

# Implementation of Malicious Things Detection at Public Places Using Deep Learning

P. Aleemulla Khan, N. Thirupathi Rao, Debnath Bhattacharyya

**Abstract:** To provide effective security in crowded or public areas in today's world is a big challenge for us. One of the major challenges is to detect or monitor potential threats such as explosive items or bombs (Abandoned luggage items). In this paper we propose an approach for automatic detection of abandoned luggage and alerting the security alliances. We use deep learning to train the system with a set of images, these images were given to the trained system which is going to visualize the objects in the image and calculate the distance between objects if the object is person and baggage or only baggage. If the distance is greater than a threshold distance limit then the system is going to raise an alarm for the security alliances.

**Key Words:** Explosive items, Deep Learning, Security alliances

## I. INTRODUCTION

Now a day's providing security in the crowded or public areas and usage of video cameras to capture the objects continuously is a huge task, video cameras are not having the intelligence to classify the objects as well as humans. To enable security to be very effective the security people should be increased and continuous monitoring is required which is very impossible. To make people more secured with more powerful security alliances and reducing the human efforts, in this paper we propose a machine which adds intelligence to the video data captured from the security cameras. We train the system with a set of images and then the input data which is video into the one image per one second. These images were given to the trained system which is going to visualize the objects in the image and calculate the distance between objects in case if the object is person and baggage in the specified time frame. If the object is baggage alone or distance between person and baggage keeps on increasing than the specified threshold distance and for a specified time gap the system will raise an alarm for the security alliances.

### A. Deep Learning

- Deep Learning is a subfield of machine learning worried about calculations inspired by the structure and function of the mind called simulated neural systems.
- Deep learning is utilized to recognize objects from satellites and distinguish sheltered or risky zones for troops.

**Revised Manuscript Received on May 06, 2019**

**P. Aleemulla Khan**, <sup>1</sup>Department of Computer Science and Engineering, Vignan's Institute of Information Technology, Visakhapatnam-530049, India

**N. Thirupathi Rao**, <sup>1</sup>Department of Computer Science and Engineering, Vignan's Institute of Information Technology, Visakhapatnam-530049, India

**Debnath Bhattacharyya**, <sup>1</sup>Department of Computer Science and Engineering, Vignan's Institute of Information Technology, Visakhapatnam-530049, India

- Automotive scientists are utilizing profound figuring out how to naturally recognize protests, for example, stop signs and activity lights.
- Deep learning is being utilized as a part of computerized hearing and discourse interpretation.

## B. NEURAL NETWORKS

An Artificial Neural Network (ANN) is a prominent method and technology used to process many applications. Utilizing video observation cameras, lawbreakers have been captured and fear monger assaults have been stayed away from. Video observation cameras must be enhanced keeping in mind the end goal to counteract wrongdoing and fear based oppression effectively. A perfect video observation framework for surrendered gear discovery must have the capacity to identify uncommon occasions, to recognize a man and a baggage. In a large number of the confined spots security ought to be extremely alarm and continue checking the regions for unattended stuff. The procedure isn't mechanized and required part of human exertion.

A deficiency of the manual video reconnaissance framework in the general population transportation territory is that the security staff can't watch all suspicious conduct caught by the camcorders and checked on the PC screen. Manual framework should be supplanted by a robotized framework which will alert the security staff when relinquished baggage is identified or caution the explorer when he or she leaves the gear unattended. This proposed framework goes for identifying relinquished gear at the prepare station and its encompassing. It will be built in three distinctive User Interface variants for three various types of clients.

## C. Image Annotation

There are three types of learning methodologies in machine learning. They are:

- Supervised learning:** In supervised learning we will give the labels for the image and make the system learn different images of same feature.
- Unsupervised learning:** In unsupervised learning we will give the features for a system to understand and name the system according to that features is done by the system.
- Semi-supervised learning:** In semi-supervised learning we have the images both with labels and without labels.

The annotation is mainly to label the images and use the supervised learning mechanism. The annotation is of two types:

- Manual Annotation
- Automatic Annotation

## II. LITERATURE REVIEW

A review paper by Kevin Smith [1] gives a component to follow protests in the scene utilizing a trans-dimensional Markov Chain Monte Carlo following model appropriate for use in nonexclusive mass following undertakings. A paper by Uijlings [2] present particular hunt which joins the quality of both a thorough inquiry and division specific pursuit empowers the utilization of the ground-breaking Bag-of-Words model for acknowledgment. A paper by Ren [4] centers on a Region Proposal Network (RPN) that shares full-picture convolutional highlights with the location arrange.

A paper by Calvin Choy [7] executes a framework which will almost certainly distinguish deserted gear utilizing the caught pictures or recordings from the camera as the contribution of the framework. A paper by Redmon [5] outline object location as a regression issue to spatially isolated bouncing boxes and related class probabilities. However it battles to sum up to items in new or abnormal viewpoint proportions or arrangements.

## III. PROPOSED WORK

Making a system to detect the objects in the images that were been collected from the video with in a time frame and measure the distance if the objects are person and the baggage continuously for threshold time period and if the distance keeps on increasing or image has only bag as object then raise an alarm.

The Purpose of the Project is to create an application which performs object detection based upon the Single shot multibox detector algorithm (SSD). The main purpose of the project is to provide a platform for the security alliances which alert them with an alarm on recording suspicious scene.

### A. SSD ALGORITHM

The name of this architecture comes from:

- **Single Shot:** This means that the tasks of object localization and classification are Done in a *single forward pass* of the network.
- **MultiBox:** This is the name of a technique for bounding box regression developed by Szegedy et al.
- **Detector:** The network is an object detector that also classifies those detected objects

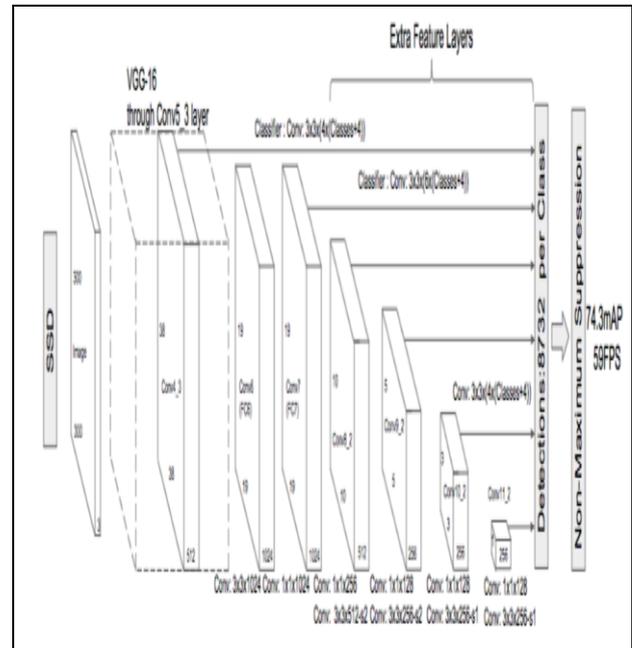


Fig 1. SSD Structure

### B. MULTIBOX

The bounding box relapse system of SSD is propelled by Szegedy's work on MultiBox, a technique for quick class-freethinker bounding box facilitate proposals. MultiBox's misfortune work additionally joined two basic segments.

- **Confidence Loss:** These measures how certain the system is of the objectless of the figured bounding box.
- **Location Loss:** This measures how far away the system's anticipated jumping boxes are starting from the earliest stage ones from the preparation set.

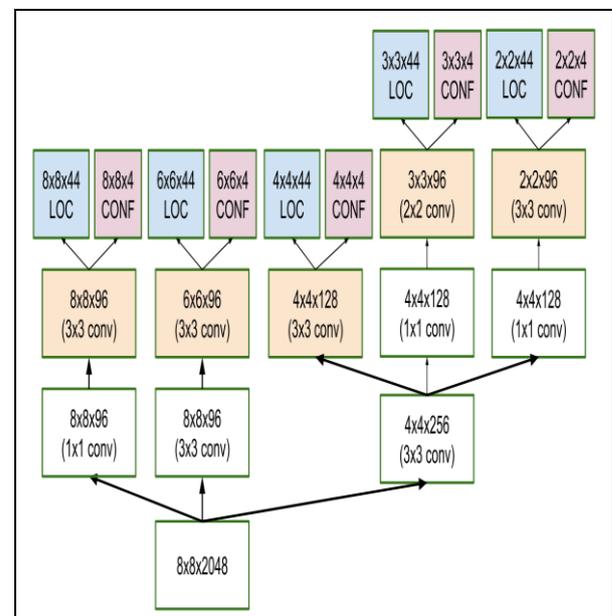


Fig 2. Multi-Box Structure

### C. WORKING MECHANISM

1. The below procedure to Detect and calculation of distance between objects using trained model known as Spyder. The mechanism is as follows,

Step-1: Import all necessary libraries.

Step-2: A video is divided into frames and stored in test directory.

Step-3: Load the graph of training model.

Step-4: Load the labels of the training model.

Step-5: Give the locations of the test directory images.

Step-6: Calculate distance between objects and if distance exceeds then blow an alarm for security.

• The below diagram shows that workflow of the proposed system for Detect and calculation of distance between objects.

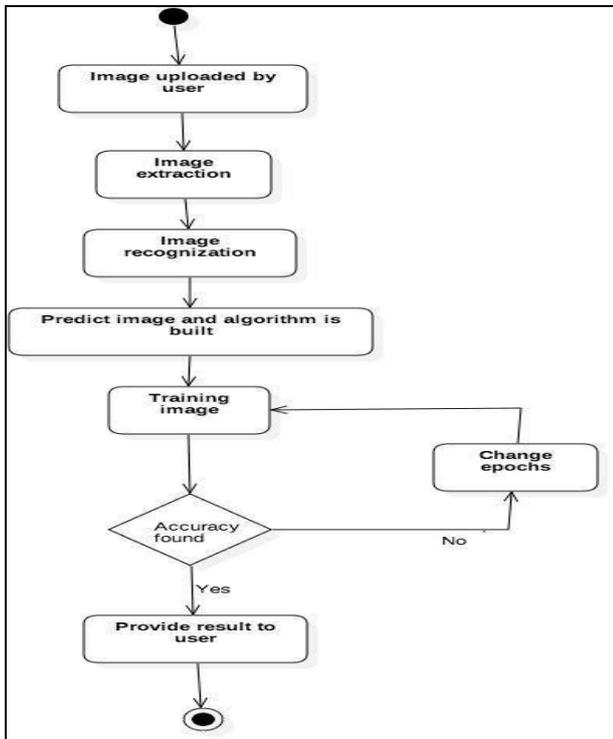


Fig 3. Working Mechanism

### D. BAGGAGE DETECTION

On March 11, 2004, ten bombs in backpacks and other little sacks, for instance, duffel packs, exploded on four traveller trains. Agents found a set up bomb in a duffel sack they accepted that was ready one of the trains. The sack, with the bomb in it, was found at a police base camp, after it had been taken to the broad fleeting burial service home. Inside they found a Spanish-made shaky, a detonator with a cell phone and a clock. The flimsy found secured was not of the sort commonly used by ETA, yet rather was somewhat increasingly "current" version of ETA's standard unstable. The touchy found clinched was not of the sort generally utilized by ETA, yet was rather progressively "current" variant of ETA's standard explosive.

i. There are security people in the busy places and the video cameras are capturing the scenes continuously. But the cameras are not having the intelligence to classify the objects as well as humans should be very fast and careful.

- ii. To enable security to be very fast the security people should be increased and continuously stay with the person from entrance to exit i.e; one passenger-one security which is impossible.
- iii. To make people more secured with powerful security alliances by reducing the human efforts we propose a machine which adds intelligence to the video data captured from the security cameras. We train the system with a set of images and then the input data which is video into the one image per one second.
- iv. These images were given to the trained system which is going to visualize the objects in the image and calculate the distance between objects in case if the object is person and baggage for all the images in the specified time frame.
- v. If the object is baggage alone or distance between person and baggage keeps on increasing than the specified threshold distance and for a specified time gap the system will raise an alarm for the security alliances.

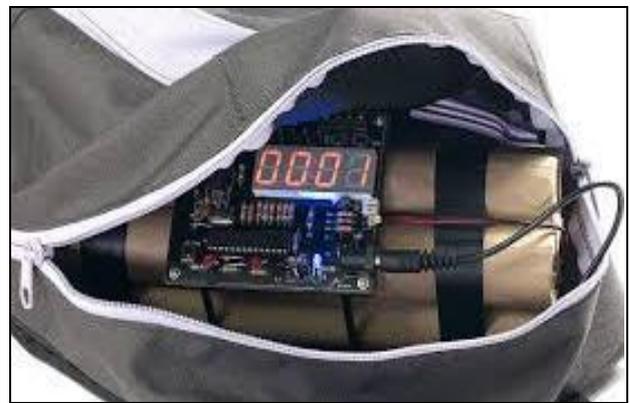


Fig 4. Baggage Detection Model

### IV. RESULTS & ANALYSIS

To work with huge amount of graphical and video information normal server not support or it takes much time to perform the operations. We need the graphical processing units to process the Tera bytes of video or graphical data quick and good performance. Hence here we used to process the video data with help of Nvidia GPU graphic server.

A Graphical Processing unit (GPU) is a specific electronic circuit proposed to rapidly control and alters memory to animate the arrangement of pictures in a packaging support got ready for respect a show device. Present day GPUs are incredibly capable at controlling PC delineations and picture getting ready, and their exceedingly parallel structure makes them more profitable than all around valuable CPUs for counts where the treatment of tremendous squares of data is done in parallel.

The Execution and Results including the following procedure and steps, they are as follows:

Step-1: Installing the Anaconda.

Step-2: Installing the Environment construction procedure.

Step-3: Installing the Python usage and versions.

Step-4: Installing the Python in Anaconda.

Step-5: Installing the Deep Learning libraries.



# Implementation of Malicious Things Detection at Public Places Using Deep Learning

Step-6: Running anaconda cmd prompt.

Step-7: Execution of training Data.

Step-8: Execution of validation.

Step-9: Final Results.

The figure below shows the process of installation o required softwares to handle the data coming from various sources,

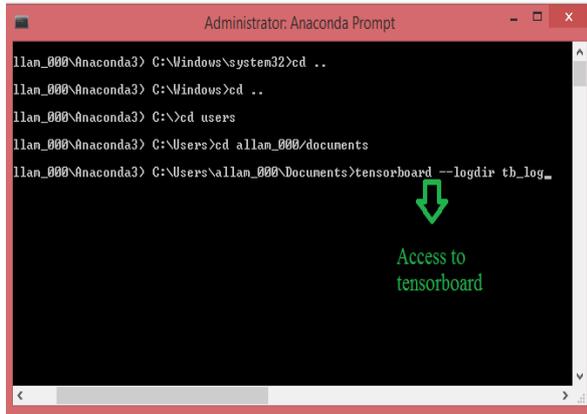


Fig 5. Accessing Tensor board of Unattended Baggage Detection

The calculations you'll utilize Tensor Flow for - like preparing an enormous profound neural system - can be intricate and befuddling. To make it less demanding to comprehend, investigate, and improve Tensor Flow projects, we've incorporated a suite of perception apparatuses called Tensor Board. You can utilize Tensor Board to imagine your Tensor Flow chart, plot quantitative measurements about the execution of your diagram, and demonstrate extra information like pictures that go through it. At the point when Tensor Board is completely designed, it would seem that this:

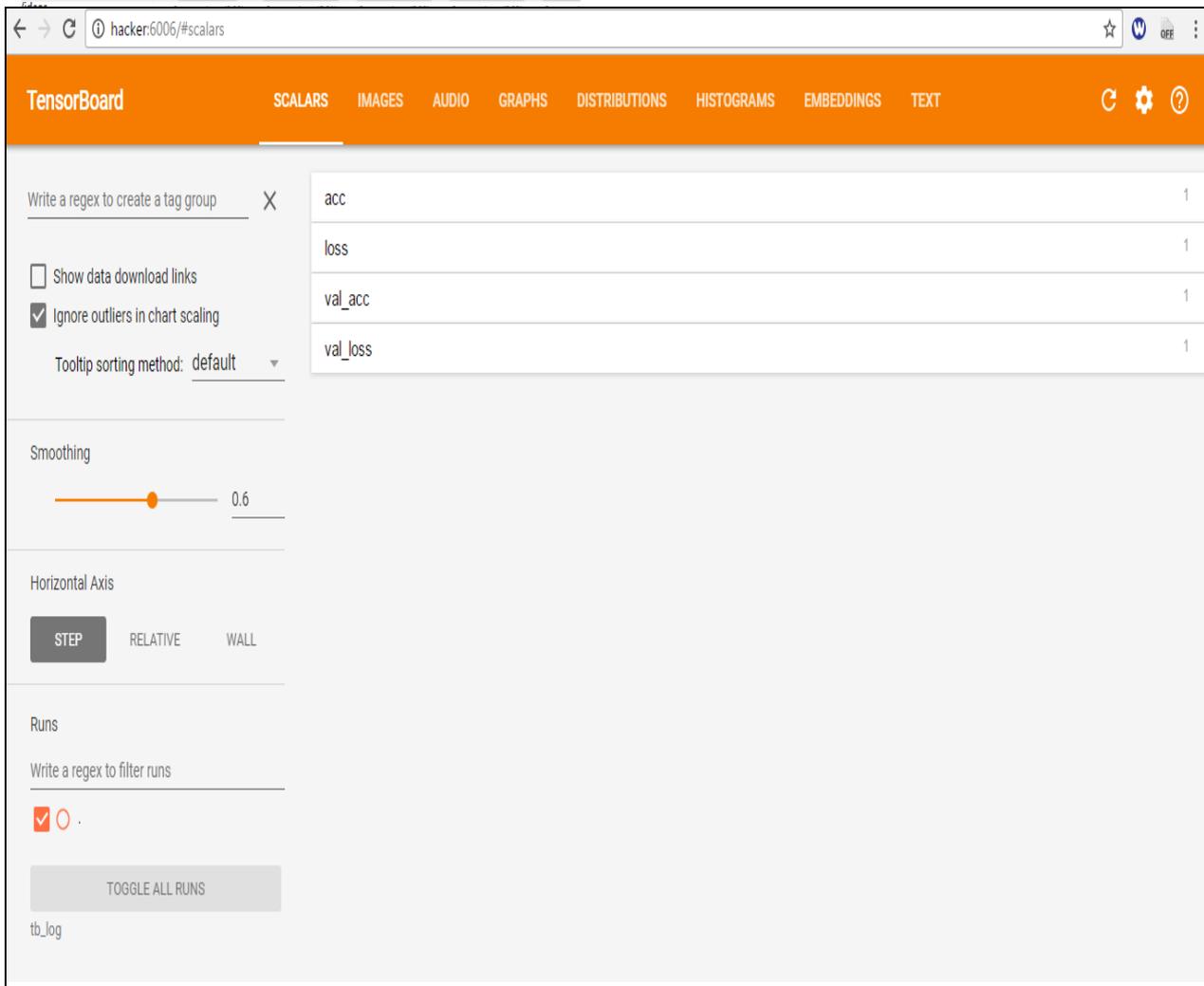


Fig 6. Tensor Board Console of Unattended Baggage Detection

```

175     use normalized_coordinates=True,
176         line_thickness=8)
177     width,height=image.size
178     ymin = boxes[0][0][0]*height
179     xmin = boxes[0][0][1]*width
180     ymax = boxes[0][0][2]*height
181     xmax = boxes[0][0][3]*width
182     ymin1 = boxes[0][1][0]*height
183     xmin1 = boxes[0][1][1]*width
184     ymax1 = boxes[0][1][2]*height
185     xmax1 = boxes[0][1][3]*width
186     print(ymin,xmin,ymax,xmax)
187     print(ymin1,xmin1,ymax1,xmax1)
188     distance=xmin-xmax1
189     print(distance)
190     category=[(category_index.get(value)).get('name') for ind
191     print(len(category))
192     if(category[0]=='backpack'or'person'):
193         if(len(category)>=2):
194             if(category[0]=='person' or category[1]=='person'
195                 if(distance)>=125.00000000):
196                 flag=flag+1;
197                 if(flag==10):
198                     plt.figure(figsize=IMAGE_SIZE)
199                     plt.imshow(image_np)
200                     winsound.Beep(2500, 1000)
201             elif(len(category)<2):
202                 if(category[0]!='person'):
203                     count=count+1
204                 if(count==5):
205                     plt.figure(figsize=IMAGE_SIZE)
206                     plt.imshow(image_np)
207                     winsound.Beep(2500, 1000)
208
209
210
211
...:     line_thickness=8)
...:     width,height=image.size
...:     ymin = boxes[0][0][0]*height
...:     xmin = boxes[0][0][1]*width
...:     ymax = boxes[0][0][2]*height
...:     xmax = boxes[0][0][3]*width
...:     ymin1 = boxes[0][1][0]*height
...:     xmin1 = boxes[0][1][1]*width
...:     ymax1 = boxes[0][1][2]*height
...:     xmax1 = boxes[0][1][3]*width
...:     print(ymin,xmin,ymax,xmax)
...:     print(ymin1,xmin1,ymax1,xmax1)
...:     distance=xmin-xmax1
...:     print(distance)
...:     category=[(category_index.get(value)).get('n
index,value in enumerate(classes[0]) if scores[0,index] >
...:     print(len(category))
...:     if(category[0]=='backpack'or'person'):
...:         if(len(category)>=2):
...:             if(category[0]=='person' or
category[1]=='person'):
...:                 if(distance)>=125.00000000):
...:                     flag=flag+1;
...:                     if(flag==10):
...:                         plt.figure(figsize=IMAGE
...:                         plt.imshow(image_np)
...:                         winsound.Beep(2500, 1000)
...:                 elif(len(category)<2):
...:                     if(category[0]!='person'):
...:                         count=count+1
...:                     if(count==5):
...:                         plt.figure(figsize=IMAGE_STZ
...:                         plt.imshow(image_np)
...:                         winsound.Beep(2500, 1000)
583.578128815 1728.73153687 1494.62654114 2503.29147339
21.5084838867 0.0 1907.56736755 1041.71066284
687.020874023
2
347.977151871 216.033461094 3809.77401733 2692.58168221
    
```

These are the dimensions of the boundary box.

Fig 7. Cell by Cell execution in Spyder

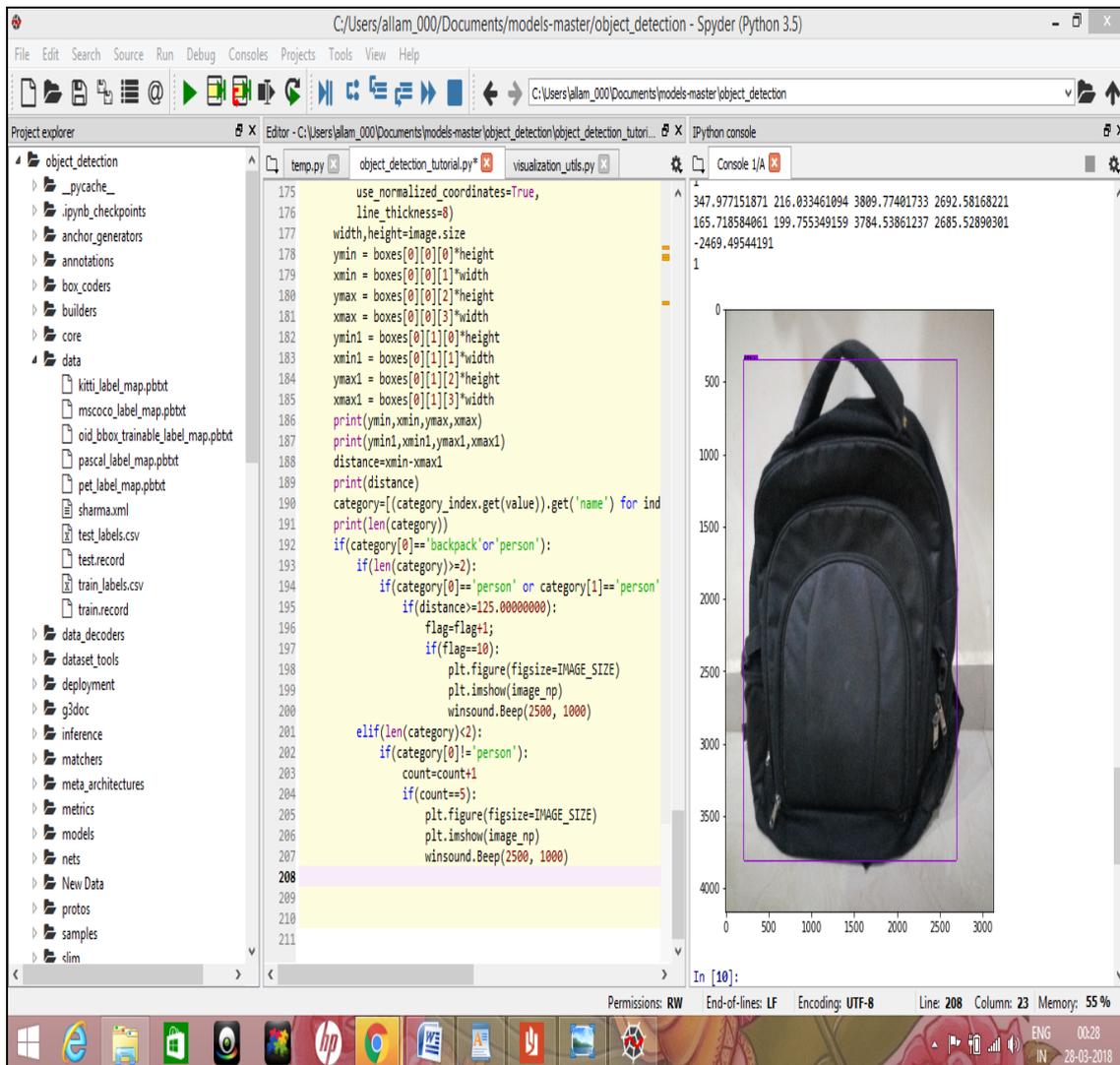


Fig 8. Showing output in Ipython Console

## V. CONCLUSION

To make it less demanding to comprehend, investigate, and improve Tensor Flow projects, we've incorporated a suite of perception apparatuses called Tensor Board. You can utilize Tensor Board to imagine your Tensor Flow chart, plot quantitative measurements about the execution of your diagram, and demonstrate extra information like pictures that go through it. At the point when Tensor Board is completely designed. To enable security to be very fast the security people should be increased and continuously stay with the person from entrance to exit i.e; one passenger-one security which is impossible. To make people more secured with powerful security alliances by reducing the human efforts we propose a machine which adds intelligence to the video data captured from the security cameras. We train the system with a set of images and then the input data which is video into the one image per one second. These images were given to the trained system which is going to visualize the objects in the image and calculate the distance between objects in case if the object is person and baggage for all the images in the specified time frame.

Today, technology is robust and the automation is been increasing in many of the industries to reduce the man efforts and the human mistakes. The systems are acting smart upon training and visualization is one such essential that has a capacity of recognition of objects. Automatized providing the system with efficient training for detecting the objects and calculating distance between those objects may help the security alliance to do more best job with less efforts. The solution that we make a system to detect the objects like person or backpack and calculate distance between the bounding boxes if both objects are present in the image. If the distance keeps on increasing for a particular time frame than the system raise an alarm for security alliances to alert them. If the frame has only backpack then the system observes it for threshold time limit and raise alarm if no person attends the bag.

## REFERENCES

1. G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
2. Kevin Smith, Pedro Quelhas, Daniel Gatica-Perez, "Detecting Abandoned Luggage Items in a Public Space", IEEE Performance Evaluation of Tracking and Surveillance Workshop (PETS), New York, June 18, 2006, pp. 75-82.
3. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector", European Conference on Computer Vision – ECCV 2016, Amsterdam, The Netherlands, October 11–14, 2016, pp. 21-37.
4. J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, A.W.M. Smeulders, "Selective Search for Object Recognition", International Journal of Computer Vision, September 2013, Volume 104, No.2, pp. 154–171.
5. Ren, S., He, K., Girshick, R., Sun, J, "Faster R-CNN: Towards real-time object detection with region proposal networks", International Conference on Advances in Neural Information Processing Systems 28 (NIPS 2015).
6. Redmon, J., Divvala, S., Girshick, R., Farhadi, A, "You only look once: Unified, real-time object detection", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
7. Ceshine Lee, A Tutorial on "Towardsdatascience.com/learning-note-single-shot-multibox-detector-with-pytorch-part-3-f0711caa65ad", July, 2017. [Last accessed on 10-4-2019]
8. Calvin Choy, May Chung, Gherry Harahap, Meriane Natadarma, Sin Lin Wu, "Surveillance System Using Abandoned Luggage Detection", An UG Thesis submitted to Man-Machine Interaction Group, Faculty of EEMCS, Delft University of Technology, July, 2007.
9. Clark, Don, J.P. Morgan, "Benefits from Chip Change", WSJ Digits Blog Retrieved September 14, 2011. [Last accessed on 10-4-2019]
10. O. M. Parkhi, A. Vedaldi, A. Zisserman, "Deep Face Recognition", British Machine Vision Conference, 2015.
11. Face Recognition Database found at <https://www.kairos.com/blog/60-facial-recognition-databases> May 7, 2015. {Last accessed on 10-03-2019}.

Recognition, Bio-Informatics, Computational Biology, Evolutionary Computing and Security. He published 200+ research papers in various reputed International Journals and Conferences. He published six textbooks for Computer Science as well. He is the member of IEEE, ACM, ACM SIGKDD, IAENG, and IACSIT.

## AUTHORS PROFILE



**Mr. P. Aleemullah Khan** received B.Tech and M.Tech from Jawaharlal Nehru Technological University, Kakinada, AP. He is currently associated with Vignan's Institute of Information Technology, Visakhapatnam-530049, India as Assistant Professor and Asst. HoD of Computer Science and Engineering of the Institute since the year 2014. His research areas include Communication Networks, Queuing Models, Stochastic Modeling, Image Processing, Pattern Recognition, Bio-Informatics, Evolutionary Computing and Security. He published 4+ research papers in various reputed International Journals and Conferences. He is the member of IE and CSI.



**Dr. N. Thirupathi Rao** received PhD (Tech., CSE) from Andhra University, Visakhapatnam, India. Currently, Dr N.Thirupathi Rao associated with Vignan's Institute of Information Technology, Visakhapatnam-530049, India as Associate Professor and Asst. HoD of Computer Science and Engineering of the Institute since the year 2016. His research areas include Communication Networks, Queuing Models, Stochastic Modeling, Image Processing, Pattern Recognition, Bio-Informatics, Evolutionary Computing and Security. He published 45+ research papers in various reputed International Journals and Conferences. He is the member of ACM, IE, CSI, and ISPS.



**Dr. Debnath Bhattacharyya** received PhD (Tech., CSE) from University of Calcutta, Kolkata, India. Currently, Dr Bhattacharyya associated with Vignan's Institute of Information Technology, Visakhapatnam-530049, India as Head of Computer Science and Engineering and Dean R&D of the Institute since the year 2015. His research areas include Image Processing, Pattern

