The Development of Real-Time Firearm Detection with CCTV in Tensorflow

Darryl Bonnici
Institute of Information & Communication Technology
Malta College or Arts, Science & Technology
Corradino Hill
Paola PLA 9032

darryl.bonnici.b43840@mcast.edu.mt

***Abstract*—This paper's purpose is to show the development of a real time fire arm model detection which can be used in surveillance cameras and security footage. This paper will also be discussing other works conducted by other researches in an attempt to develop the same detection model and therefore the focus is the ability of the software to identify and classify a fire arm, whether a pistol or a rifle is present in real time. Implementation methods are discussed for such a software together with the benefits and risk that the software can deliver.**

***Index Terms*—Mcast, gun, firearm, detection**

I. INTRODUCTION

According to [1] the study on the reality of home security and their response times, reports that a verified alarm requires human eyes on the scene before contact with local police is made. This has been done to reduce false alarms by 95%, however this creates a problem of adding important time to the response time of a crime which can be upto, if not more than 10 minutes.With this information at hand, and taking into account that robbers can be armed begs the question if these crimes can be tackled quicker in order to safeguard the inhabitants of the household. Hence, the hypothesis of this study is that using real-time object detection is it possible to accurately and quickly identify fire-arms through CCTV. Given the hypothesis mentioned above, the following research questions were identified:

Which data set is ideal to identify different firearms?

Which model of real-time object detection works the fastest in terms of time?

Which model of real-time object detection works the best in terms of accuracy?

II. LITERATURE REVIEW

Real-time object detection for fire-arms is already in place in most areas with high-security. Although not common, it is being used both by the public and private sector. Although proving useful in terms of decreasing the alert time, the sliding-window approach used is highly accurate however lacks the speed of detection due to its approach of scanning each frame for up to 4 to 15 seconds, which would lead to a delay in the detection. Most of these detection models being used are custom built and therefore are quite intensive to build. Building a faster detection model using another approach is possible but is not always as accurate and due to the high-nature of the topic being fire-arms and crime, accuracy tends to be the leading factor at the compromise of time.

Gun violence across the globe still remains high even with defense mechanisms being implemented in establishments and households, most of which make use of surveillance systems. Gun violence can range from armed robbery to murder. Most of the armed robberies occur in public places such as malls and places of business, although a significant number of these robberies also occur in private households. According to demographics by [2] it is registered that in the U.S during 2019, a total of 82,102 cases of armed robberies were reported, together with another 101,120 cases of strong-armed robberies were reported. Unfortunately, reports show that police, when alerted, reach the scene of the crime when it is too late. This is mostly because, alerts need to be manually triggered in the case of public spaces and places of business, whilst in private households when an alert is triggered, it needs to be verified before being passed to the local police.

Research conducted by [3] in Developing a real-time gun detection classifier, collected over 3000 images of firearms and resized images to 640x 480 pixel to be compatible with TensorBox framework and achieved a baseline of 58 % accuracy on the revolver classification and 46% on the rifle classification. Using GoogLeNet OverFeat-3 consistently performed the best on training with accuracy. GoogLeNet Overfeat uses is a sliding-door window approach towards detection, which analyses each frame into sub-frames. Although GoogLeNet Overfeat is recognized for its accuracy, it is not viable for real-time detection as the processing can be time consuming. This process can take anywhere from 2-15 seconds per frame, which would undoubtedly would not be efficient enough for the task ahead. Considering that most CCTV capture video at 30 frames per second, this would add up quickly especially when taking into account that to draw a weapon can take up to at least 60 frames. Testing different version of Overfeat, Lai and Maples managed to achieve an accuracy of 93% when detecting for revolvers and rifles from real-time detection of images from movie scenes which included guns. Table I shows their results on testing with different models of Overfeat.

TABLE I
COMPARING DIFFERENT MODELS OF OVERFEAT [1]

Model	Train Acc	Test Acc
VGGNet	0.57	0.46
Overfeat-1	0.62	0.56
Overfeat-2	0.69	0.64
Overfeat-3	0.93	0.89

Another research conducted by Wang, [4] on Recognizing Firearms from Images and Videos in Real-Time with Deep Learning and Computer Vision, test multiple real time object detection models and software on the same data set of firearms. After running all test, Wang compares all the data acquired from the tests conducted to identify which solution is the best. As can be seen from Table III , speed and accuracy have been given high priority, however accuracy has been giving a slightly higher opportunity. From the results collected from the findings, it is clear that the Darknet YOLO was proven to be best solution. Although this was used on a single image and trained on a data set of 50 images, Wang concludes that to be used in real-time fire arm detection, both Darknet YOLO and Mask R-CNN should be used in a two-phase recognition software, where YOLO is used for a preliminary scan and Mask R-CNN used as a secondary scan to validate the findings.

Research conducted on [5] Gun Detection in surveillance videos using Deep Neural Networks, by Lim et al, describes how comparing two data sets has an effect on the result of the guns being detected. According to their investigation, it shows that having high-resolution images in the data set will not give an accurate reading as most likely frames from the CCTV footage is at a lower resolution which result in images from frames being of lower quality, which makes the recognition software more difficult to recognize. Even though a high end graphics processing units(GPUs) were used (two NVIDIA Quadro P500 graphics processing units) for the training and testing with a combined video memory of 32 GBytes, the importance of variety of quality within the data sets is important to have a more accurate and efficient score. Table II, shows a comparison of accuracy between two data sets.

TABLE II
COMPARISON OF DETECTION ACCURACY BETWEEN DATA SETS IN TERMS OF MAP PERCENTAGE

Model	Dataset	Avg. Precision, IoU	Avg. Precision, Area
-	-	0.5:0.95 0.5 0.75	S M L
1	Granada	0.114 0.281 0.053	0 0.110 0.800
2	Ours	0.223 0.442 0.202	0.180 0.224 0.717

TABLE III
DECISION MATRIX OF SOLUTIONS

Solution Rubrics	Accuracy	Speed	Throughput	Modularity	Final Score on 100
Rubrics Weight	35 %	30%	15%	20%	100%
TensorFlow	60	55	80	90	67.5
DeepDetect	50	60	60	80	60.5
Mask R-CNN	80	75	80	90	80.5
Facebook Detectron	75	40	50	90	63.5
Darknet YOLO	70	100	95	80	84.5

TABLE IV
COMPARING DIFFERENT MODELS OF MOBILENET

Model Name	Speed (ms)	COCO mAP	Outputs
SSD MobileNet v2 320x320	19	20.2	Boxes
SSD MobileNet V1 FPN 640x640	48	29.1	Boxes
SSD MobileNet V2 FPNLite 320x320	22	22.2	Boxes
SSD MobileNet V2 FPNLite 640x640	39	28.2	Boxes

Table IV is data taken from the official Tensorflow GitHub model data, in which a comparison is made on the different models of MobileNet model, which is used in conjunction with the TensorFlow API to build the real-time object detection of custom models and data sets.

Another approach of [4] recognizing firearms from images and videos in real time with deep learning and computer vision is using Tensorflow API, which is one of the most mature open-source machine learning platforms with high efficiency in both training and processing, which allows for custom object training and detection. However, being an open-source platform may be vulnerable to updates and out-dated systems, yet the owner of Tensorflow, Google offers consistent support and documentation.

III. RESEARCH METHODOLOGY

After carefully studying previous works that have been conducted on the same subject matter. It is clear that a problem arises when implementing a real time fire arm detection software that is accurate and efficient at detecting fire arms. Finding a balance between the two is hard, and surely will be at a cost for one of those attributes. The aim of this paper is to develop a real time fire arm detection with the use of CTTV in Tensorflow, that is both accurate and efficient. The main research question is that if Real time fire arms detection can be efficiently used to improve security amongst the public and private sector. The hypothesis is that this will be a start for implementing this software into day to day security and help in the fight against armed crime. Our objective are:

1) Creating custom data sets for each scenario and focusing on the quality of the data sets to be the same of that of a CCTV camera.
2) Find a suitable model that is both accurate and efficient.
3) Build a test with real fire arm models and prove that it can be used.

To achieve the objects mentioned above, we developed these research questions below to help guide us in our development journey to create such a software and prove/disprove our hypothesis. It is of utmost importance that we answer these questions and they are key landmarks in our development stages to build a software that is both capable of detecting fire arms in real time whilst also being fast, accurate, efficient and can be widely used.

1) Does a data set need to be created for such a research?
2) Which current technology is best suited to address the problem?
3) How does the proposed solution compare with existing solutions?

To answer our research questions, we've built a research pipeline, shown in Fig 1. In this pipeline, we first gathered all data sets required such as images of two categories of guns, which for the purpose of our test, we made use of a pistol and a rifle. The images collected had to be classified individually. After collecting the images for our data sets we proceeded to load the data into our software and transform the data sets to be of the same dimensions of that of a CCTV camera. Following the transformation of the data sets, we split our data sets into training and testing for the software to be able to train on, and eventually to test on. After several of hours training the software with our data sets, we ran the test and the software was able to give a predication fire arms visible on the CCTV, which for the purpose of this test as represented with a webcam.

IV. Technologies Used

For the purpose of the development of the real time firearm detection, below is a list of the hardware specifications which was used develop, train and test the software.

1) CPU: Intel Core i5 - 4690 3.50GHz
2) RAM: 12GB
3) GPU: NVIDIA GeForce GTX 1060 - 6GB
4) Webcam: Aukey PC-LM1E

For us to build the real time firearm detection, we made use of different packages together with different programs and programming languages. All the software used throughout this development is listed below:

1) Python
2) Jupyter
3) Tensorflow
4) OpenCV
5) Keras

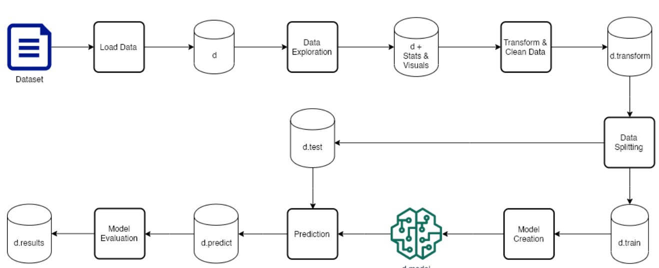

Fig. 1. Research Pipeline

Results on the confidence and speed of the real time detection was captured for the purpose of quantitative analysis. The speed of the detection is important as this shows us the the efficiency of the software, whilst on the other hand confidence is important as this gives an actual representation on how confident the software is at recognizing firearms from the trained models.

V. Findings & Discussion of Results

The data collection was fairly simple to do. We did our best to get an image of every angle of the firearms available to us to conduct the tests. This was done via sourcing images from Google Images, however we noticed that not all images covered different angles and thus we resorted to collect some other images from videos available through YouTube, of the firearms both in closed and open environment. During the duration of the development of this software, only 77 images were used for the training and testing. This factor was effected mostly by time, as sourcing of these images and labelling each individual image took hours of time, considering it takes upto 5-10 minutes for each image to be downloaded, edited and labelled.

The **choice of model** which was used for the training and testing of the data set, was the MobileNet V1 FPN 640x640. This was chosen as a result of the low-end hardware specifications, but also the speeds and mAP were taken into consideration and thus the decision was made the MobileNet V1 FPN 640x640 was right for this study as it offers both high speed and high mAP and also outputs in bounding-boxes, which makes it easier to identify. As mention before, this Table IV shows the speed of the model compared to other models.

After running the software and testing it with two real-life firearms replicas, the test was a success in answering the research questions asked before.The custom data had to be built from scratch mostly due to the fact that images found on pre-existing data sets either do not include all list of fire arms or do not take into consideration low and grainy life feeds, therefore making it hard for this specific application. However, based in the results achieved in Table V, clearly shows that this application has potential. After the comparison of the results obtained with previous findings, considering that our data sets were only made up of 77 images, and previous works collected over 3000 images while using far better hardware for other tests, the results achieved are impressive.

TABLE V
Recall Results

Model	Class Name	Avg. Time to Detect	Avg. Accuracy
Pistol	asg ics ble xae	$\leq 0.5sec$	62%
Rifle	m4 carbine	$\leq 1sec$	61%

VI. Conclusion

In conclusion, we identified some strong suits while other potential improvements. Firstly, we've identified that a custom data set can be much more efficient for testing and training, however the time taken to collect and classify all the images requires great amount of time but this is important as the custom data set will allow the application to adapt to certain environments of the building it is being used in, and therefore make it faster and more accurate. Secondly, based on previous works, we opted for the MobileNet model, which has yielded great results with minimal issues. For this scenario where time is of the essence, and hardware can be a limitation, the MobileNet model can sufficient to achieve a quick and accurate response, with a custom data set containing at least 50-100 images per weapon, this is to train every angle possible. Our objective for this study was to study the fire arm classification and if being the software is implemented, can reduce crime and or increase police efficiency. The objective became skewed along the way, as during our research process, it became clear that there are many variables to having a reliable real time fire arm detection, and due to the subject matter, false alarms can end in deadly scenarios. Therefore, the objective post-research was on the focus of having this software being used as a tool for security guards that are monitoring CCTV, to alert them of a potential firearm, however visual confirmation is still required to avoid accessory false alarms.

As mentioned before, limitations to making this application work better, were present. The one and only limitation which discouraged greater results, as time. Time to find more images for the custom data sets and time for training testing. Therefore, it is recommend anyone taking on this task, to build a custom data set consisting of 100-200 images per class. Having better hardware specifications, will improve testing training time drastically.

Appendix A
Supporting material

Refer to Figures 2 and 3 , for screenshots taken from the tests.

Fig. 2. Testing Pistol

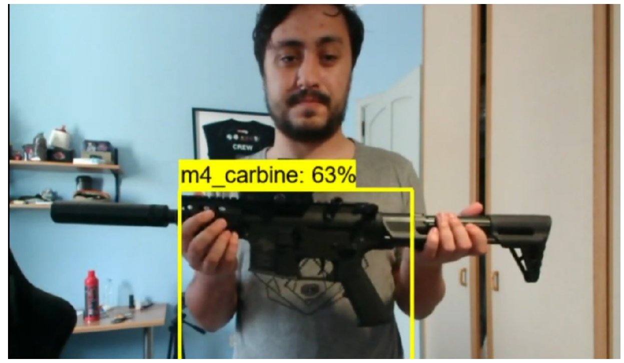

Fig. 3. Testing Rifle

Appendix B
Disclaimer

As a disclaimer, fire arms used in the testing for real time fire arms detection are actually Airsoft 1:1 replicas of real firearms and do not shoot real bullets. Images in data sets are a mixture of real and replica fire arms.

References

[1] T. Borys, "The harsh reality of home security systems their response times," Available at https://www.deepsentinel.com/blogs/news/the-harsh-reality-of-home-security-systems-their-response-times/ (2021/01/15).

[2] Statista.com, "Number of robberies in the united states in 2019, by weapon used," Available at https://www.statista.com/statistics/251914/number-of-robberies-in-the-us-by-weapon/ (2020/09/01).

[3] J. Lai and S. Maples, "Developing a real-time gun detection classifier," Available at http://cs231n.stanford.edu/reports/2017/pdfs/716.pdf.

[4] T. Wang, "Recognizing firearms from images and videos in real-time with deep learning and computer vision," Available at https://medium.com/@tont/recognizing-firearms-from-images-and-videos-in-real-time-with-deep-learning-and-con

[5] V. M. B. J. M. L. K. W. J. S. JunYi Lim, Md Istiaque Al Jobayer, "Gun detection in surveillance videos using deep neural networks," Available at http://www.apsipa.org/proceedings/2019/pdfs/379.pdf.

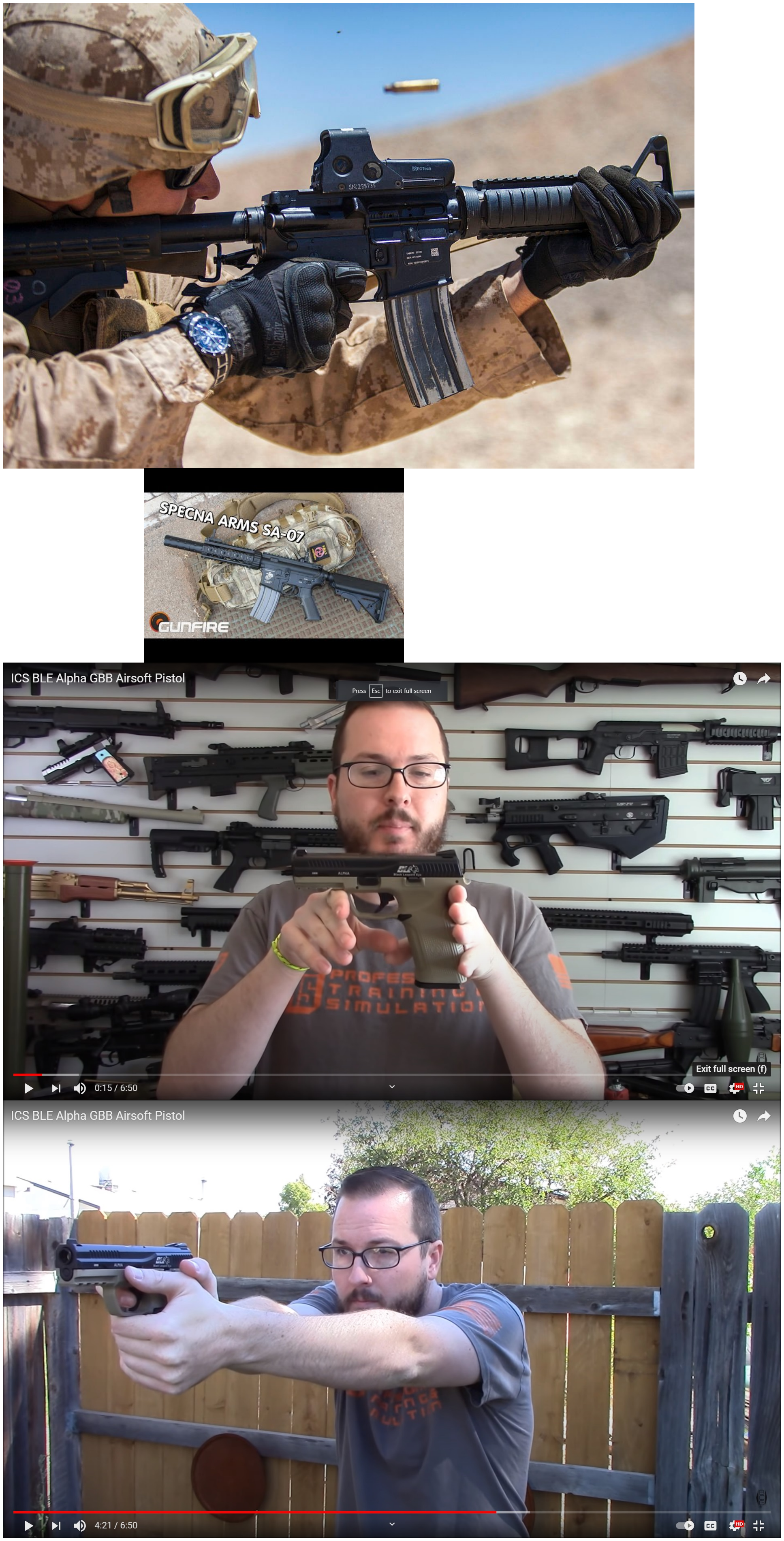

Fig. 4. Some Images used in the custom data-sets