## horizontal line



[Detecting Forbidden Items](https://github.com/mennaAyman/Detecting_Forbidden_Items)

25.04.2020

# •Introduction and History

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly.

# •Problem Definition



Whenever you’re travelling, you always see an x-ray device at airports, the purpose of airport security screenings is to prevent dangerous goods from entering aircraft and to ensure that all flights reach their destinations safely. Therefore, all passengers must walk through a metal detector before entering the gate area, and both hand baggage and checked-in baggage are scanned with an X-ray machine.

From a passenger’s point of view, an X-ray scanner looks like a boxy tunnel and a conveyor belt, which moves the baggage through the tunnel. Inside the box, the baggage is scanned with X-rays: scanning is enabled by the fact that X-rays penetrate different substances to a different extent.

The machine produces X-rays with a special tube, which is lined with lead. In the lead lining there is a narrow, around one-centimeter-wide gap through which the X-rays are directed into the tunnel. The conveyor belt carries each piece of baggage through the X-ray beam, and on the opposite side of the tunnel, a detector measures the amount of radiation which has penetrated the scanned item. Dense substances, such as lead, absorb the most radiation, blocking the X-rays’ progression.

Based on the amount of radiation which has passed the piece of baggage, a computer forms a close to real-time image of the items, an airport employee then takes a look at the image and decides whether this baggage could enter the aircraft or not.  
In our project we aim to make a machine learning software which completes the process automatically without the need of human intervention or assistance.

# 

# •Objective

The objective of this project is to detect, prevent, or mitigate terrorist threats in relation to trivial amounts of explosives and personal weapons.

So this project aims to develop a comprehensive software to approach airport security checkpoints and detect any forbidden item in baggage, by analyzing X-ray images coming from the airport baggage scanner.

It will help ensure robust and controllable security performance, also ensure reliability, speed, cost-effectiveness, and ease-of-use.

# 

# •Related Work

***DataSet:*** Our dataset, named SIXray, consists of 1,059,231 X-ray images, in which 6 classes of 8,929 prohibited items are manually annotated. It raises a brand new challenge of overlapping image data, meanwhile shares the same properties with existing datasets, including complex yet meaningless contexts and class imbalance.

- They used two representative CNN object detection models, Faster R-CNN and RetinaNet .

Faster R-CNN is an object detection algorithm which is the combination of its predecessor Fast R-CNN and Region Proposal Network (RPN). The Fast R-CNN utilizes external region proposal, this architecture has its own region proposal network, which consists of convolutional layers that generate object proposals and two fully connected layers that predict coordinates of bounding boxes. The corresponding locations and bounding boxes are then fed into objectness classification and bounding box regression layers. Finally, the objectness classification layer classifies whether a given region proposal is an object or a background region while a bounding box regression layer predicts the object.

localisation, at the end of the overall detection process. RPN’s purpose is to propose multiple objects that are identifiable within a particular image.

-***The TensorFlow object detection API*** is the framework for creating a deep learning network that solves object detection problems.

There are already pretrained models in their framework which they refer to as Model Zoo. This includes a collection of pretrained models trained on the COCO dataset, the KITTI dataset, and the Open Images Dataset. These models can be used for inference if we are interested in categories only in this dataset.

# 

# •Contribution

We used the Tensorflow Object Detection API. We tried two pre-trained models; SSD mobilenet and faster RCNN inception\_v2. We trained both models on our dataset and made the necessary modifications in the configuration files to fit our problem. The accuracies of both models were compared, and the faster RCNN inception\_v2 model gave the highest accuracy.

# 

# •Proposed Solution

we will go over all the steps needed to create our object detector from gathering the data all the way to testing our newly created object detector.

The steps needed are:

1. Gathering data
2. Labeling data
3. Generating TFRecords for training
4. Configuring training
5. Training model
6. Exporting inference graph
7. Testing object detector

# 

# •Results

We have used faster RCNN inception\_v2 model , we trained 25 hours on gpu ( 250k step ) , using batch size of 1

--Accuracy at 0.5 IOU :

average precision (Gun) = 0.973162

average precision (Knife) = 0.901536

average precision (Pliers) = 0.931210

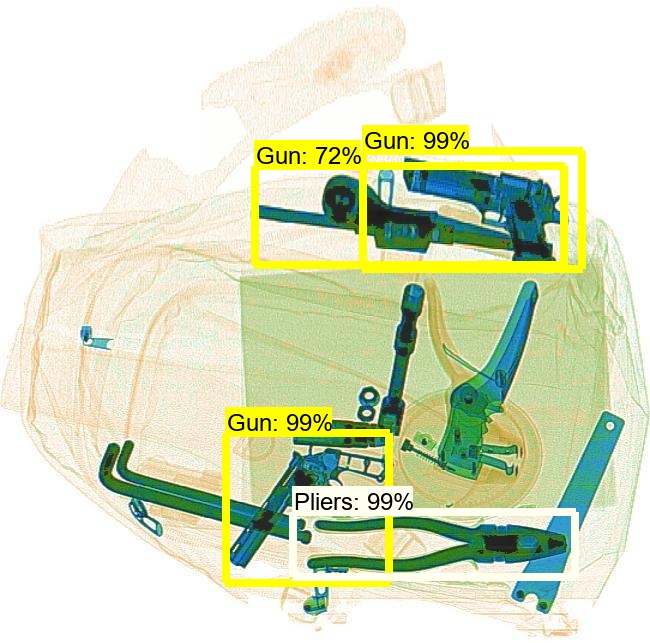
average precision (Scissors) = 0.886818

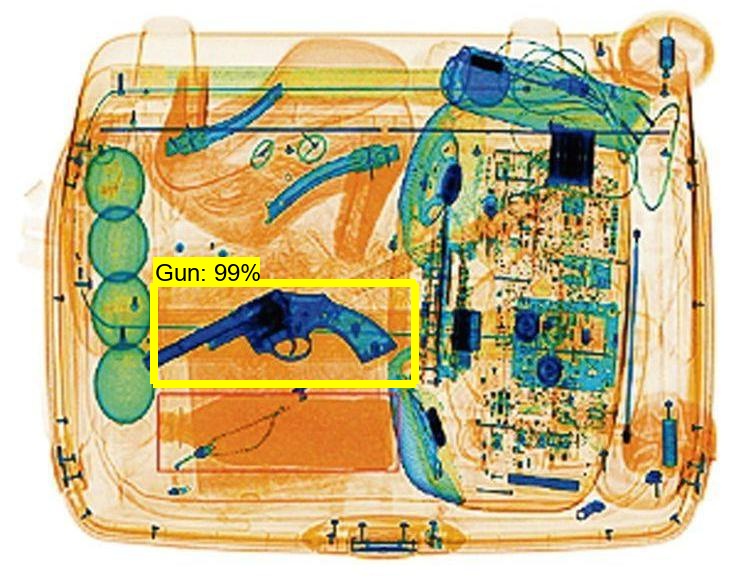
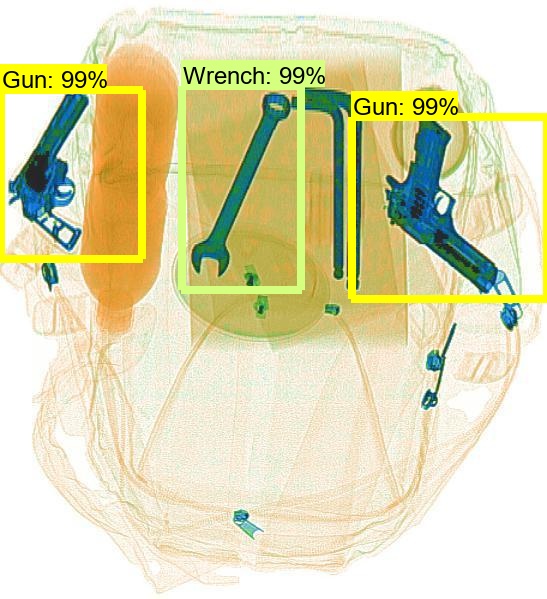
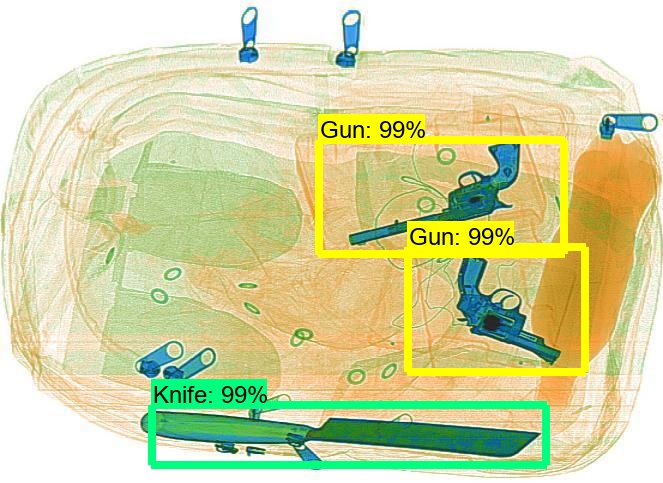
average precision (Wrench) = 0.847684

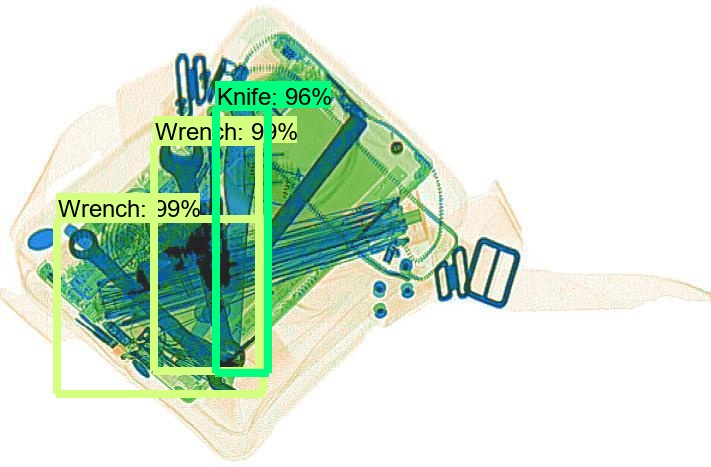
Mean average precision (mAP) = 0.907677

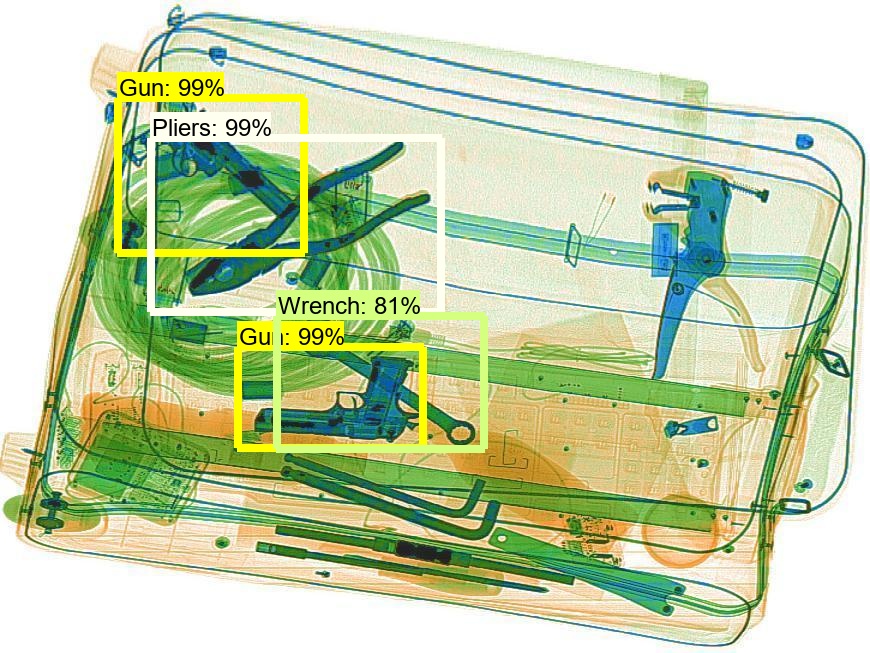
--We also trained our data using SSD , but it’s results was not good as that using faster RCNN

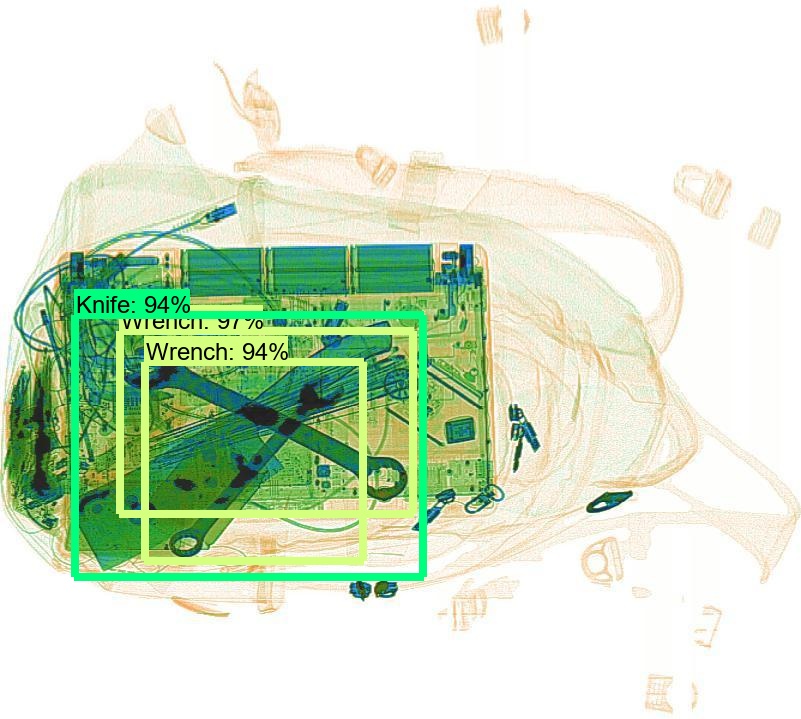
--Some of the results after training using faster RCNN inception\_v2 are on the following pictures :











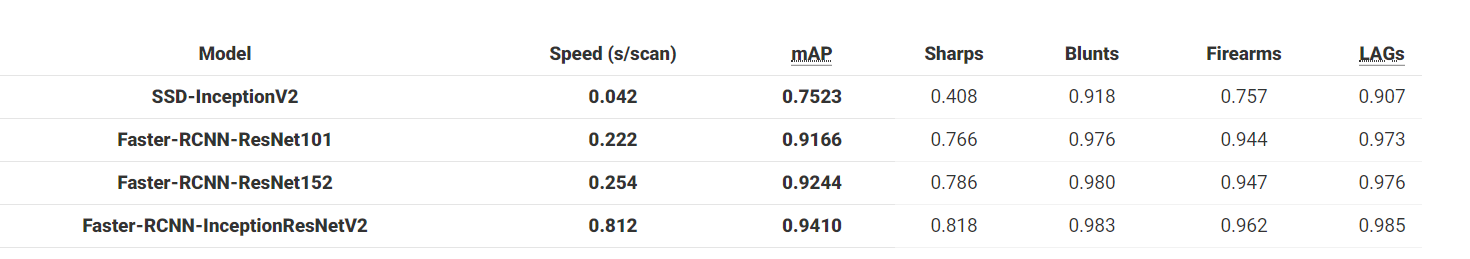
# 

# •Comparison with related work

Using faster rcnn other results were the following

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Method** | **Gun** | **Knife** | **Wrench** | **Pliers** | **Scissors** | **Mean** |
| **ResNet34** | 89.71 | 85.46 | 62.48 | 83.5 | 52.99 | 74.83 |
| **ResNet50** | 90.64 | 87.82 | 63.62 | 84.8 | 57.35 | 76.85 |
| **ResNet101** | 87.65 | 84.26 | 69.3 | 85.29 | 60.39 | 77.38 |
| **Inception-v3** | 90.05 | 93.8 | 68.11 | 84.45 | 58.66 | 77.01 |
| **DenseNet** | 87.36 | 87.71 | 64.15 | 87.63 | 59.95 | 77.36 |

some other results with a different data set



link : <https://www.groundai.com/project/toward-automatic-threat-recognition-for-airport-x-ray-baggage-screening-with-deep-convolutional-object-detection/1#id9>

# 

# •Demo

For real time forbidden items detection , see this video

<https://youtu.be/gOYRTYMdq3c>

# 

# •Conclusion and future work

Using faster RCNN with inception Resnet v2, it can produce results with better accuracy, but it’s slower..

We can modify the parameters of this model as we wish as well from the configuration file.