IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 2, February 2021

DOI 10.17148/IJARCCE.2021.10209

Prevention of ATM-Robbery using Machine Learning

Ankit Shukla[1], Rohit[2], Aditya Kumar[3]

40713303117, IT, HMRITM, Delhi, India[1]

35213303117, IT, HMRITM, Delhi, India [2]

42713303117, IT, HMRITM, Delhi, India[3]

Abstract: The idea that the design and implementation of a real-time ATM robbery project came from a standpoint of real-time ATM robberies. This project provides a warning at a time when the thief is about to break the ATM.so machine, overcoming obstacles in the existing systems in our society. Whenever a thief brings robbery tools to an ATM or when a thief tries to use a tool to break in, the CCTV camera at the ATM detects whether the person is coming with tools using in-depth learning and machine learning methods. Here OpenCV is used as a speaker and the python language is used for in-depth learning strategies with Haar Cascade, the Yolo V4 and for object detection. A warning message is also sent to the bank and an alarm is sounded to alert local authorities.

Keywords: ATM, Deep learning, Open CV, Haar Cascade, Python, Yolo V4, CCTV, Alert message

I. Introduction

In today's technologically advanced world, autonomous systems are rapidly evolving and gaining popularity. As the use of computers and mobile devices has improved in the financial sector. Also, crime and robberies related to the financial sector have increased and ATM-related robberies have also increased dramatically in recent years. Between robberies in the financial sector has increased by an average of 90 percent. For 2017-2018 the loss of money for robbery, theft, decommissioning and ATM theft is Rs.18.63cr. The ATM is only available after a robbery or an ATM attack. Therefore, this project works as a way to prevent ATM robberies in real time when a person is guaranteed to have any ATM robbery tools. Therefore, through in-depth learning and machine learning techniques to find someone with an important theft tool. Number of ATM fraud incidents reported across India in 2018, by leading state

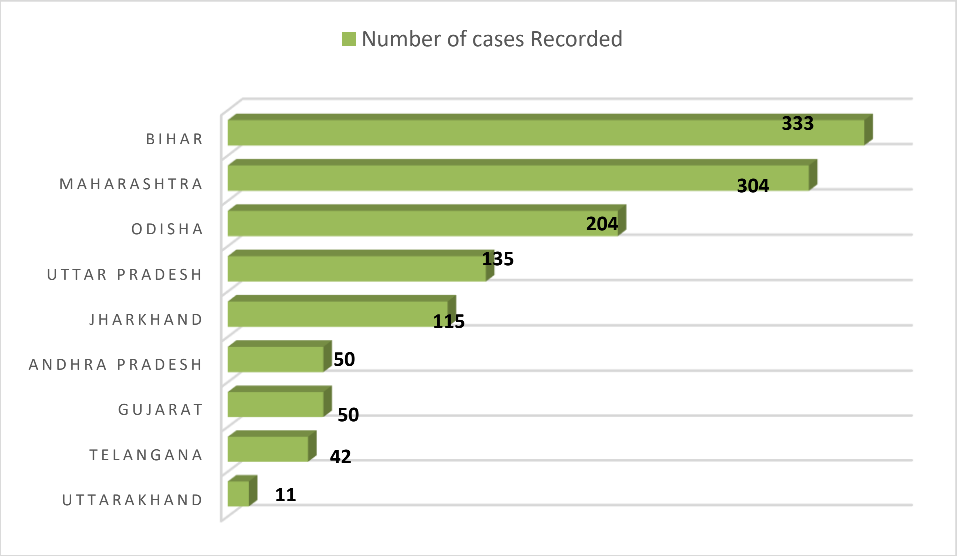


Figure 1 ATM FRAUD STATEWISE GRAPH

DOI 10.17148/IJARCCE.2021.10209

II. PROPOSED SYSTEM

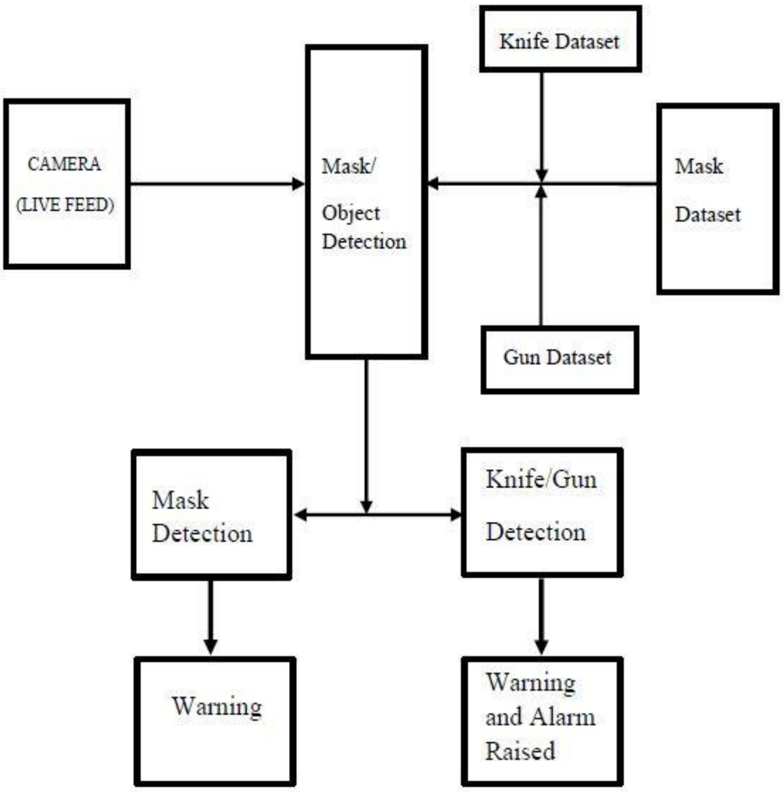


Figure 1 BLOCK DIAGRAM

Here we present the theory of the ATM-robbery prevention using machine learning technique the overall block diagram of the proposed system is explained. Each block of the block diagram is explained.

III. OBJECT DETECTION

A. Image Processing

The image processing is the process of detection of the real-world object like a bottle, helmet, car etc. The image processing is also used in the detection of the faces, eyes, mouth etc. The image processing is done from the images, video and also in the live stream data. this image processing application is commonly used in the security, survey etc.

B. Haar Cascade

Haar Cascade classifiers are an effective way for object detection. This method was proposed by Paul Voila and Michael Jones in their paper.

Rapid Object Detection using a Boosted Cascade of Simple Features .Haar Cascade is a machine learning-based approach where a lot of positive and negative images are used to train the classifier.

- Positive images – These images contain the images which we want our classifier to identify.
- Negative Images – Images of everything else, which do not contain the object we want to detect.

C. Object detection workflow

Every single Object Detection Algorithm has a different way of operational, but they all work on the same outlook. Feature Abstraction, They extract features from the input images at indicators and use these features to regulate the class of the image. Be it through Mat Lab, Open CV, Viola Jones or Deep Learning.

DOI 10.17148/IJARCCE.2021.10209

In Haar Cascade, the algorithm has four stages:

1. Haar Feature Selection
2. Creating Integral Images
3. Adaboost Training
4. Cascading Classifiers

Initially, the algorithm needs a lot of positive images of faces and negative images without faces to train the classifier. Then we need to extract features from it.

First step is to collect the Haar Features. A Haar feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums.

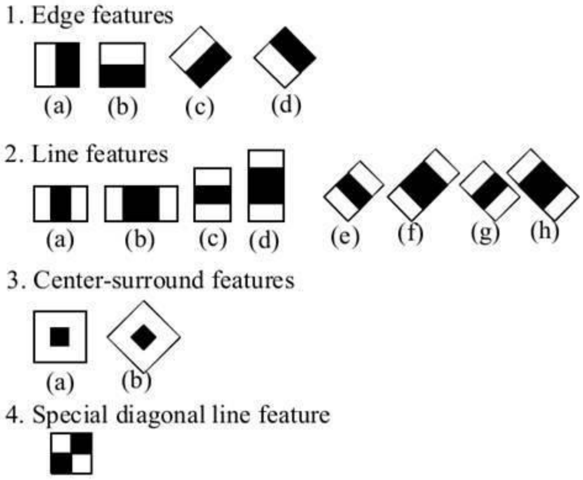


Figure 3 INTEGRAL IMAGES

Features:

Basically for face detection, the classifier looks for the most relevant features on the face such as eyes, nose, lips, forehead, eyebrows because we know that although people have different looks, these features are in the similar positions on the face.

Now if a new input image is given to the classifier, it compares the Haar Features from the xml file and applies it to the input image. If it passes through all the stages of haar feature comparison, then it's a face, else not.

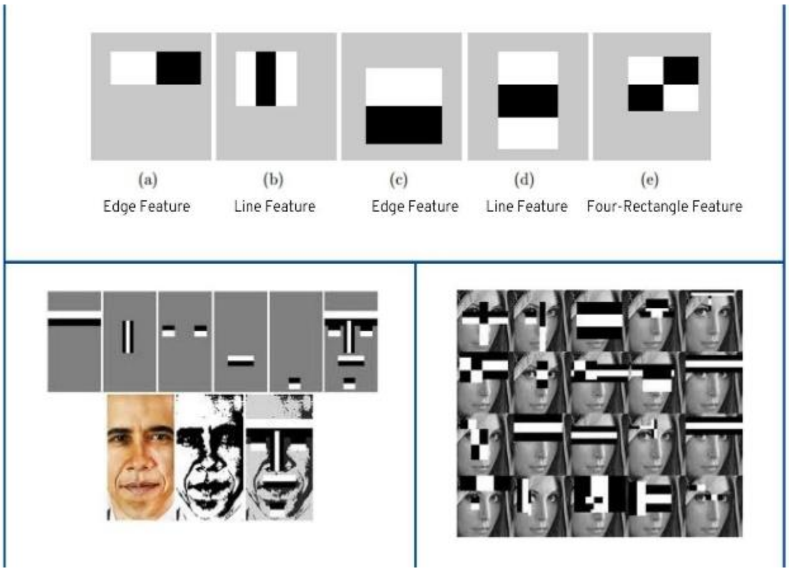
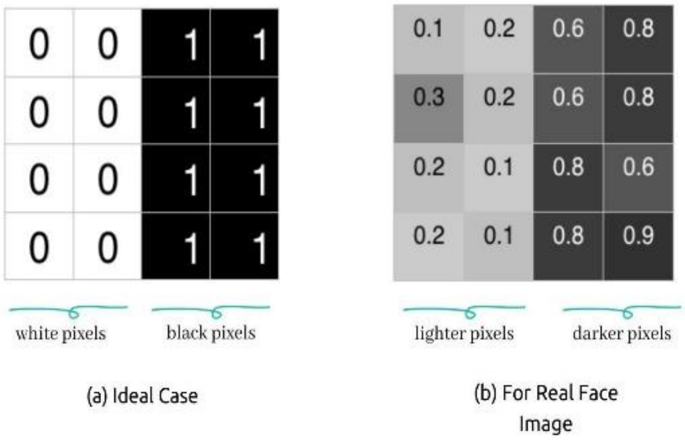


Figure 2 HAAR CASCADE FEATURES



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

(a) Ideal Case

0.1	0.2	0.6	0.8
0.3	0.2	0.6	0.8
0.2	0.1	0.8	0.6
0.2	0.1	0.8	0.9

(b) For Real Face Image

Figure 5 Pixel intensities for an ideal case and for real face image

D. Yolo V4

There is a very large amount that has been shown that there is an imprint of the Convolutional Neural Network (CNN) accuracy. Practical testing of combinations of such features on large datasets, as well as the theoretical justification of results, is required. Some features operate on certain models in particular and also for certain problems in particular, orly for small-scale datasets; during all features, such as batch-normalization and residual-connections, or arrlicable to majority of models, tasks, and datasets.

Statistical Analysis

We used Convolutional Neural Network to get our job done. The YOLOv4 is a Convolutional Neural Network that serves our purpose well. Our preferred model has 106 layers. 75 of them are convolutional layer and 31 of them are other layers. The model is FCN or Fully Convolutional Network. Means that there is no dense layer like typical CNN or Convolutional Neural Network. The other 31 layers are nor dense layer. They are shortcut, route, upsampling and yolo layers. The final output of the network is a feature map. A 1x1 convolution layer is used to determine the final output feature map. As it is 1x1 convolution layer, the size of the feature map is exactly as the feature map of previous layer. The result is then interpreted asconsidering each element in the feature map as a cell in image at last layer. Each cell produces a fixednumber of bounding boxes. Each bounding box has certain number of attributes. They are the center coordinates, the dimensions, the objectness score and class confidences. So, each bounding box has 5 (center coordinates, dimensions and objectness score) + C (class

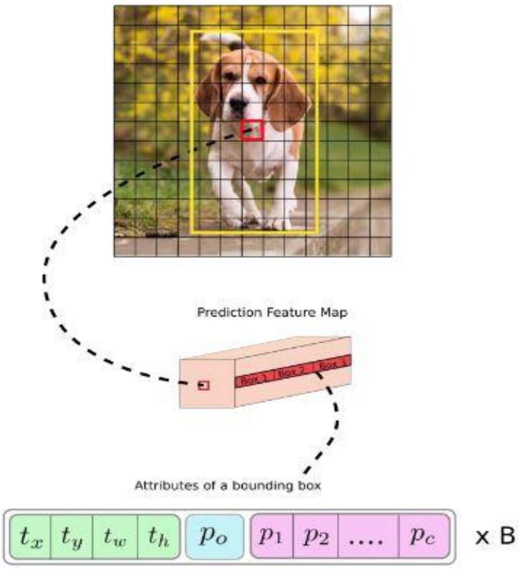


Figure 3 Yolo v4 Boundary Box

confidences)attributes.Let the size of the input image is 416 x 416 and the stride of the network is 32. Stride of the network act as a factor by which the out image is smaller than input image. So, the output image in this case will be 13 x 13.We can see the bounding boxes in figure 6.

The network has another extraordinary thing, the anchors. Anchors are the pre-defined bounding boxes of different sizes. We can determine the bounding box from scratch by convolution but this takes much more resource and time. But using anchors makes it simple. We just need to modify the anchors according to our needs. They are the sizes of the objects (width, height) in images that are converted to the network size (width, height in cfg). Formula showing in equations below is the rules for making bounding box predictions from network output.

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{tw}$$
$$b_h = p_h e^{th}$$

Here, bx, by, bw, bh are the co-ordinates of center, height and width prediction. tx, ty, tw, th is the network outputs. cx and cy are the top-left co-ordinates of the cell. pw and ph are anchors value of the bounding boxes.

E. Data Processing

Images are required in object detection training. Images must be the same size and images of the same size are provided as an image to enter in the training algorithm. The training algorithm detects images as it does all in-depth learning strategies, divides images into multiple pieces and processes each clip. After processing each piece of the image the processed image piece is converted into a dynamic and saved as a .model file. Stored variables are given different values for each variance. Flexible values are stored in the form in pickle format. Flexible values bearing the characteristics of professional images.

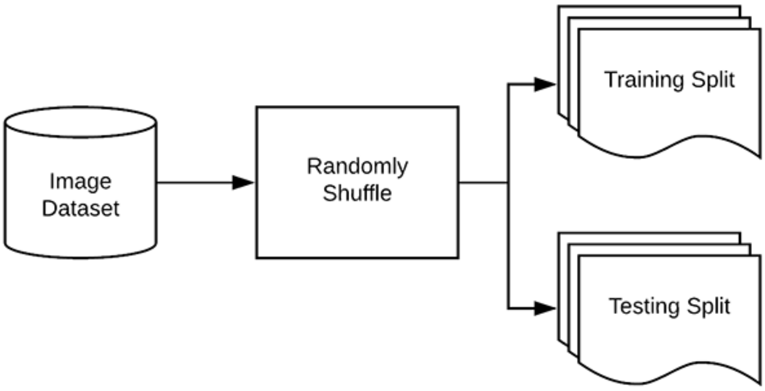


Figure 7 Training Data Split

F. Python Tkinter ATM GUI

Tkinter is one of the most popular Python GUI libraries for developing desktop applications. It’s a combination of the TK and python standard GUI framework.

Tkinter provides diverse widgets such as labels, buttons, text boxes, checkboxes that are used in a graphical user interface application.

The button control widgets are used to display and develop applications while the canvas widget is used to draw shapes like lines, polygons, rectangles, etc. in the application. Furthermore, Tkinter is a built-in library for Python, so you don’t need to install it like another GUI framework.

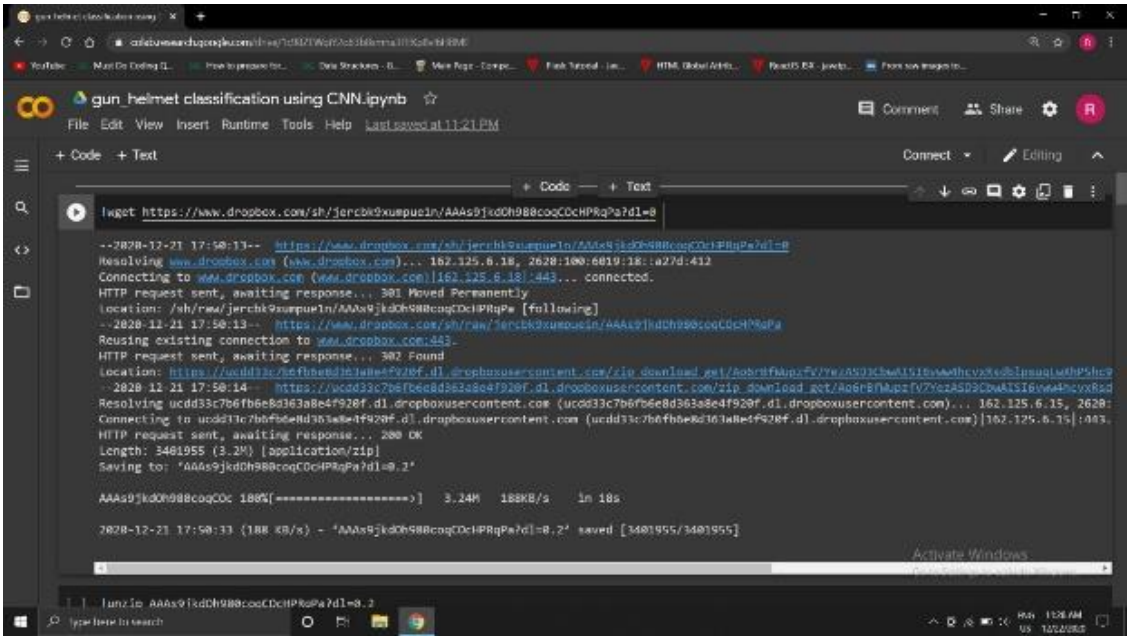
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 2, February 2021

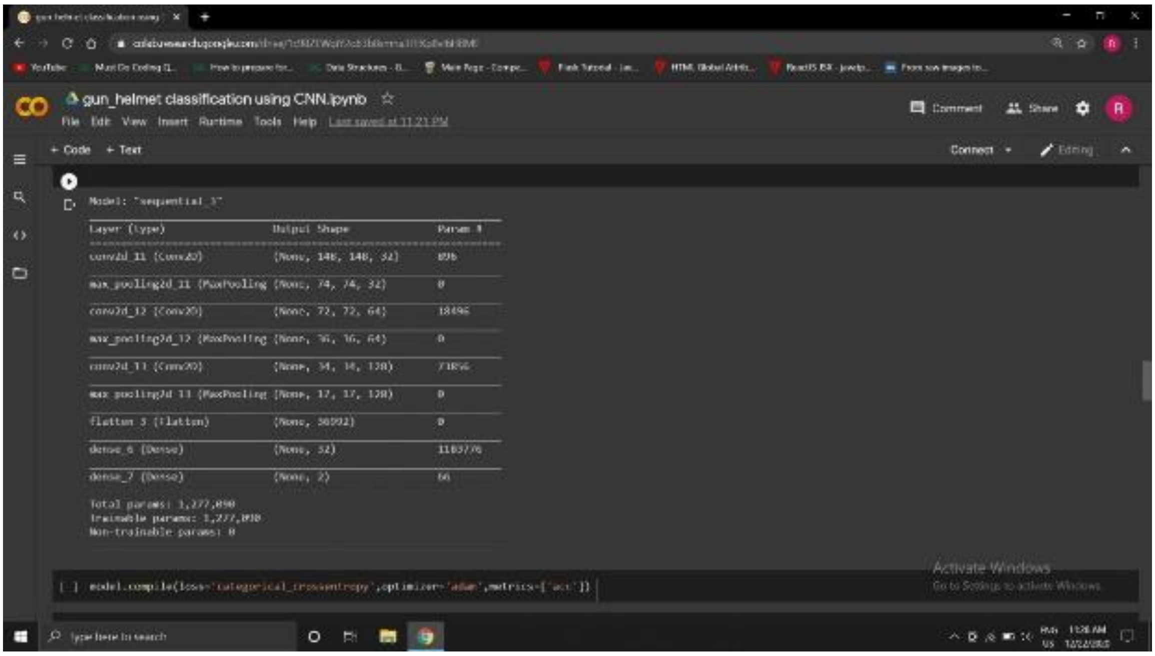
DOI 10.17148/IJARCCE.2021.10209

IV.TRAINING


gun_helmet classification using CNN.ipynb
File Edit View Insert Runtime Tools Help Last saved at 11:21 PM
+ Code + Text
Comment Share
Connect Editing

```
!wget https://www.dropbox.com/sh/jercbk9xumpue1n/AAAs9jkdOh988coqCOcHPRqPa?dl=0

--2020-12-21 17:50:13--  https://www.dropbox.com/sh/jercbk9xumpue1n/AAAs9jkdOh988coqCOcHPRqPa?dl=0
Resolving www.dropbox.com (www.dropbox.com)... 162.125.6.18, 2620:100:6019:18::a27d:412
Connecting to www.dropbox.com (www.dropbox.com)|162.125.6.18|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: /sh/raw/jercbk9xumpue1n/AAAs9jkdOh988coqCOcHPRqPa [following]
--2020-12-21 17:50:13--  https://www.dropbox.com/sh/raw/jercbk9xumpue1n/AAAs9jkdOh988coqCOcHPRqPa
Reusing existing connection to www.dropbox.com:443.
HTTP request sent, awaiting response... 302 Found
Location: https://ucdd33c7b6fb6e8d363a8e4f920f.dl.dropboxusercontent.com/zip_download_get/Ao6rBfWupzfV7YezASD3CbwAIS16vwwAhcvxRsdblpsuqLwhP5hc9
--2020-12-21 17:50:14--  https://ucdd33c7b6fb6e8d363a8e4f920f.dl.dropboxusercontent.com/zip_download_get/Ao6rBfWupzfV7YezASD3CbwAIS16vwwAhcvxRsd
Resolving ucdd33c7b6fb6e8d363a8e4f920f.dl.dropboxusercontent.com (ucdd33c7b6fb6e8d363a8e4f920f.dl.dropboxusercontent.com)... 162.125.6.15, 2620:
Connecting to ucdd33c7b6fb6e8d363a8e4f920f.dl.dropboxusercontent.com (ucdd33c7b6fb6e8d363a8e4f920f.dl.dropboxusercontent.com)|162.125.6.15|:443.
HTTP request sent, awaiting response... 200 OK
Length: 3401955 (3.2M) [application/zip]
Saving to: 'AAAs9jkdOh988coqCOcHPRqPa?dl=0.2'

AAAs9jkdOh988coqCOc 100%[===================>]   3.24M   188KB/s    in 18s

2020-12-21 17:50:33 (188 KB/s) - 'AAAs9jkdOh988coqCOcHPRqPa?dl=0.2' saved [3401955/3401955]

[ ] !unzip AAAs9jkdOh988coqCOcHPRqPa?dl=0.2
```

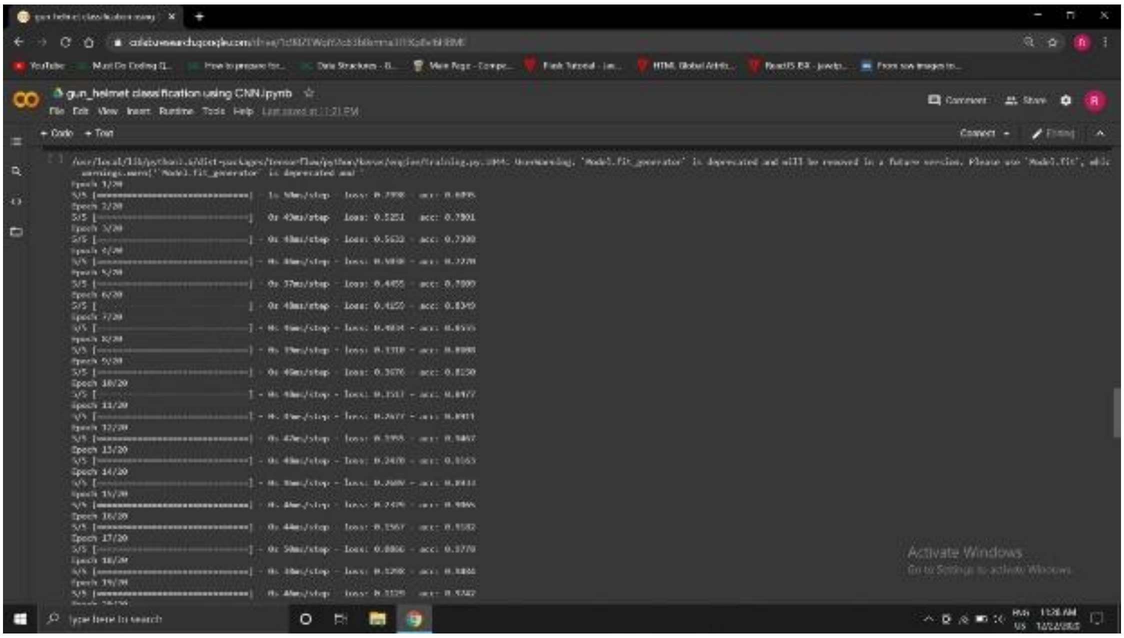
Figure 8 DATASETS


gun_helmet classification using CNN.ipynb
File Edit View Insert Runtime Tools Help Last saved at 11:21 PM
+ Code + Text
Comment Share
Connect Editing

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_11 (Conv2D)           (None, 148, 148, 32)      896
_________________________________________________________________
max_pooling2d_11 (MaxPooling (None, 74, 74, 32)        0
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 72, 72, 64)        18496
_________________________________________________________________
max_pooling2d_12 (MaxPooling (None, 36, 36, 64)        0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 34, 34, 128)       73856
_________________________________________________________________
max_pooling2d_13 (MaxPooling (None, 17, 17, 128)       0
_________________________________________________________________
flatten_3 (Flatten)          (None, 36992)             0
_________________________________________________________________
dense_6 (Dense)              (None, 32)                1183776
_________________________________________________________________
dense_7 (Dense)              (None, 2)                 66
=================================================================
Total params: 1,277,090
Trainable params: 1,277,090
Non-trainable params: 0

[ ] model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['acc'])
```

Figure 9 Model Building

DOI 10.17148/IJARCCE.2021.10209


gun_helmet classification using CNN.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
Comment Share
Connect
Editing
Activate Windows
Go to Settings to activate Windows.
Type here to search

```
/usr/local/lib/python3.6/dist-packages/tensorflow/python/keras/engine/training.py:1844: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which
  warnings.warn('`Model.fit_generator` is deprecated and '
Epoch 1/20
5/5 [==============================] - 1s 50ms/step - loss: 0.7998 - acc: 0.6095
Epoch 2/20
5/5 [==============================] - 0s 40ms/step - loss: 0.5251 - acc: 0.7501
Epoch 3/20
5/5 [==============================] - 0s 48ms/step - loss: 0.5633 - acc: 0.7300
Epoch 4/20
5/5 [==============================] - 0s 40ms/step - loss: 0.5018 - acc: 0.7270
Epoch 5/20
5/5 [==============================] - 0s 37ms/step - loss: 0.4455 - acc: 0.7800
Epoch 6/20
5/5 [==============================] - 0s 40ms/step - loss: 0.4150 - acc: 0.8349
Epoch 7/20
5/5 [==============================] - 0s 46ms/step - loss: 0.4034 - acc: 0.8555
Epoch 8/20
5/5 [==============================] - 0s 39ms/step - loss: 0.3310 - acc: 0.8088
Epoch 9/20
5/5 [==============================] - 0s 40ms/step - loss: 0.3676 - acc: 0.8150
Epoch 10/20
5/5 [==============================] - 0s 40ms/step - loss: 0.3517 - acc: 0.8477
Epoch 11/20
5/5 [==============================] - 0s 41ms/step - loss: 0.2677 - acc: 0.8911
Epoch 12/20
5/5 [==============================] - 0s 42ms/step - loss: 0.3995 - acc: 0.8467
Epoch 13/20
5/5 [==============================] - 0s 40ms/step - loss: 0.2470 - acc: 0.9163
Epoch 14/20
5/5 [==============================] - 0s 38ms/step - loss: 0.2689 - acc: 0.8933
Epoch 15/20
5/5 [==============================] - 0s 40ms/step - loss: 0.2479 - acc: 0.9066
Epoch 16/20
5/5 [==============================] - 0s 44ms/step - loss: 0.1567 - acc: 0.9502
Epoch 17/20
5/5 [==============================] - 0s 50ms/step - loss: 0.0866 - acc: 0.9770
Epoch 18/20
5/5 [==============================] - 0s 38ms/step - loss: 0.1298 - acc: 0.9466
Epoch 19/20
5/5 [==============================] - 0s 40ms/step - loss: 0.1129 - acc: 0.9742
```

Figure 10 Model Training

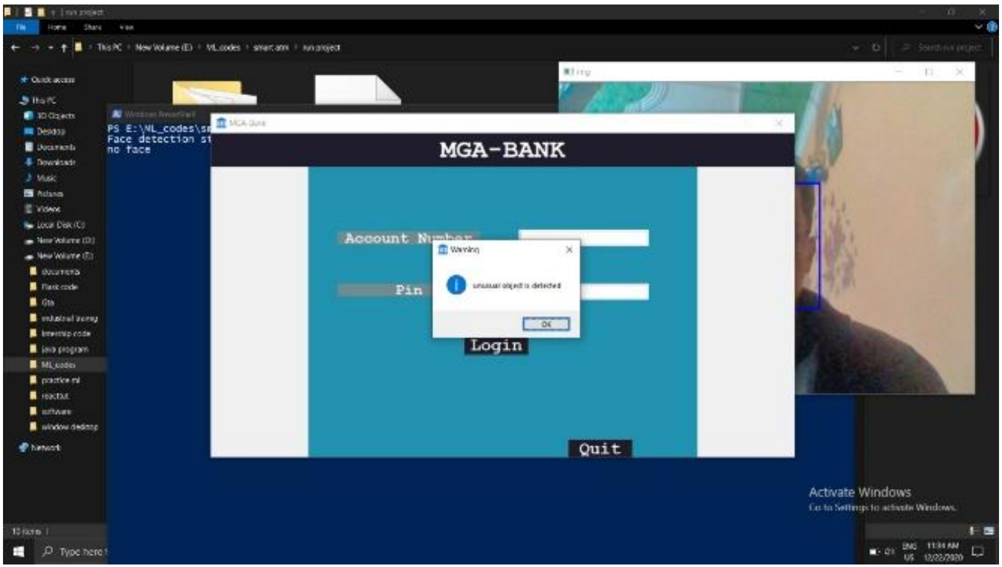
V. RESULT AND DISCUSSION

We implemented the model by using suitable on Yolo V4 and Haar Cascade Open CV with the importing of the packages keras.preprocessing.image for both ImageDataGenerater and img_to_array, keras. optimizers, sklearn.preprocessing, sklearn.model_selection, pyimagesearch.smallervggnet, matlabplot, imutils, numpy, argparse, random, pickle, OpenCV. These packages are used for grabbing the images from the sample and convert the images into the various segmented variables.

Each variable is given the values for the purpose to differentiate the variables form each other. The variables hold the features of the sample images that fetched to the training process.

The values of the testing images or the video are cut down into the frames and each frame is cut down into various parts and given values. The values are compared with the trained images value and detect and give accuracy and send mail while it matches the trained images.

VI. OUTPUT



```
PS E:\ML_codes\sm
Face detection st
no face
```

unusual object is detected

Figure 4 OUTPUT

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 2, February 2021

DOI 10.17148/IJARCCE.2021.10209

VII.CONCLUSION AND FUTURE SCOPE

Implementing the object detection in the ATM security by detecting harmful objects by using the Haar Cascade with the Yolo V4 is the best model that will take only smaller time for the training the images for the detection. It has the best advantage of giving the perfect Accuracy percentage than the training cascade with the Cascade trainer GUI. This model can be used for training a greater number of objects of the user need. This model can be implemented in various real-time factors for various purposes.

VIII.FUTURE SCOPE

- In future, more images can be trained.
- Can be implemented many government purposes
- Can be used for the survey prediction of people
- Can be added some more modules for more security purposes.

REFERENCES

[1] Jubaer, A. N. M.; Sayem, Abu "Real time Weapon Detection Using Convolutional Neural Network", Department of CSEDaffodil International University, (2020)

[2] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao" YOLOv4: Optimal Speed and Accuracy of Object Detection", Computer Vision and Pattern Recognition (cs.CV); Image and Video Processing (eess.IV),(2020)

[3] https://www.geeksforgeeks.org/yolo-you-only-look-once-real-time-object-detection/?ref=lbp

[4] https://docs.opencv.org/master/dc/d88/tutorial_traincascade.html

[5] https://docs.python.org/3/library/tkinter.html

[6] Soo, Sander. "Object detection using Haar-cascade Classifier." (2014).

[7] Jose L. Salazar González, Carlos Zaccaro, Juan A. Álvarez-García, Luis M. Soria Morillo, Fernando Sancho Caparrini,Real-time gun detection in CCTV: An open problem,Neural Networks,Volume 132,(2020)

[8] Springer, Cham "Gun and Knife Detection Based on Faster R-CNN for Video Surveillance",(2019