

# WEAPON DETECTION AND CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORK

<sup>1</sup>Mohammed Abid Chaudhary, <sup>2</sup>Sufiyan Sayed, <sup>3</sup>Shaikh Mohd Danish

<sup>1</sup>UG Student, Electronics and Telecommunications Engineering, A.I.K.T.C, India;

<sup>2</sup>UG Student, Electronics and Telecommunications Engineering, A.I.K.T.C, India;

<sup>3</sup>UG Student, Electronics and Telecommunications Engineering, A.I.K.T.C, India

Email: {<sup>1</sup>abid.ch1234@gmail.com, <sup>2</sup>sufiyansayed41@gmail.com, <sup>3</sup>skmohddanish@gmail.com}

## Abstract

Limiting gun violence has become a major concern in recent years. Deep Learning models can help reduce gun violence by automatically detecting weapons from security cameras. In this project, we will be implementing the use of Deep learning algorithms, to detect any firearms / weapons to improve response time and reduce potential harm. The proposed system in this project is a weapon detection system based on CNN. The project will involve the use of various resources for implementation, such as: Tensorflow, OpenCV, Python, CNN, Google Colab, Numpy, Pandas

Key Words - TensorFlow, Weapon Detection, OpenCV, Convolutional Neural Network, Image Classification, Machine Learning

## 1. Introduction

Gun violence remains persistently high in a lot of places. Many gun-related crimes, such as armed assault and robbery, take place in public settings where surveillance systems are already in place. Most systems, rely on frequent human supervision, which typically necessitates an unreasonably high level of attentiveness which can be expensive and inefficient when numerous video feeds are available. The proliferation of body cameras for police officers, as well as the increase in surveillance cameras in public locations, could be used to develop weapons detection systems.

When a gun is identified in real time, video systems could warn police and surveillance personnel, allowing for faster action. The challenge of real-time detection is addressed by our method. We initially set out to determine the most efficient statistic for our project in order to lead our endeavor. On the Image dataset of weapons, we will employ the Python-OpenCV module and a Machine learning-based classification algorithm for our model.

This paper is further organized into the following 4 sections: the first section gives details of literature survey of recent research, second section gives the details of proposed work carried out and implementation details, third section displays the results of the project, last section provides a conclusion for this paper.

## 2. Literature Review

### 2.1 Automatic Gun Detection from Images Using Faster R-CNN

Faster R-CNN is used to detect guns in images. A set of images of guns taken in different situations with different backgrounds are used to train Faster R-CNN. Then the model is evaluated using another set of images.

Faster R-CNN has been widely used for Object

Detection with satisfying results. Different CNN architectures such as ResNet50, Inception-ResNetV2, VGG16 and MobileNetV2, have been used separately as feature extractors in faster R-CNN in order to select the most efficient one. MobileNetV2 has been chosen due to its suitability for mobile applications.

To evaluate the efficiency of Faster R-CNN in detecting Weapons, YOLOv2 with ResNet as a feature extractor is also used for detecting guns. Dataset has been collected from IMFDB (Internet Movies Firearm Database). The dataset contains 7,909 images with different sizes where 4,656 are labeled and 3,263 are unlabeled. To label the images they used the MATLAB ImageLabeler tool.

They have tried different models such as CNN architectures, ResNet50, Inception ResNetV2 and VGG16 and Finally MobileNetV2 are used as feature extractor in faster R-CNN. In addition, YOLOv2, an Object Detection approach based on Convolutional Neural Network known for its high speed is used also to compare its results with Faster R-CNN results. Two cloud service platforms such as Microsoft Azure and AWS have been used during the training of the model.[1]

### 2.2 Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications

This research paper focuses on automatic weapon detection utilizing SSD and Faster RCNN techniques based on CNN. Two datasets were used in the implementation. One included pre-labelled images, and other had images that were manually labelled. Fatkun Batch Image Downloader was used to acquire the datasets in bulk. For weapon detection, SSD and Faster RCNN algorithms are simulated on a pre-labelled and self-created image dataset.

Both algorithms are efficient and produce decent results, but their use in real time requires a compromise between speed and accuracy. In terms of performance, the SSD algorithm is faster, at 0.736 s/frame. Meanwhile, faster

RCNN provides a speed of 1.606 s/frame, which is slow when compared to SSD. Faster RCNN has a higher accuracy of 84.6 percent. SSD, on the other hand, has a lower accuracy of 73.8 percent. Because of its higher speed, SSD provided real-time detection, but faster RCNN provided superior accuracy.

Some resources used for implementations are: OpenCV 3.4, Python, COCO Dataset, Anaconda and TensorFlow.[2]

### 3. Methodology

The following flow chart highlights the proposed model used in this project:

Tensorflow Object Detection API

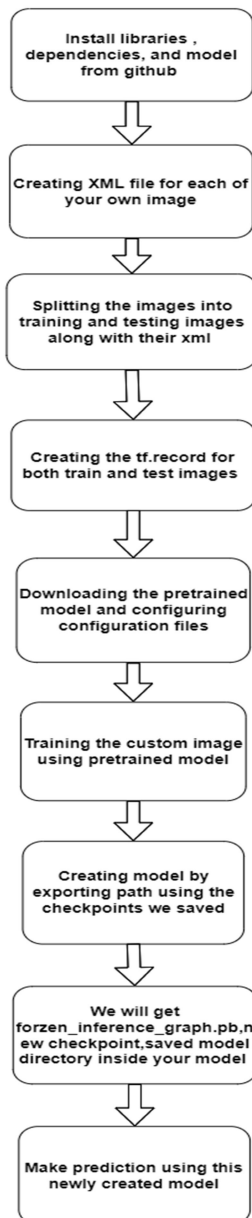


Fig. 1 flow chart of the methodology used

### 3.1 Data Acquisition

#### 3.1.1. Data collection and cleaning

The relevant photos, as well as a pre - trained model, were gathered from the internet using Python web scraping. They were also manually vetted to remove any photographs that were unnecessary or unhelpful.

Using a simple python program, all of the photos in the data - set were renamed in a format appropriate for the procedure.

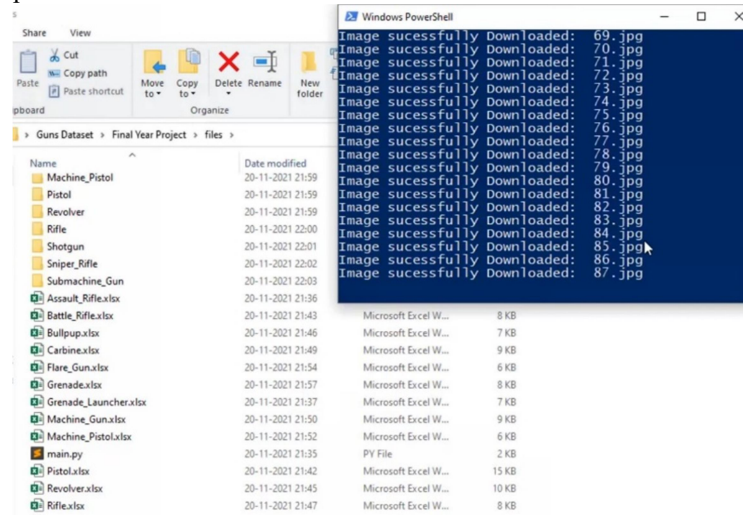


Fig. 2 Image collection via web scraping

#### 3.1.2. Data labelling

The images in the dataset were then manually labelled with their respective weapon types using the Labellmg GitHub repository, while XML files for each image were also created.

After that, the dataset is divided into training and test data in 4:1 ratio.

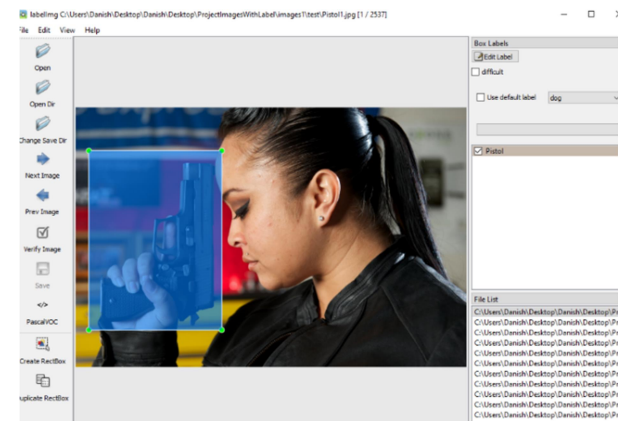


Fig. 3 Labelling pistol using Labellmg

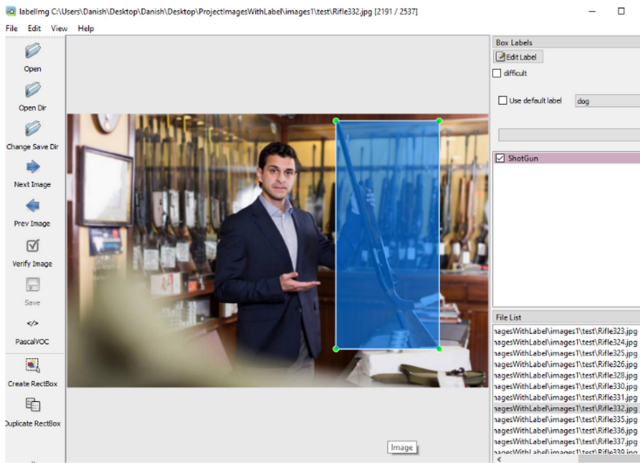


Fig. 4 Labelling shotgun using Labellingmg

### 3.2 Data Augmentation

The images in the data - set were augmented with python programs to enhance the model's accuracy and expand the range of positions in which it can detect weapons.



Fig. 5 Data Augmentation

Data Augmentation, as shown in Fig. 5, can be used to add some much-needed variety to the data - set, allowing the model to detect objects in a variety of positions and conditions. Thus, improving the model's accuracy.

### 3.3 Training the Model

The initial phase of training entails selecting an optimal pre-trained model to aid our training from the plethora of pre-trained models available on sites like GitHub, etc. We chose MySSD mobnet V1 50 320x320, a pre-trained model that includes a Python TensorFlow object detection file called model main tf2.py that allows us to train our model.

Before we can start training, we need to configure pipeline.config with important parameters like the number of steps, batch size, and paths to checkpoints, test, and training data.

The checkpoints allow us to restart the training from set points in the event of a failure, allowing us to complete the training in less time.

The batch size is usually chosen in powers of 2 (i.e.:

2,4,8,32 etc). For large amounts of data, we recommend using a batch size of 32 because it proved to be the most accurate in our tests for datasets with a large number of images.

The number of steps is up to the user's preference; increasing the number of steps offers more accurate results up to a specific threshold, but increases the training time.

We can start training the model after we've finished modifying the configurations listed before, to better fit the scope of this project.

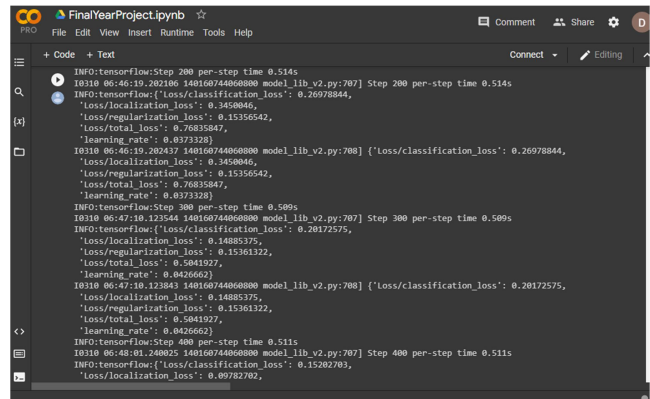


Fig. 6 Training in progress on Google Colab

## 4. Results

This research examines how data was collected from various sources, as well as how it was filtered and labelled. Web scraping was used to collect a large amount of information from a variety of sources. We then used TensorFlow to develop and train a weapon identification model using the data we collected, allowing us to detect weapons of various types

We were able to achieve the desired results after carefully optimizing the training parameters to better suit our project, with our model being able to detect firearms in real time.

The overarching purpose of the model was to provide another security and surveillance tool that might potentially replace or eliminate human involvement.



Fig. 7 Real-time Weapon detection

## 5. Conclusion

We propose a system that provides a machine with a visual sense to identify guns and the capacity to alert security personnel when a weapon detected.

Current surveillance capabilities must be strengthened as swiftly as possible, with much more resources to enable surveillance that goes beyond the capabilities of human operators. Smart surveillance systems could effectively replace today's archaic surveillance systems.

As low-cost storage, video infrastructure, and greater video processing capabilities become more widely available, current infrastructure will become obsolete.

## References

- [1] R. M. Alaqil, J. A. Alsuhaibani, B. A. Alhumaidi, R. A. Alnasser, R. D. Alotaibi and H. Benhidour, "Automatic Gun Detection From Images Using Faster R-CNN," 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH), 2020, pp. 149-154
- [2] Akshaya, R. Kumar H.S. and D. Bhat, "A novel method to recognize object in Images using Convolution Neural Networks," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 425-430
- [3] H. Jain, A. Vikram, Mohana, A. Kashyap and A. Jain, "Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications," 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), 2020, pp. 193-198
- [4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Neural Information Processing Systems 2015*, 2015, pp. 241–294.
- [5] R. Olmos, S. Tabik, and F. Herrera, "Automatic Handgun Detection Alarm in Videos Using Deep Learning," *CoRR*, vol. abs/1702.0, Feb. 2017.
- [6] G. K. Verma and A. Dhillon, "A Handheld Gun Detection using Faster R-CNN Deep Learning," *Proceedings of the 7th International Conference on Computer and Communication Technology - ICCCT-2017*, 2017.
- [7] M. M. Fernandez-Carrobles, O. Deniz, and F. Maroto, "Gun and Knife Detection Based on Faster R-CNN for Video Surveillance," *Pattern Recognition and Image Analysis Lecture Notes in Computer Science*, pp. 441–452, 2019.
- [8] R. F. Azevedo Kanehisa and A. Almeida Neto, "Firearm Detection using Convolutional Neural Networks," in *11th International Conference on Agents and Artificial Intelligence*, 2019, pp. 707– 714.
- [9] Sandeep Kumar, Aman Balyan, Manvi Chawla, "Object detection and recognition in images," *IJDRI* 704166, vol. 5, pp. 1029-1034, 2017.
- [10] Khushboo Khurana, Reetu Awast hi, "Techniques for object recognition in images and multi object detection," *IJARCET*, vol. 2, pp. 1383 -1388, 2013.