International Journal of Scientific Research & Engineering Trends
Volume 7, Issue 4, July-Aug-2021, ISSN (Online): 2395-566X

Smart Automation Attendance System using Neural Networks

Satwik Ram Kodandaram
AI/NLP Scientist
Information Services Group,
Karnataka, India.
satwikram29@gmail.com

Kushal Honnappa
AI/NLP Scientist
Avkara/ Vmitis
Rajasthan, India..
kushal.h1999@gmail.com

Rishab Darshan S
Department of Computer Science,
Visvesvaraya Technological University,
Karnataka, India.
rishabds99@gmail.com

***Abstract*- A Traditional Attendance system will be a huge burden on teachers. After marking down, it manually, they have to upload it to some database to maintain the student's records. When there is a batch of students more than 500, doing it manually is imaginary. To resolve this problem, a smart and auto attendance management system is utilized. But authentication is a very important issue in this system. The smart attendance system is generally done with the help of biometrics. Face recognition is one of the best biometric methods to improve this attendance system. Being a prime feature of biometric verification, facial recognition is used in several applications like video monitoring. By utilizing this system, the problem of proxies and students being marked present even though they are not physically present in the class can be easily solved. The main implementation step used in this system is face detection and recognizing the detected face. This paper proposes a model for implementing an automated attendance management system using ANN and CNN. After this, the student's faces are mapped to their USN (University serial number) or ID. When the student's face is recognized and if they are present in the current location of the teacher, automatically the USN mapped to that student is marked as present in the database. In this model, the teacher needs to hosts the system through a web application or android application. While hosting the teacher's current location is taken. This model will be a successful technique to manage the attendance and records of students.**

Keywords- Students, teachers, attendance, face detection, neural networks.

I. INTRODUCTION

All colleges or schools have to track attendance in some way, so they can recognize which students are arriving late and which are always sharp on time and to maintain the record. The traditional method involved old-fashioned punch clocks, signatures on paper sheets, or some other kind of manual system that requires human oversight. If the attendance is taken manually, few problems may occur some of them are

- It is very difficult to handle huge students. For example a student's batch of more than 500 students.
- It is very time-consuming to call all the student's names and marktheir attendance.
- There will be chances of giving proxy (Fake) attendance, where it is difficult to handle the proxy.
- Now, what if we want to submit or send the attendance record to the university where exam ticket is issued based on a certain percentage of attendance, again from manual to data entry job need to be done which is again time-consuming.
- Inaccurate and subject to manipulation ('time theft')
- High possibility of human error.

A solution to the above system is the smart automation attendance system via face detection and using the current location. If the student location matches the current location of the teacher then the only student is marked as present in the database. Face detection is one of the important biometrics used in a variety of applications that identifies human faces in digital images.

Fig 1. Face Recognition.

Face detection can be done using neural networks which is part of artificial intelligence, where a machine can learn by itself without being explicitly programming. The important thing in this system is to train the machine with the student's images so that machine will learn by itself to detect the faces of students.

II. LITERATURE REVIEW

There are many systems developed in colleges and industries to keep track of attendance. But the performance and stability problems.

1. An attendance system using face detection in a classroom environment:
This System works perfectly fine when there is a fewer number of students. If there is a student batch of 200-500, multiple cameras must be used. Students have to focus on cameras where there might be chances of mixing up of faces and some students might not be captured.

2. Biometric-based System:
The Biometric- based systems take a unique part of the human body and are used for an attendance management system. Example Iris, retina scan. The iris recognition system is a useful system but the main drawback is it can cause eye injuries.

3. Bluetooth Based System:
This system is highly efficient and 100% proxy can be avoided. However, the system requires 8 connections active at a time. Bluetooth doesn't allow more than 8 connections this is due to the master and slave concept.

III. NEURAL NETWORKS

1. The Neuron:
In biology neurons consists of branches called Dendrites and a long tail called axon as shown in Figure 2.

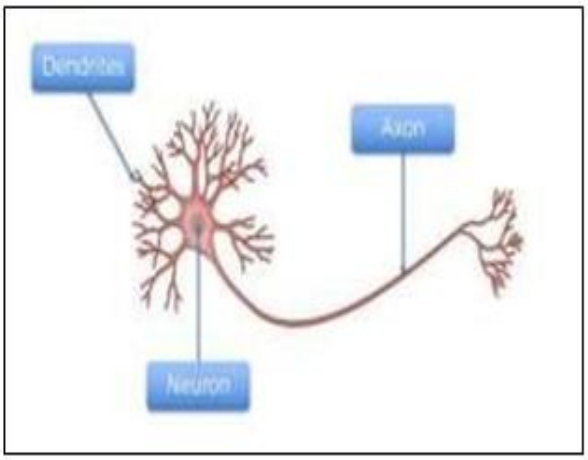

Fig 2. Neural Network system.

Dendrites are receivers and axons are transmitters. The key point to understand neurons is they are themselves useless. It is like a hand; five hands can do something. Dendrites are connected in neurons. Electrical signals are passed through neurons.

2. How to Represents neurons in Artificial Intelligence:
Technically neurons can be represented as below: Neuron is also called a node that receives an input signal and produces an output signal. These input signals come from other neurons called input layers. We also get input from hidden layer neurons. Here input signal is the independent variable in a model. Ex- age of a person, etc. we have to standardize and normalize the input variables. Output values can be continuous. Ex – price value.

Binary (Y or N Example whether a person will quite not.). Categorical example multiple values. Each signal consists of considerable weights. Weights are very important they are adjusted to learn by themselves. As shown in Figure 2, Neurons will decide what input signal is important based on these weights.

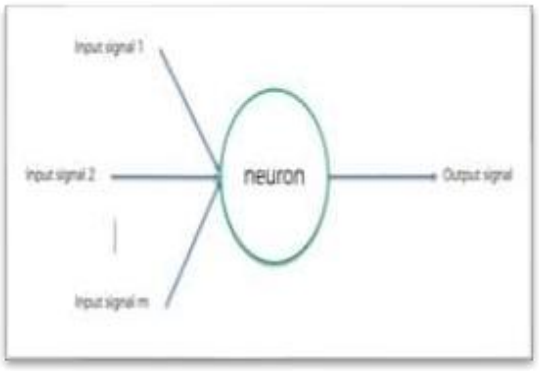

Fig 3. Neuron.

3. How does Neural Network works in Artificial Intelligence:
As shown in figure 3, in the first step all the weights are added and multiplied with independent variables. And a bias is also added in this step.

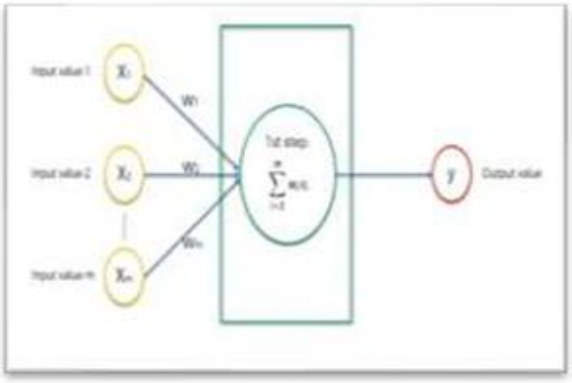

Fig 4. Neural networks in AI.

The basic formula for this is as follows

Y = w1*x1 + w2*x2 + w3*x3…+ wm*xm + Bias

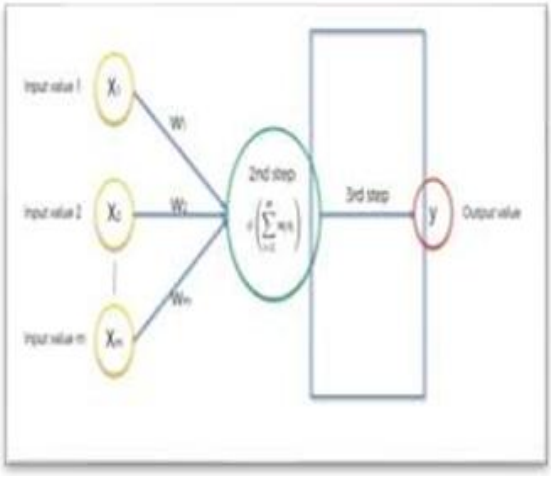

Fig 5. A neural network Activation function.

In the second step activation function is applied and in the third step, the signals are sent to the output layers. This complete step is called forward propagation.

The output of this is called as y predicted. This y predicted is compared with the actual y value and the weights are updated using the chain rule. This step is called Back Propagation

IV. FACE DETECTION USING CNN

1. Cerebral Cortex:
Human brain is divided into 4 parts. One of the parts is the cerebral Cortex. The Cerebral Cortex is the thin layer of the brain that covers the outer portion of the brain. Inside the cerebral cortex, we have the visual cortex.

2. Visual Cortex:
Visual Cortex is responsible for seeing the images. Suppose we are seeing a cat, that information is passed into our sensory organ eyes. Once the information is passed it will be passed into several neurons and then reaches the cerebral cortex region.

In the cerebral cortex we have a visual cortex, In the visual cortex, we have multiple layers say v1, v2, v3, and so on. These layers play a very important role. Suppose the v1 layer is responsible for finding the edge of the cat-like the eye edge of that, body edge of the cat and it is calculated by the v1 layer.

Then the information goes to the next layer that is v2, some more information is gathered in v2 like whether the cat is moving or sleeping, or any other object is present besides say dog and it is responsible to distinguish between cat and dog. Likewise, the information is passed to different layers and different information is gathered in different layers.

In each layer different operations are performed this is called filters in CNN. Finally, the image will be created in the visual cortex.

3. Convolution Neural Networks:
In the previous section we saw how the human brain recognizes the image. The same steps are followed in CNN.

4. The CNN has two important parts:

- **Feature Extraction:** In this part, the neural network will perform series of convolution and pooling operations. If we had a picture of a cat, it will recognize that the cat has two ears, 4four legs and it is sleeping.
- **Classification:** Here fully connected layer will treat as a classifier on the images. Suppose if two images are given one with a cat and the other with a cat, the model will classify the two different images with cat and dog.

The important steps involved in Convolution Neural Networks are:

5. Convolution:

0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

6. Original Image:

1	0	-1
2	0	-2
1	0	-1

7. Image Filter:
In computer, images are represented in 0's and 1's. Here image consists of a 6x6 matrix and a filter consists of a 3x3 matrix. As discussed earlier v1 was responsible for finding the edge of the image. Likewise here filter represents the vertical edge filter. That is if we apply this filter to any image we can determine the vertical edge of the image. Multiply the first 3x3 matrix with the filter and add everything. Now jump by 1 and follow the same procedure. Once we come to end we should step one down

and continue the same procedure. The result of the multiplication of the matrix is a 4x4 matrix. This is called the Convolution operation.

0	-4	-4	0
0	-4	-4	0
0	-4	-4	0
0	-4	-4	0

8. Result:

These filters play a very important role. Here we just discussed the vertical edge filter. Likewise, we can have so many filters. Filters like finding the edge of the face, horizontal edge filter, and so on.

By looking at the result we can conclude a formula that image size n = 6 and filter size f =6.

$$\textbf{Result} = n - f + 1$$

$$\textbf{Result} = 6 - 3 + 1 = 4$$

If we look at the result, we are getting a 4x4 matrix but an image size 6x6 matrix. Here we are losing some of the information of the image. To avoid that we will apply the concept called Padding.

9. Padding:

$$n - f + 1 = 6$$
$$n = 6 - 1 + f$$
$$n = 5 + 3 = 8$$

Here we are getting n = 8 that is if we apply a padding p =1, that is one more row and column on an image we will get an image size of 8x8 matrix. On multiplying the image with a filter. We get the result as follows.

$$\textbf{Result} = n + 2p - f + 1$$

$$\textbf{Result} = 6 + 2 - 3 + 1 = 6$$

Here we get the resultant matrix of 6x6 and we are not losing any data. If we apply the padding formula we can retain the original size of the image.

We can apply n number of convolution and padding to train the images.

10. Max Pooling:

We saw how convolution and padding operations work. In max-pooling, one of the filters called the max-pooling filter is placed on the filters. Here we take the maximum value of the pixels and place it over a separate matrix.

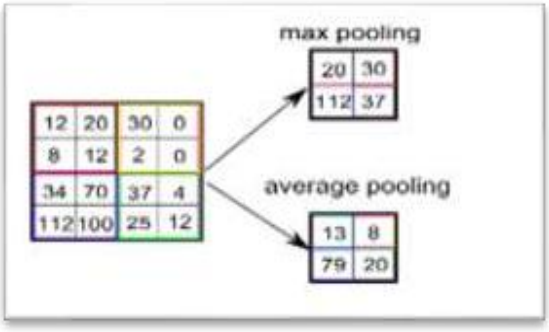

Fig 6. Max pooling.

Before we can barely recognize the face of the image. When max pooling is applied, we are taking only a high pixel value, so we can recognize the face of the image.

Flatten

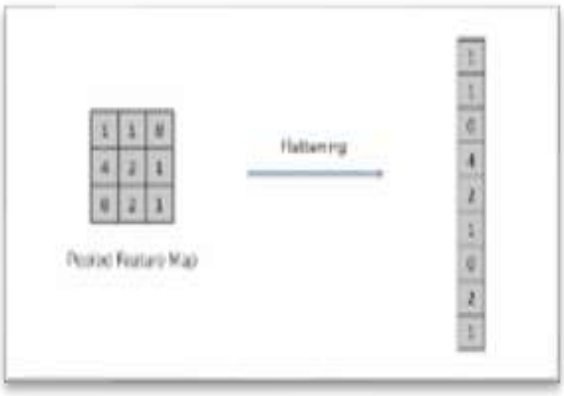

Fig 7. Pooled Feature Map.

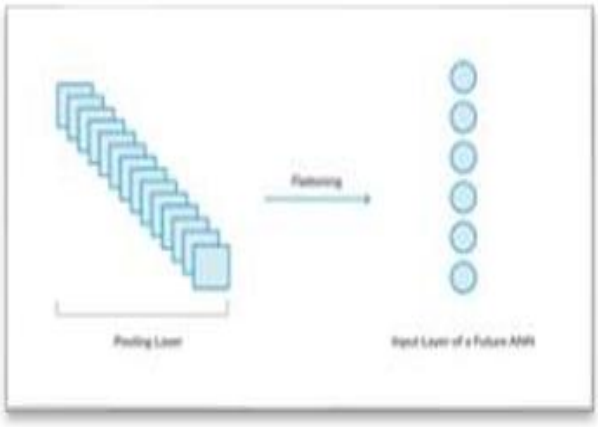

Fig 8. Pooling Layer.

11. Full Connection:

A full connection operation is applied at last. Where after flattening the outputs of flattened images are the input to the neural networks. As explained the working of neural network the same procedure is followed here

12. Back Propagation:

The output of the model is yhat, that is the predicted value. The actual output is y. Suppose yhat is 0 that is the

predicted value and the actual y value is 1. Here we try to see the difference between y and yhat, for that we use the loss function.

The loss function is calculated as:

$$\textbf{loss} = (y - yhat)^2$$

When we apply the loss function:

$$\textbf{Loss} = (1 - 0)^2 = 1$$

The predicted value is now matching with the actual value. Here the predicted value was the opposite. In the general case, we have to update the input weights so that the predicted value should match the output value. Here we define the loss function and our goal is to minimize the loss function so that yhat should match y. So in every epoch, we aim to update the value of the weight so that the loss value should decrease in every epoch. To achieve this we use optimizers, for Example, Gradient Descent.

13. MTCNN:

MTCNN- Multitasking Cascaded Convolutional Networks is a model for face detection which is available as a python library that is easy to use and install. It adopts a 3-stage neural network detector to detect the face of an object as well as bounding boxes with excellent precision. Not only it can capture the face but also it is capable of identifying facial landmarks like eyes, nose, and corners of a mouth.

Also, it can achieve a significantly faster rate of image processing of around 60-100 fps. Overall, this model is considered to be better suitable for real-time face detection of objects in images or videos [4].

Fig 9. Example MTCNN.

V. DATA AUGMENTATION

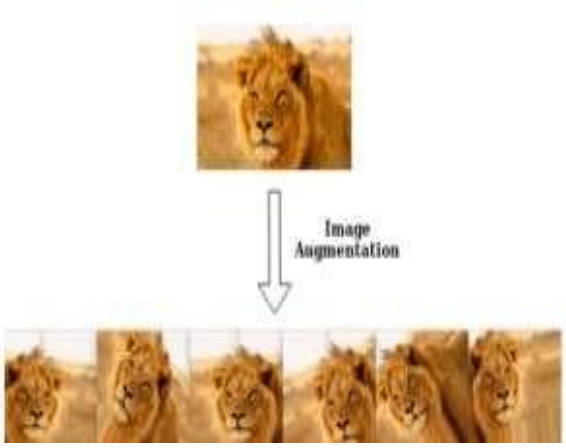

Fig 10. Data Augmentation.

Data Augmentation is one of the important steps in CNN. Data Augmentation helps us to increase the number of data available for training models, without actually collecting new data. Suppose if a have a cat image, is going input to the CNN, and the model will predict the output. With the help of data augmentation, we can transform the input cat image into different kinds of images.

Here still the output remains the same, we are just transforming the images to a different kind. Example flipping of image, Horizontal shifting, vertical shifting, and inverting of image. We can also zoom the image and send it to CNN. Whenever we have less data we can apply this technique to increase the images for training the model and to achieve the best accuracy.

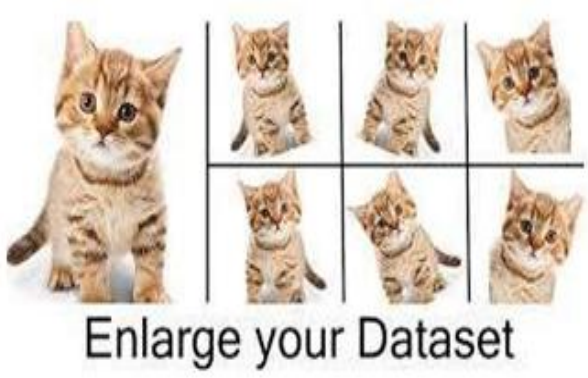

Fig 11. Data Augmentation.

If a student gives his face with a side angle our model should be able to predict the student. These transform images added are called invariance. Data Augmentation helps us to increase the images for training the model by almost 10x the original image. Suppose if we have 10

images, by applying this technique we can get 100 images thus improves the accuracy.

Fig 12. Data Augmentation generated.

The figure fig 12 shows the augmented images using our algorithm.

VI. PROPOSED MODEL

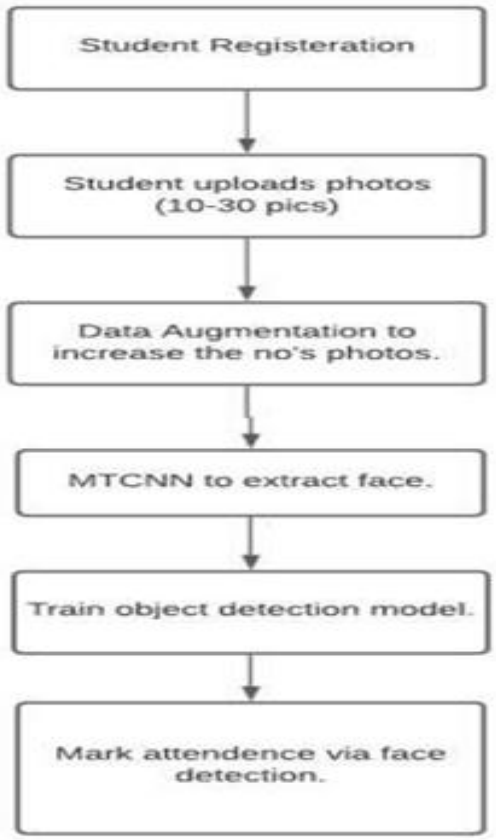

Fig 13. Proposed model flowchart.

As we all know how the technology is improved, everyone will have at least a basic smartphone, or students pursuing a degree will have a basic laptop. We will provide a website where students need to sign up to the website by providing all their information like USN, ID, and NAME, etc.

Here student image is compulsory where it is mapped into their ID or USN. Once the student is done with the sign-up process, they can log in to give their attendance by just giving their face identity (just like face unlock in the smartphone)

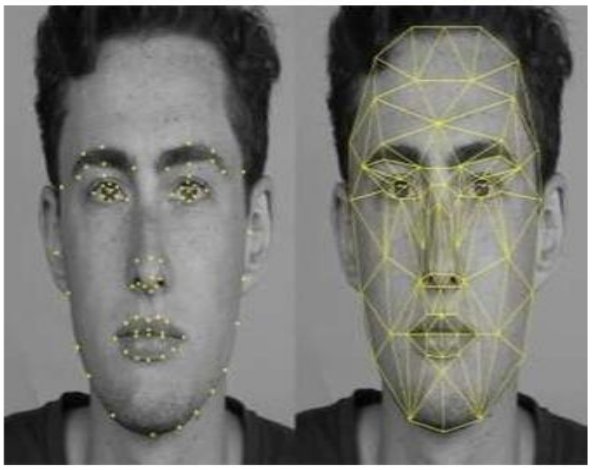

Fig 14. Face recognition.

1. How does it work?
On the website, there will be two login options

- Student login
- professor login

The subject professor needs to log in and hosts the attendance system where students can log in and give their attendance. Once the subject professor hosts the system, the professor's current location will be taken.

When students give their attendance by giving their face identity, if he/she is present in the current location then automatically image mapped to USN or ID will be marked as present in the database. If the student is not present in the current location of the professor then the student will be marked as absent in the database. Image recognition is done using Neural networks.

2. Methodology:
Image recognition is done using Tensor flow (Keras)
3. Techniques used:

- ANN(Artificial Neural Network)
- CNN(Convolution Neural Network)

3. The steps involved are:

- Initializing the model using Sequential class.

- Adding the Input layer that is images and applying the Convolution 2D method.
- Adding the Hidden layer and applying the activation function.
- Max pooling method is applied then.
- Flattening the image using the Flatten method.
- Adding the output layer and applying a suitable activation function. This is called the full connection method.
- Compiling the model.
- Predicting the model.
- Evaluating accuracy.
- Deploying the model using frameworks like Django web Frameworks or Flask

4. Web Developing steps:

- The frontend is done using Html, CSS, and JavaScript.
- Backend is done using Django or Flask.
- The database used Postgresql, SQL, etc.

VII. OBSERVATION

Fig 15. Home page.

Every time student data is trained again and again. If the student's image is recognized and if his/ her location is the same as the professor's current location then that student is marked as present in the database.

Fig 16. Face detection using MTCNN.

```
def draw_faces(filename, result_list):
        # load the image
        data = plt.imread(filename)
        # plot each face as a subplot
        for i in range(len(result_list)):
                # get coordinates
                x1, y1, width, height = result_list[i]['box']
                x2, y2 = x1 + width, y1 + height
                # define subplot
                plt.subplot(1, len(result_list), i+1)
                plt.axis('off')
                # plot face
                plt.imshow(data[y1:y2, x1:x2])
                filename = 'face'+str(i)+'.jpg'
                cv2.imwrite(filename, data[y1:y2, x1:x2])
        # show the plot
        plt.show()
```

```
draw_faces(filename, faces)
```

Fig 17. Face extraction using MTCNN.

Fig 18. Attendance system via face detection.

VIII. CONCLUSION

By using this system, we can overcome the current manual attendance system. 100% of proxy attendance can be reduced. Easy to maintain and evaluate student attendance for every subject.

REFERENCES

[1] M. Sajid, R. Hussain, and M. Usman, "A conceptual model for automated attendance marking system using facial recognition," Ninth International Conference on Digital Information Management (ICDIM 2014), Phitsanulok, 2014, pp. 7 -10.

[2] K. Puthea, R. Hartanto and R. Hidayat, "A review paper on attendance marking system based on face

recognition," 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, 2017, pp. 304-309.

[3] S. Bhattacharya, G. S. Nainala, P. Das and A. Routray, "Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environment," 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), Mumbai, 2018, pp. 358-360.

[4] Xiang, J., & Zhu, G. (2017, July). Joint face detection and facial expression recognition with MTCNN. In 2017 4th international conference on information science and control engineering (ICISCE) (pp. 424-427). IEEE.