Automatic Attendance Management System Using Face Recognition

Department of Computer Science - Informatics and Engineering, Bologna, Italy

Academic Year 2020-2021

Authors

Yellam Naidu Kottavalasa

Ganesh Pavan Kartikeya Bharadwaj Kolluri

Usha Padma Rachakonda

Contents:

1.Introduction:

1.1Problem Statement:

In this digital age, it is a difficult job to maintain the attendance record with day-to-day events and usually, it takes time to use the traditional method and it is a tedious task, sometimes it has a risk of proxy attendance.

Aim and Objectives:

The main objective of this project is to design a system which can mark the attendance by recognizing the students face. Apart from the traditional method. There are so many ways to mark attendance, but here we are going with Face Recognition. When a student enters the class, the Camera will detect the face in the frame and after that, it will recognize the student and it will mark attendance for him/her in the excel sheet.

It is not very efficient to maintain and record attendance in a class by means of a manual process. Since bunking the courses or offering proxies for the absentees, the students of the current generation become enjoyable and fantastic. It becomes a daunting task to manually enter attendance in logbooks and it can be easily manipulated.

For the purpose of attendance, the use of face recognition Marking is the smart way to handle attendance. Face recognition is more precise and quicker. The procedure, among other methods, decreases the likelihood of Attendance by proxy. Face recognition provides passive recognition of images. It is an efficient way to mark attendance among all other techniques.

2.Methodology

2.1 Approach:

Firstly, we must start with data collection. This data collection process is an automated task that is achieved by a face detection algorithm (Viola-Jones Algorithm). It will detect the face in the whole frame and then we will crop the image according to ROI (Region of Interest). Next, we are going with the Transfer Learning method which means we are going to use a pre-trained Convolution Neural Network (CNN). Which is Vgg-16, and we have to modify the output layer according to our task. At last, our pre-trained CNN model will recognize the student. Then we must mark attendance for that student with a timestamp.

2.2 Dataset:

We are going to collect our own images and to increase the number of images for enhancing the accuracy of the model and we are going to use the Data Augmentation technique.

Model Implementation:

Here we are going with a pre-trained model which is Vgg-16 with trained ImageNet weights. Firstly, we must collect the data.

Data Collection:

This code file collects images for the training set and test set for each person and stores them in the file path mentioned there.

Firstly, we have to import required libraries such as NumPy and cv2 (OpenCV).

To detecting faces, we are going with a **Viola-Jones** algorithm. We are loading the **Haar Cascade** file (Frontal_Face_Haar_Cascade.xml file) which will identify the face in the whole image.

To collecting face images, we have to capture images from a webcam using cv2.VideoCapture() method and then we have to read each frame and detect the faces using a viola-jones algorithm.

We are going to collect the Training set (240 images per student) and Test set (60 per student) separately and storing them in a specified path.

Face Recognition:

Firstly, we have to import required libraries and define the training set, the test set images path, and input image size which should be [244,244,3] because of **Vgg-16** trained with [244,244,3] dimensions of images only.

Then instantiate the **Vgg-16** model from TensorFlow's applications module with input shape, "imagine" weights and include top should be "False" because we will modify the output layer of the **Vgg-16** model.

We have to make it False for each and every layer to train. Because there are already pre-trained. If we want to train the model again it is computationally very expensive and time-consuming. So, we will go with the pre-trained weights.

Here we are going to provide how many classes we must predict using our same old **Vgg-16** model and lastly, we will give folders length (number of students) as output dense layer's Neuron and SoftMax as activation function. That will be going to provide us predicted probability density function of our vgg's output layer.

Model summary:

```
Model: "functional_1"
_________________________________________________________________
Layer (type)              Output Shape            Param #
=================================================================
input_1 (InputLayer)        [(None, 244, 244, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)       (None, 244, 244, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)       (None, 244, 244, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 122, 122, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)       (None, 122, 122, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)       (None, 122, 122, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 61, 61, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)       (None, 61, 61, 256)      295168
_________________________________________________________________
block3_conv2 (Conv2D)       (None, 61, 61, 256)      590080
_________________________________________________________________
block3_conv3 (Conv2D)       (None, 61, 61, 256)      590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 30, 30, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)       (None, 30, 30, 512)      1180160
_________________________________________________________________
block4_conv2 (Conv2D)       (None, 30, 30, 512)      2359808
_________________________________________________________________
block4_conv3 (Conv2D)       (None, 30, 30, 512)      2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 15, 15, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)       (None, 15, 15, 512)      2359808
_________________________________________________________________
```

```
block5_conv2 (Conv2D)       (None, 15, 15, 512)      2359808

block5_conv3 (Conv2D)       (None, 15, 15, 512)      2359808

block5_pool (MaxPooling2D)   (None, 7, 7, 512)        0

flatten (Flatten)          (None, 25088)          0

dense (Dense)             (None, 3)             75267
=================================================================
Total params: 14,789,955
Trainable params: 75,267
Non-trainable params: 14,714,688
```

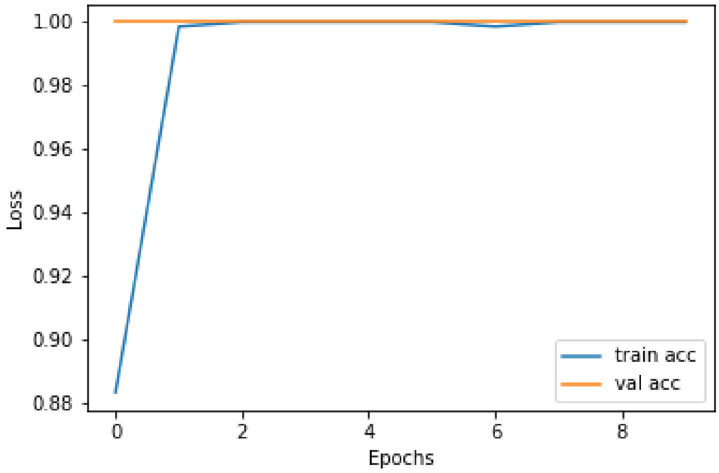
- Here we are going to compile the model with “categorical_crossentropy” loss function, “Adam” op timizer and accuracy as metric for the model.
- Here we want to enhance the accuracy of the model. So, we must increase the images by using Dat a Augmentation technique.
- Lastly, we must fit the model with parameters: training set, validation set, 20 epochs.

Model Training:

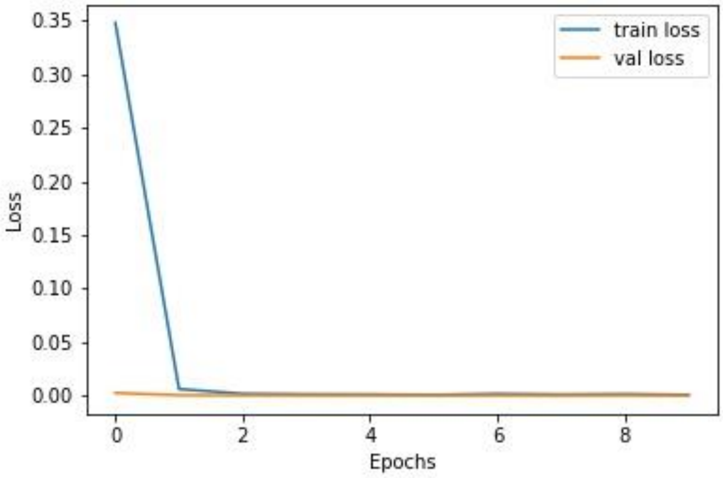
```
Epoch 1/10
23/23 [============================] - 22s 944ms/step - loss: 0.3482 -  accuracy: 0.8833
- val_loss: 0.0025 - val_accuracy: 1.0000
Epoch 2/10
23/23 [===========================] - 17s 742ms/step - loss: 0.0062 -    accuracy: 0.9986
- val_loss: 4.3585e-04 - val_accuracy: 1.0000
Epoch 3/10
23/23 [============================] - 16s 696ms/step - loss: 0.0018 - accuracy: 1.0000
- val_loss: 2.9394e-04 - val_accuracy: 1.0000
Epoch 4/10
23/23 [===========================] - 17s 751ms/step - loss: 0.0013 -    accuracy: 1.0000
- val_loss: 2.8761e-04 - val_accuracy: 1.0000
Epoch 5/10
23/23 [============================] - 18s 780ms/step - loss: 0.0012 - accuracy: 1.0000
- val_loss: 2.4117e-04 - val_accuracy: 1.0000
Epoch 6/10
23/23 [=========================] - 17s 730ms/step - loss: 7.5342e-04 - accuracy: 1.0000
- val_loss: 2.3172e-04 - val_accuracy: 1.0000
Epoch 7/10
23/23 [============================] - 17s 753ms/step - loss: 0.0017 - accuracy: 0.9986
- val_loss: 2.0407e-04 - val_accuracy: 1.0000
Epoch 8/10
23/23 [============================] - 19s 814ms/step - loss: 0.0012 - accuracy: 1.0000
- val_loss: 2.5320e-04 - val_accuracy: 1.0000
Epoch 9/10
23/23 [============================] - 17s 744ms/step - loss: 0.0015 - accuracy: 1.0000
- val_loss: 2.7298e-04 - val_accuracy: 1.0000
Epoch 10/10
```

```
23/23 [==========================] - 18s 790ms/step - loss: 7.5506e-04 - accuracy: 1.0000
- val_loss: 1.6765e-04 - val_accuracy: 1.0000
```

Visualizing Training Accuracy and Validation Accuracy:



Visualizing Training Accuracy and Validation Loss:



- Finally, we are going to save the model because it's going to be a tedious task if we want to train the model every when and then. After saving the model we can use the model whenever we want it.

Recognizing and Marking Attendance:

- Firstly, we are going to import the required libraries and load the Haar cascade file for detecting faces from the whole image.We have to Load the trained model saved previously to recognize the students.

- Here marking_attendance() method going to mark the attendance of a particular student when our mo del recognizes the student on the frames. Then this going to mark attendance to the student with their ID, Name, and Timestamp.

- Here comes the Recognizing part, firstly we will capture the frames from the webcam and use the trai ned model for classifying each student. classification is done by softmax probability values we will s ome conditions for matching the particular student. we are using some threshold value which is 0.5. i f the particular predicted probability is greater than 0.5. we are going to assign the ID and Name and c alling the mark_attendance method to mark the attendance.

Conclusion:

The main objective of this project is to design a system which can mark the attendance by recognizing the students face. Apart from the traditional method. There are so many ways to mark attendance, but here we are going with Face Recognition. When a student enters the class, the Camera will detect the face in the frame and after that, it will recognize the student and it will mark attendance for him/her in the excel sheet.

References:

[1] Sanjay Badhe, Kunal Chaudhari, Sneha Kale, Tejaswi Mane, “Smart Attendance Management System”, International Journal of Computer Applications.

[2] Zhang, Xiao zheng, and Yong sheng Gao, "Face recognition across pose: A review." Pattern Recognition42.11 (2009): 2876-2896.

[3] Lacey et al, “A Longitudinal Study of AutomaticFace Recognition‖”, IEEE ICB, 2015, pp: 1- 8.

[4] Ms. Snehal Houshiram Gorde1, et al, “A Review on Face Recognition Algorithms”, Volume III, Issue I Issn No.:2350-1146, I.F-2.71.

[5] C. Szegedy et al., "Going deeper with convolutions", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 1-9, Jun. 2015.

[6] Selvi, K. Senthamil, P. Chitrakala, and A. Antony Jenitha. "Face recognition based attendance marking system." Corresponding Author: S. Rajkumar*, Email: rajkumarsrajkumar@ gamil. com (2014).

[7] Deshmukh, Badal J., and Sudhir M. Kharad. "Efficient Attendance Management: A Face Recognition Approach." (2014).