Firearm Detection using Convolutional Neural Networks

Rodrigo Fumihiro de Azevedo Kanehisa and Areolino de Almeida Neto
Federal University of Maranhao (UFMA), São Luís, Brazil
rodrigokanehisa@gmail.com, areolino@ufma.br

Keywords: Firearm Detection, Computer Vision, Darknet YOLO.

Abstract: This papers studies the application of the YOLO algorithm to create a firearm detection system, demonstrating its effectiveness in this task. We also constructed a dataset based on the website Internet Movie Firearm Database (IMFDB) for this study. Individuals carrying firearms in public places are a strong indicator of dangerous situations. Studies show that a rapid response from law enforcement agents is the main factor in reducing the number of victims. However, a large number of cameras to be monitored leads to an overload of CCTV operators, generating fatigue and stress, consequently, loss of efficiency in surveillance. Convolutional neural networks have been shown to be efficient in the detection and identification of objects in images, having sometimes produced more accurate and consistent results than human candidates.

1 INTRODUCTION

An Individual carrying firearms in public settings is a strong indicator of possible dangerous situations. Recently there has been an increase in the number of incidents in which individuals or small groups make use of firearms in order to injure or kill as many people as possible. Among the most notable of these events, called mass shootings, are the Columbine massacre (USA, 37 victims), the attack on Uotya Island (Norway, 179 victims), the Realengo massacre (Brazil, 13 victims) and that one against the Charlie Hebdo newspaper (France, 23 victims).

Surveillance systems such as closed-circuit television (CCTV) and drones are becoming increasingly common. Research shows that the installation of CCTV systems helps to combat mass shooting incidents (Kayastha, 2016). Scotland Yard used CCTV images as evidence in 90% of homicide cases in 2009 (Barrett, 2013). In England, it is estimated that there are around 5 million security cameras nationwide (Barrett, 2013). Despite helping to combat crime, the large number of cameras leads to a huge overhead for its operators.

Automatic surveillance systems have begun to emerge in recent years, mainly for the use in intelligent transportation systems (ITS). These include traffic surveillance (Bramberger et al., 2003) and vehicle recognition (Baran et al., 2015). Another application of cameras in surveillance can be on security, then a system capable of automatically detecting firearms in images would enable a faster and more efficient response from law enforcement agents. One of the most promising techniques for the creation of automatic surveillance systems is machine learning and computer vision.

This paper is divided into five sections. In addition to this introductory section, it is presented the theoretical basis necessary for understanding this work in the following section. In the third section, the methodology and the tools used are discussed. In the fourth one, the results obtained during this study are presented. And finally, in the fifth section, conclusions of this research are discussed.

2 RELATED WORK

(Verma and Dhillon, 2017) propose the use of convolutional neural networks (CNN) for detection of firearms. The training was made with the use of transfer learning. The tests were performed with a dataset built from the Internet Movie Firearm Database (IMFDb). The work could detect and classify three types of guns, pistols, revolvers and shotguns.

In (Bertozzi, 2017), a method was proposed for detection of firearms and potentially dangerous situations using CNN, in addition to proposing the use of proprietary hardware to capture the images. This work was aimed to detect and react to the presence of firearms or other weapons when used in a threatening way.

In (Ardizzone et al., 2014), bottom-up saliency map and top-down saliency map techniques were used to create probabilistic models for the position of firearms in position-based images of the face of a person carrying it, also with the use of an extracted dataset from the site IMFDb.

In (Grega et al., 2016), a method was proposed for automatic detection of dangerous situations in CCTV systems, through the use of image processing and machine learning. Sliding window techniques, fuzzy classifiers and canny detectors were used for detection of knives and firearms in a video. In addition to the detection system, the authors constructed and made available their dataset.

3 BACKGROUND

3.1 Darknet - YOLO

Darknet YOLO is a state-of-the-art object-detection system based on convolutional neural networks (Redmon and Farhadi, 2016). It was initially developed using Darknet, an open-source neural network framework written in C and CUDA (Redmon, 2016). Traditional classifiers use sliding window techniques or selective search to find candidate regions and identify the desired object. In this way, regions with high probabilities are considered detections (Redmon and Farhadi, 2016).

Unlike other methods, YOLO does not repurposes a classifier for detection. This algorithm looks at the image only once. To perform the detection, the image is subdivided into multiple subregions. For each sub-region, five bounding boxes are considered and the probabilities of each of them contain an object are calculated. By looking only once to the image, YOLO executes a much faster detection, being a hundred times faster than Fast R-CNN (Redmon and Farhadi, 2016). Figure 1 demonstrates the detection process.

Despite its efficiency, YOLO has some limitations. YOLO imposes strong spatial constraints by using bounding box predictions since each sub-region provides a limited number of bounding boxes, each of which can only have one class. This spatial limitation limits the number of near objects this model can detect. It also has difficulties with small objects appearing in groups, such as flocks of birds (Redmon et al., 2016).

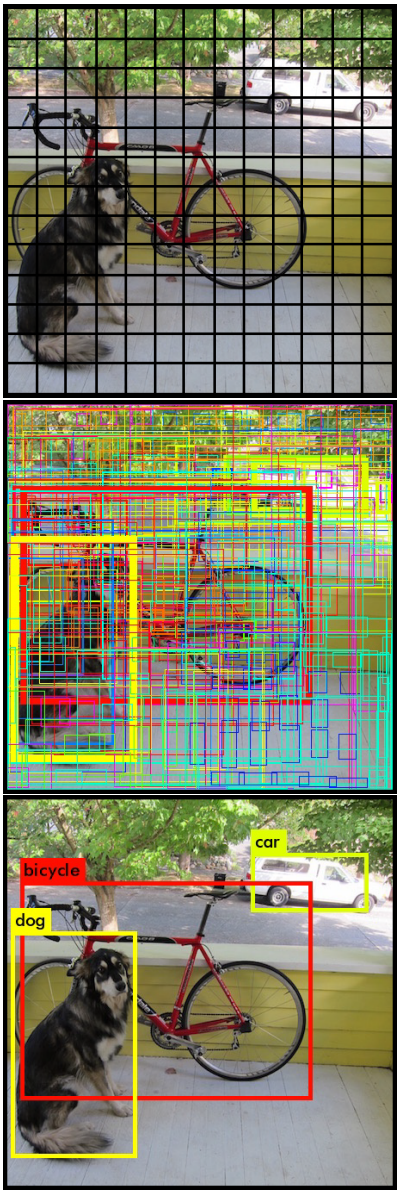

Figure 1: Operation of Darknet YOLO.

4 METHODS

4.1 Dataset

IMFDb is an on-line database of firearms images used or displayed on films, television shows, video games and animes. There are included actor related articles and some famous characters, such as James Bond, listing specific weapons associated with his works (Bourjaily, 2009).

The site also includes an image hosting section similar to Wikimedia Commons, which contains

Figure 2: Example of an IMFDb article.

firearm photos, manufacturer logos, screenshots and related art (Bourjaily, 2009). From this session, the dataset used for positive cases was extracted. About 20,000 images were extracted, of which approximately 4600 were used since the dataset annotation process was manual.

4.2 Scrapy

To construct the dataset, a web crawler was developed using the Scrapy library, which is an open source Python library used for scraping sites, extraction of structured data from web pages and web crawling. It can be used for a large number of applications such as data mining, archiving or information processing. It is currently maintained by Scrapinghub Ltd., a web-scraping development and services company (Korobov, 2015).

The architecture of the scrapy system is based on the construction of spiders, which are entities responsible for navigation within the provided URLs. In addition to browsing and accessing links, a spider also processes the information it finds filtering and downloading it. A spider was constructed for this paper that searches for images and downloads them inside the IMFDb, looking for pages of specific models of firearms.

The criterion used for the search was the popularity of firearms models. The web-crawler looked for guns such as AK-47 or M-16 assault rifles (UN. SECRETARY-GENERAL et al., 1997), Colt M1911, Glock and Sig pistols, shotguns such as Remington M870, Mossberg 500 (Brauer, 2013). Often, works containing these models also contained other models not specified during searches. In this way, the dataset contains images with a wide variety of models.

4.3 Google Images Download

For the creation of the dataset of negative images, the Python script Google Images Download was used. This tool is a command line program to search for keywords or key phrases in Google Images and optionally download them (Vasa, 2018).

Figure 3: Examples of negative images.

The images sought were allusions to firearms, both in shape and texture, such as hair dryers and handheld power drills. Items incorrectly classified during preliminary tests were also searched such as umbrellas or metal handrails. The dataset of negative cases acquired contained 1939 images. In this way, a negative dataset was obtained to designate objects that should not be classified as firearms.

4.4 Dataset Annotation

After the acquisition of the dataset, it was prepared using the tool BBox-Label-Tool (Shi Qiu, 2016) to

mark the ROI (Regions of Interest) in the images. This process was performed only for the set of positive images since in the set of negative images there is no ROI.

Only images containing small arms, by the United Nations definition in the SALW protocol (Small Arms and Light Weapons), were considered. This category includes handguns (pistols and revolvers) and long guns (submachine guns, carbines, rifles, shotguns and light machine guns) (UN. SECRETARY-GENERAL et al., 1997) (Small Arms Survey et al., 2007). Grenade launchers, rocket launchers, vehicle-mounted weapons and explosives were ignored.

Figure 4: Dataset annotation tool.

For each image, a text file with the same name was generated. This file contained the number of labels in the image and the coordinates of the bounding boxes on each of these. In some cases, there are no labels for an image because they did not meet a minimum quality criterion, having a poor resolution or bad lighting. These text files were saved with zero labels and were then the images eliminated from the positive dataset.

From the 20 thousand images obtained, only 4646 were marked. The process of marking ROI on the images was executed manually. Each image was analyzed individually and the regions containing firearms were marked. For this reason, although 20 thousand images were downloaded, only 4646 were used.

4.5 Network Training

The dataset was randomly divided into two groups, training and testing. About 90% (5916 images) of the images were used for training and 10% (669 images) for testing. The learning rate used was 0.0001, the momentum of 0.9 and the decay of 0.0005. The batch size was 64 images and the input layer size was 416×416. Finally, anchors were generated for the training process based on the positive dataset, using the k-means clustering algorithm.

The network architecture used in this paper was based on the architecture used by (Redmon and Farhadi, 2016) in the PASCAL VOC challenge (Everingham et al., 2015). This architecture was chosen because it proved to be efficient for the classification and detection of objects during the mentioned challenge.

During the training process, each epoch took an average of 5.5 seconds. Up to the first 1000 epochs, a checkpoint of the network was saved for every 100 epochs, after the first 1000 epochs, this interval was increased to 1000 epochs. The graph in Figure 5 shows the network loss function during training.

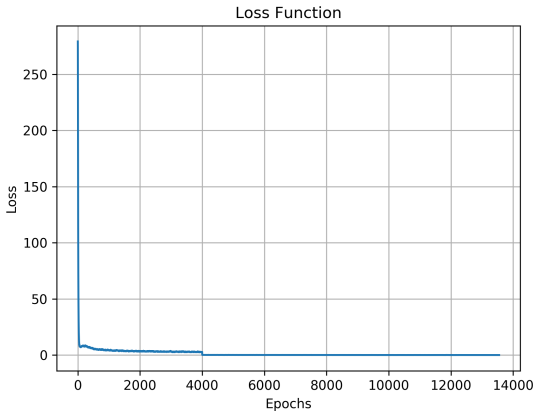

Figure 5: Loss function of the training process after 13 thousand iterations.

5 TESTS AND EVALUATIONS

In this session, the results of this work are presented. These tests were performed to analyze the performance of the methodology. The system was evaluated in two main criteria: classification and detection. The classification problem consists in classify images by the presence of firearms. In the detection problem, the system tries to identify the location of guns, in addition to their presence in the image. The dataset used for testing contains around 10% of the images, there were a total of 669 images, 446 were positive and 223 were negative, containing a total of 554 marked objects.

5.1 Metrics

The classification problem was evaluated in a traditional way. For each test image, the system aimed to

identify the presence or absence of a firearm. Sensitivity, specificity, accuracy and ROC curve (Receiver Operating Characteristic) were used to evaluate the classification (Fawcett, 2006).

For the detection problem, the PASCAL VOC challenge metrics were used (Everingham et al., 2008). These metrics were chosen to generate an objective comparison criterion of the technique used in this work with future techniques.

The intersection over union (IOU), also known as the Jaccard index, is the computation of the bounding box bounding box intersection of ground truth. This metric is widely used in object detection challenges such as PASCAL VOC (Everingham et al., 2010) and for evaluation of algorithms like HOG + SVM and CNN's (Rosebrock, 2016).

In the PASCAL VOC challenge, an IOU value of 50% is used, with any detection with confidence above this value being considered a match, i.e., a positive classification (Everingham et al., 2015). Values with IOU less than 50% are considered errors, even if they contain the object to be detected. The detected bounding box must have the same label of the ground truth and the region must be detected only once to be considered a true positive (Everingham et al., 2010).

5.2 Results

In this session, the results obtained in each of the experiments are presented. These tests were performed to analyze the performance of the methodology used. The dataset used for testing consist of a total of 669 images, 446 were positive and 223 were negative, containing a total of 554 marked objects.

Initially, tests were performed with the training done using only images with positive cases. This network was trained for six thousand epochs. These tests were done to evaluate the detection efficiency of the tool and to check which objects were detected incorrectly. After that, the dataset of negative cases was constructed, based on the results obtained in the preliminary tests and subsequent training was performed. This training was done for 13 thousand epochs.

The application proved to be efficient in detecting small arms of all types and models. Demonstrating the ability to locate objects even when a partial occlusion occurs. In addition to being able to detect all types of small arms, the system was able to detect weapons such as anti-aircraft artillery and vehicle mounted guns, as well as rocket launchers and grenade launchers.

The application is also able to distinguish 'L' shaped objects, such as drills and hair dryers, commonly compared to firearms. This is true even when they are welded so as to refer to the carrying of weapons.

The results obtained during the image classification tests were promising. From 669 test images, 644 were correctly classified: 427 were true positive and 217 were a true negative. From those incorrectly classified, there were six false positives and 19 false negatives, these results can be seen in the table 1. The values obtained were 95.73% of sensitivity, 97.30% of specificity and 96.26% of accuracy. The ROC curve drawn from these results can be seen in Figure 6.

Table 1: Confusion Matrix.

Predicted /Real	P	N
P'	427	6
N'	19	217

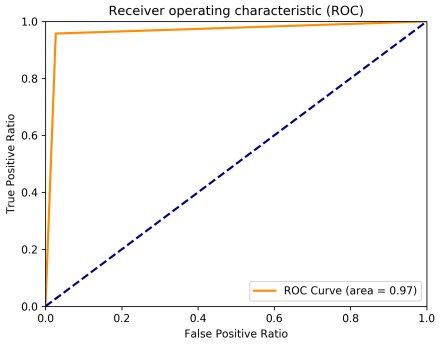

Figure 6: ROC curve.

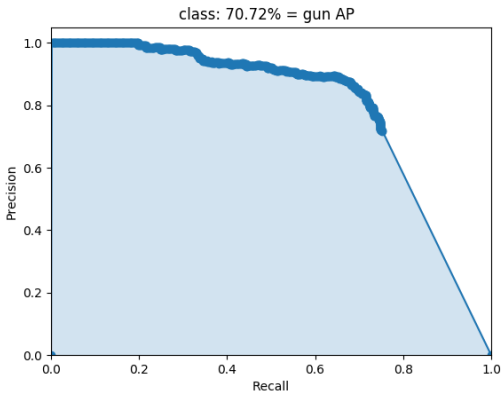

Figure 7: Precision and Recall.

The Mean Average Precision (MAP) value was 70.72%, comparable to state-of-the-art applications for detecting objects for a single class, this result can be seen in Figure 7. It can be emphasized that in some cases, the error occurred due to differences of IOU between the bounding box and the ground truth, but as can be seen in the images below, the system is capable of performing useful detections even in this situation. Other cases considered

errors by this methodology were cases of detection of previously unmarked weapons in the dataset occurred, such as rocket launchers and vehicle mounted weapons. These results, although correct, reduce the MAP within the metric used.

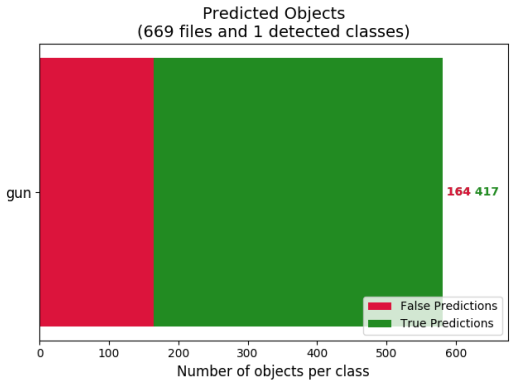

Figure 8: Object detection result chart.

In the images below, some of these cases are demonstrated. The regions predicted by the system are indicated by the red box and the region of ground truth the blue box. In the Figure 9 there is an example of a valid detection, however, with IOU less than 50%.

The Figure 10 demonstrates cases of objects not marked, by not being classified as small arms, which were detected by the system. These images consider only individual detections, so only one detected regions are highlighted in each image even when there are other visible firearms such as Figure 10.

5.3 Firearm Detection Demonstrations

The following images demonstrate the results achieved with this system. The Figure 11 contains a wall with multiple replicas of firearms. From the 17 objects present, 9 were correctly detected. Failures occur due to limitations of this technique with clusters of objects in close proximity.

In Figure 12, there are two hair dryers alluding to firearms and a weapon under partial occlusion. This test demonstrates the system's capacity to differentiate guns from objects with a similar shape. In addition to showing the technique's robustness to partial occlusion.

In Figure 13, an almost totally occluded revolver is detected. Only part from the barrel, grip and trigger guard are visible. This image shows the system's ability to identify objects almost completely occluded.

The following images demonstrate cases of error. In Figure 14, it can be seen that both guns were detected correctly, however, part of the motorcycle's

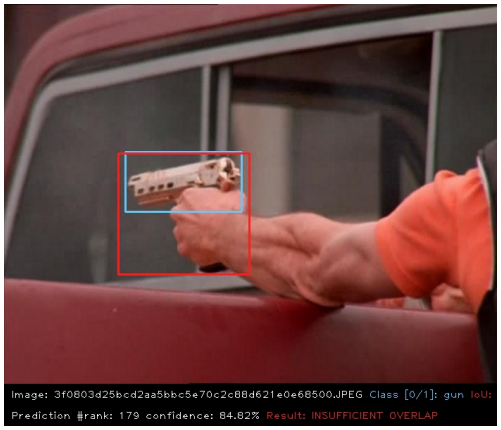

Figure 9: Demonstration of a valid detection, but with a low MAP value.

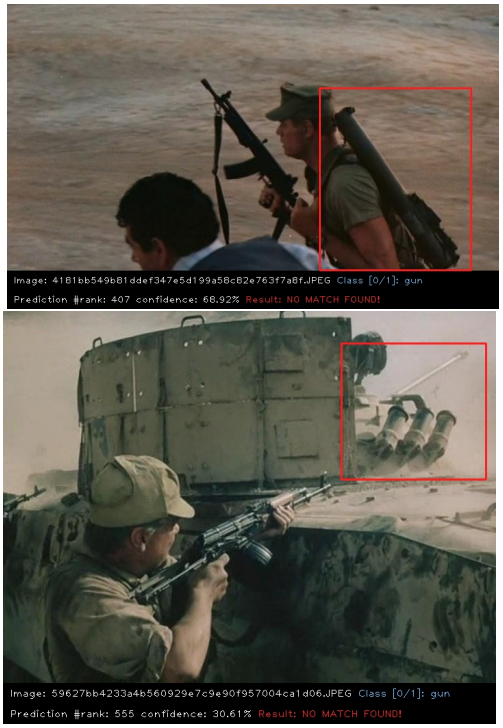

Figure 10: Demonstrations of detection of correct and valid but unmarked objects.

handlebar and suspension were incorrectly classified as a firearm. This is a common error case when there are metal or L-shaped objects close to hands.

Detection failures occurred in situations where there are metallic or frosted objects, usually close proximity or overlapping with hands, especially if these objects are elongated. The faults also occurred with 'L' shaped, metallic, polished or dark objects, usually in contrast with a light background. Among the objects generating false positives are metal handrails when near or covered by hands, poles

Figure 11: Demonstration of positive results in an environment with multiple objects.

Figure 12: Demonstration of positive results in an environment with partial occlusion and with alluding objects.

Figure 13: Demonstration of positive results in a environment with partial occlusion.

and metal structures containing protrusions.

The Figure 15 shows the rifle correctly detected, however, the side railing was also highlighted. Again detection occurred with an elongated metal object.

Although some cases of error have occurred, these are minimal and do not make the system unfeasible, given the sensitivity, specificity and accuracy reached respectively 95.73%, 97.30%, 96.26%, evidencing the system's ability to identify firearm-containing images with great efficiency, as well a MAP rate of 70%, shows the system's efficiency in locating objects.

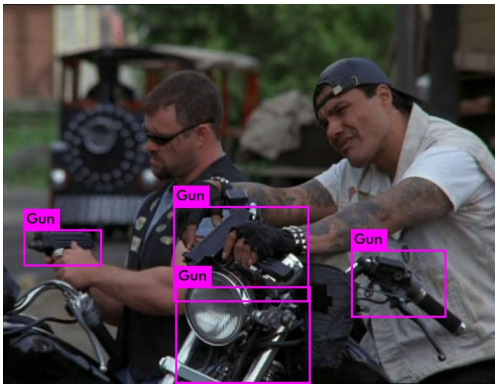

Figure 14: Demonstration of an error case.

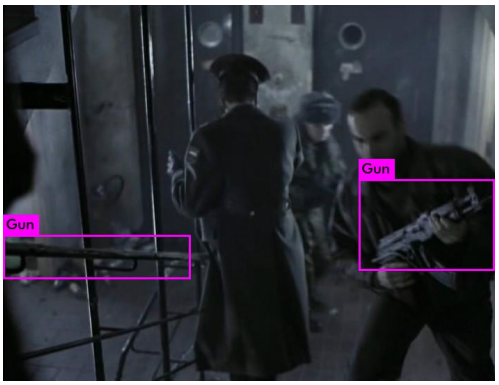

Figure 15: Demonstration of an error case.

6 CONCLUSIONS

This work demonstrated the feasibility of using CNN's to create a generic firearm detector. The method used proved to be robust and able to correctly detect weapons that were not presented during the training (different models and in many environments). The dataset constructed for this work proved sufficient variation to allow the system to be able to understand the concept of a firearm.

The results obtained, 95.73% of sensitivity, 97.30% of specificity, 96.26% of accuracy and 70% of MAP, demonstrate the technique's efficiency. The technique used stands out for its speed of detection, capable of being executed in real time. Another contribution is the first application of this technique for detecting firearms.

Despite the system's limitations, when studying the cases of errors, one can realize that these are the result of a relatively homogeneous dataset in terms of image quality and acquisition method. This situation can be overcome by the inclusion of lower quality images or acquired with equipment for use in a real application.

Based on achieved results with this technique, it is possible to implement this system, for video input streams generated by security cameras, allowing use in real-time environments. There are also plans to use the detector in its current state for automated annotation of the remaining dataset, as well as including images from other sources such as security cameras and other capture methods such as infrared cameras, night vision and x-ray. The dataset can be found at the link in the footnote [1].

ACKNOWLEDGEMENT

The authors acknowledge FAPEMA, CAPES and CNPq for financial support in the development of this work. Special thanks to UFMA and MecaNET for technical support.

REFERENCES

Ardizzone, E., Gallea, R., La Cascia, M., and Mazzola, G. (2014). Combining top-down and bottom-up visual saliency for firearms localization. In *Signal Processing and Multimedia Applications (SIGMAP), 2014 International Conference on*, pages 25–32. IEEE.

Baran, R., Glowacz, A., and Matiolanski, A. (2015). The efficient real-and non-real-time make and model recognition of cars. *Multimedia Tools and Applications*, 74(12):4269–4288.

Barrett, D. (2013). One surveillance camera for every 11 people in britain, says cctv survey. *The Telegraph*, 10.

Bertozzi, N. (2017). *Advisors: Susan Jarvis*. PhD thesis, Worcester Polytechnic Institute.

Bourjaily, P. (2009). Bourjaily: The internet movie firearms database | field & stream. Retrieved in June, 04, 2018 from https://www.fieldandstream.com/blogs/gun-nut/2009/04/bourjaily-internet-movie-firearms-database.

Bramberger, M., Pflugfelder, R. P., Maier, A., Rinner, B., Strobl, B., and Schwabach, H. (2003). A smart camera for traffic surveillance. In *Proceedings of the first workshop on Intelligent Solutions in Embedded Systems*, pages 153–164.

Brauer, J. (2013). The us firearms industry.

Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2008). The pascal visual object classes challenge 2007 (voc 2007) results (2007).

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874.

Grega, M., Matiolański, A., Guzik, P., and Leszczuk, M. (2016). Automated detection of firearms and knives in a cctv image. *Sensors*, 16(1):47.

Kayastha, R. (2016). Preventing mass shooting through cooperation of mental health services, campus security, and institutional technology.

Korobov, M. (2015). Scrapy, a fast high-level web crawling & scraping framework for python. . Retrieved in June, 26 ,2017 from https://github.com/scrapy/scrapy.

Redmon, J. (2013–2016). Darknet: Open source neural networks in c. `http://pjreddie.com/darknet/`.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Redmon, J. and Farhadi, A. (2016). Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*.

Rosebrock, A. (2016). Intersection over union (iou) for object detection - pyimagesearch. Retrieved in May, 09, 2018 from https://bit.ly/2QWgBoa.

Shi Qiu, P. (2016). Bbox-label-tool. Retrieved in June, 26, 2017 from https://github.com/puzzledqs/BBox-Label-Tool.

Small Arms Survey, G., of International Studies (Geneva, G. I., and Switzerland) (2007). *Small arms survey 2007: guns and the city*. Cambridge University Press.

UN. SECRETARY-GENERAL, UN. Group of Experts on the Problem of Ammunition and Explosives, Chairperson, of Experts on the Problem of Ammunition, U. G., and Explosives (1997). General and complete disarmament: Small arms. Retrieved in May, 14, 2018 from https://bit.ly/2DEkhrU.

Vasa, H. (2018). Google images download. Retrieved in April, 21, 2018 from https://github.com/hardikvasa/google-images-download.

Verma, G. K. and Dhillon, A. (2017). A handheld gun detection using faster r-cnn deep learning. In *Proceedings of the 7th International Conference on Computer and Communication Technology*, pages 84–88. ACM.

[1] https://github.com/Rkanehisa/Firearm_Detection