAGRAM DESCRIPTION

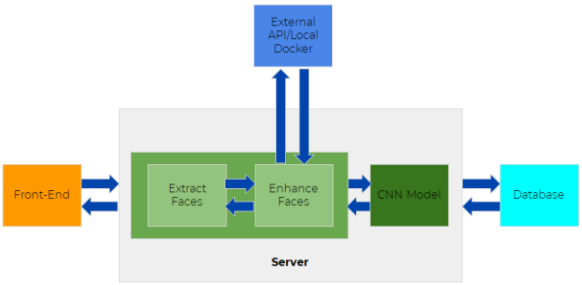


Figure 2: Block Diagram

- A front end web/mobile application captures an image and sends it to the server.
- The server extracts the faces and enhances them using an external API.
- The enhanced image is sent to the CNN model and then the face is identified.
- The identified faces from all the extracted faces are grouped together to find out who all attended the class and the data is then stored in a database.
- The data from the database can also be fetched by the front end application.

IV. METHEDOLOGY

For the working of this project, multiple methods and techniques are used to identify the faces. As mentioned in the block diagram above, an image of class goes through 3 stages.

- Recognition and extraction of faces
- Enhancement of faces to increase colour density in the pixels
- Identification of the face from a pre trained model built using Deep Learning Neural networks

Extraction:

Initially the user sends a POST request and sends an image to the endpoint of the server. This endpoint accepts an image in a jpg, jpeg or a png format. Once the endpoint receives the image, it stores it in the local storage of the server using built in python libraries and functions. Once the image is stored in the server, we use open-CV to extract faces. This functionality is written separately as a face_scrapper function.

How OpenCv Extracts faces -

- The function saves the image as a cv2 object.
- It then grey's out the complete image. This is done so that CV2 can process through the image faster with better results.
- A cascade classifier is defined with built in cascade xml files.
- The cascade classifies calls a multi selection detect method which detects patterns with given arguments
- Few of the arguments given are scaling factor, minimum number of neighbours and the minimum size of the pattern to be detected (30 * 30 usually for a face)
- The multiselect ion detect method returns an object which has all the 4 points of all rectangles of all the faces detected.
- We Loop through the object and colour the rectangles for reference and at the same time, each rectangle is again converted back to a coloured image and saved in another folder. This folder consists of all the faces openCv has detected.

The first step of the project is completed by extracting the faces from the input image for attendance.

Enhancement:

All the extracted images need to be enhanced because, when multiple faces are extracted from a single picture, they either reduce in quality or the resolution decreases. The colour density around a part of an image decreases and the images aren't that sharp. So, to overcome this problem, we use an open apache licensed enhancing model called the Image Resolution Enhancer developed by IBM. This model has the capacity to upscale the image by a factor of 4. This works only when the input image to the model has a resolution between 100x100 - 500x500 pixels.

The main central model used to build this model is the Generative Adversarial Network (also a neural network) which was trained with 600,000 images from an open dataset of Open Images v6. This neural network is based on the SRGAN- tensorflow repository on github and also an online article called as the Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. The backend server which we created takes all those extracted images. The extracted images are stored in a folder. The server loops through those images and takes every image one by one. These extracted images are then sent to the endpoint where the IBM Image enhancer is hosted. This is already hosted on the cloud. It can also be hosted locally using docker.

The enhancer enhances it and sends back the enhanced image back to the request. We then store these images in a buffer and store them locally in another directory.

This is the 2nd step of the process and now we have the enhanced images ready to be passed on to the Face Detection model built using Neural Networks

Face Identification:

For face identification we need to train the model. Training consists of 3 phases

- Image Resizing: We first resize the image to a fixed resolution. As the images received from the enhancer might be of different resolution and aspect ratio, we use the PIL library in python. We loop through the images in a directory and then resize them into an aspect ratio of 1:1. The resolution can be for 64*64, 128*128, 512*512 pixels. This is dependent on the developer. As the resolution increases, so does the density of the neural network. As the neural network keeps getting denser, the more processing power and time it will need.
- Image augmentation and flooding to create the dataset: The next step is to augment the images and flood them. We use the ImageDataGenerator preprocessing module from the keras library. This module augments the input image into any number of images we want. For the sake of this project, we augmented all the images of a single person to a total of 700 images. So, each class(person) in the dataset consists of 700 images.
- Model training and testing: The third and final step is to train the model with Neural Networks. We use the Tensorflow Keras Library in python as they contain almost all the algorithms and required functions to build the model. We again use the ImageDataGenerator to preprocess the image. This is a very basic preprocessing used to store all the images into 2 variables as 2 different batches. We then loop through the batches and separate the training and the test dataset into a python list. These lists are further converted into numpy arrays for the Neural Network model which we build later. After the preprocessing of the images is done and finally converted into a numpy array, they are ready to be trained.

We then create a sequential model from keras which is basically used to create a layered neural network. We add multiple layers to this model for it to work.

- The first layer is the Convolution layer with 32 neurons for the input image to connect to. We use the activation function of relu in this.
- The second layer is the pooling layer with a pool size of 2*2. We use the MaxPooling2d module while others also can be used.

- The third layer is again the Convolution layer
- The fourth layer is the pooling done to the CNN in the previous layer
- Once the layer is pooled, they are flattened out to a single array, which is basically converted into those many neurons of 1 dimension.
- These final layers are used for classification of the image.
- We add a dense layer of 500 neurons with an activation function of relu.
- We finally add the output dense layer with several neurons equal to the number of faces or the number of classes.
- This model is finally optimized with adam optimizer to find the minimum gradient descent.

Once the model is trained, we store the fitted model into a .h5 file which is also called as a pickle file which is used to predict the values later.

V. EXPERIMENT & PROGRESS

We iteratively kept building our model by trying different input types and parameters.

Versions:

1. We trained the model using 64*64 resolution images of students. Here, we could not get much of an accuracy as the resolution was too low for the model to extract any data from the images.
2. Next, we input images of resolution 128*128 and we noticed an increase in our accuracy. The model was able to detect faces but only of the trained images. We'd have to either train the model with more images or higher resolution images.
3. We input images of resolution 512*512 but reduced the number of images. The image size was quite high hence it took quite some time as computing power for the model to run was high. The accuracy was what we wanted but the time complexity was too high. Hence, this is not deployable.
4. Meanwhile, alongside the experimental model, we build a working backend server in Flask that contains API's and our face_scrapping code and enhancement code. To this app, through the API, we can perform a post request and send the input image of the classroom. It will then detect each Individual face and perform enhancement of the image.

VI. RESULT & ANALYSIS

With the field of facial recognition and identification being used in multiple fields ranging from attendance to access control, it has a wide range of applications. Keeping that in mind, high accuracy is a must especially when it comes to the field of access and security. Our model has been trained for 10 classes with each class having a dataset of 500 augmented images. Even with a smaller number of images we were able to achieve high accuracy scores, as shown in the table below. With more images the accuracy will increase and thereby the efficiency of the entire system.

	METHOD	ACCURACY
1	FDAR-NET	0.80
2	IAU-NET	0.78
3	SPP(PYRAMID POOLING)	0.73
4	3-STAGE CNN	0.664
5	OPEN CV, CNN	0.69
6	**OUR METHOD**	**0.82**

Table 1: Accuracy Comparison

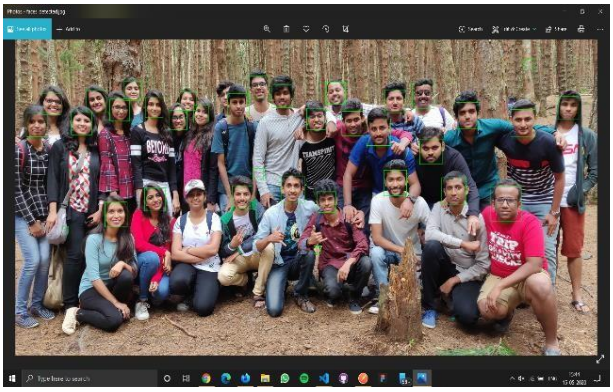
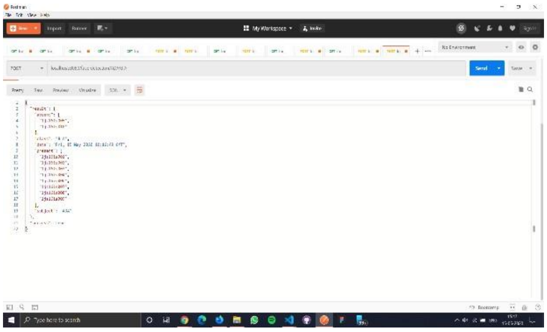


Figure 3: Framed Faces using OpenCV



```
{
    "result": {
        "absent": [
            [illegible]
        ],
        "class": [illegible],
        "date": "Fri, 15 May 2020 [illegible] GMT",
        "present": [
            [illegible]
        ],
        "subject": [illegible]
    },
    [illegible]
}
```

Figure 4: Attendance Marked

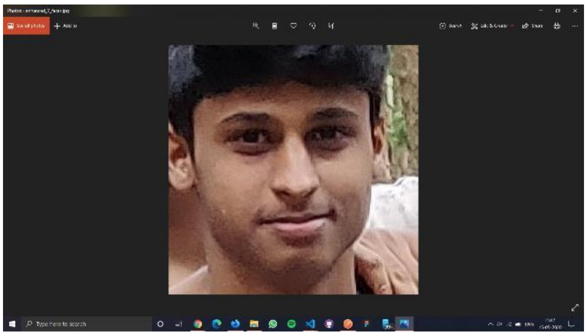


Figure 5: Enhanced Image

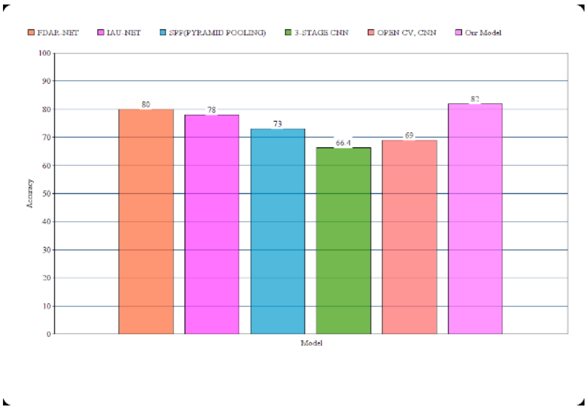


Figure 6: Bar Graph of Accuracy Score

VII. CONCLUSION

This paper proposes an alternative solution to the traditional way of taking attendance. The method used here is CNN, where we must train our model and make it capable of identifying the faces of students. To the model we input a cropped and enhanced family images of students from the class photo taken by the teacher. The attendance is marked automatically after identification of students. This saves a lot of time for the teacher and we are achieving this through a simple image taken from the lecturer's cell phone.

VIII. REFERENCES

[1] Hongxin Liu, Xiaorong Shen, Haibing Ren, "FDAR-Net: Joint Convolutional Neural Networks for Face Detection and Attribute Recognition", 2016 9th International Symposium on Computational Intelligence and Design

[2] Bikang Peng, Anilkumar Kothalil Gopalakrishnan, "A Face Detection Framework Based on Deep Cascaded Full Convolutional Neural Networks", 2019 IEEE 4th International Conference on Computer and Communication Systems

[3] Oshin Misra, Ajit Singh, "An Approach to Face Detection and Alignment Using Hough Transformation with Convolution Neural Network", Proc of IEEE 2016

[4] Mingzhu Luo, Yewei Xiao, Yan Zhou, "Multi-scale face detection based on Convolution Neural Network", Proc of IEEE 2018

[5] RONG Q, RUI-SHENG JIA, QI-CHAO MAO, HONG-MEI SUN, LING-QUN ZUO, "Face Detection Method Based on Cascaded Convolutional Networks", Proc of IEEE 2019

[6] Kalachugari Rohini, Sivaskandha Sanagala, Ravella Venkata Rathnam, Ch.Rajakishore Babu, "Face Recognition Based Attendance System,
Proc of IJITEE", Volume-8 Issue-4S2 March, 2019

[7] Pooja, J. Gaurav, C. R. Yamuna Devi, H. L. Aravindha, M.Sowmya, "Smart Attendance System Using Deep Learning Convolutional Neural Network", Springer Nature Switzerland AG 2019

[8] Mr. Omkar Sawant, Mr.Yash Jain, Mr. Anand Kulkarni, Mr. Shubham Salunkhe, Prof. Mrs Shantiguru, "E - Attendance System Using Opencv and CNN", Proc of IJARCCE, Vol. 8, Issue 4, April 2019

[9] Rajeev Ranjan, Vishal M. Patel, Senior Member, Rama Chellappa, "HyperFace: A Deep Multitask Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition",
Proc of IEEE, VOL. XX, NO. XX, 2016

[10] Poornima S, Sripriya N, Vijayalakshmi B, Vishnupriya P, "Attendance Monitoring System using Facial Recognition with Audio Output and Gender Classification", IEEE International Conference on Computer,
Communication, and Signal Processing, 2017

[11] Y.B. Ravi Kumar, C.K. Narayanappa, Dayananda P,"Weighted full binary treesliced binary pattern: An RGB-D image descriptor",Heliyon, Volume 6, Issue 5,2020,e03751

[12] E.Varadharajan,R.Dharani, S.Jeevitha, B.Kavinmathi, S.Hemalatha, "AUTOMATIC ATTENDANCE MANAGEMENT SYSTEM USING FACE DETECTION", 2016 Online International Conference on Green Engineering and Technologies (IC-GET)

[13] S.U. Jung, Y.S. Chung, J.H. Yoo, and K.Y. Moon, "Real-time face verification for mobile platforms," Advances in visual computing, 2008, pp. 823 – 832.

[14] S. Zhu, C. Li, C. C. Loy and X. Tang, Transferring landmark annotations for cross dataset face alignment. ArXiv preprint arXiv: 1409.0602, 2014.

[15] Detecting and Aligning Faces by Image Retrieval ,Xiaohui Shen, Zhe Lin, Jonathan Brandt and Ying Wu, CPVR-2013, pp. 4321 - 4328.

[16] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," Science, vol. 291, no. 5, pp. 599 – 600, Jan. 2000.