Detecting Unauthorized Individuals with Firearms

Vignesh Prakash
Graduate Student, Univeristy at Buffalo
UBID : 50478782
prakash8@buffalo.edu

***Abstract*—Gun violence has become a major issue in developed nations, particularly in the United States. In 2021 alone, over 200 mass shootings were reported in the US, as defined by the Gun Violence Archive (GVA) as incidents where four or more people, excluding the shooter, were shot. The Centers for Disease Control and Prevention (CDC) reported that in 2019, firearms were responsible for 39,707 deaths in the US, including homicides, suicides, and unintentional deaths. Shockingly, on average, more than 100 people die each day due to firearms in the US, according to the Giffords Law Center. Even children are not immune, with over 2,400 children and teens killed by guns each year. There is no easy solution to this problem, but it is important to have a work around to find ways to reduce gun violence. One potential solution is to focus on the detection of threats posed by unauthorized individuals carrying firearms. To address this issue, the deployment of an autonomous system capable of alerting law enforcement officials of such threats is critical. Object detection algorithms using computer vision have been developed over the years, and this project aims to use these algorithms to create a firearm detection system that can identify unauthorized individuals carrying guns in public and protected areas. The ultimate goal of this project is to enhance security measures and enable swift responses to potential threats, ultimately mitigating gun-related violence and protecting the public.**

***Index Terms*—Gun Violence, Threat Detection, Firearm Detection**

I. Overview of the project

A. Application

This project is centered around creating an innovative application that combines Ultralytics YoloV5 model and Google's Vision Transformer model. The primary goal of this application is to detect guns in an image and identify the person holding the gun as either authorized or unauthorized based on several characteristics such as armor, uniform, and badge on the cap, among others.

By inputting an image, the application can determine the presence of guns by generating an array of bounding boxes indicating the number of detected firearms in the image. There are two outputs in the model, one is the array of bounding boxes indicating the detected guns in the image frame and dictionary of probabilities that can identify whether or not an authorized person is present in the image, providing valuable insight to the user.

B. State of the Art

Object detection in computer vision has made significant advancements with the introduction of advanced deep learning models based on convolutional neural networks (CNNs). These models require large datasets to accurately detect objects within an image and localize them with high precision.

In computer vision, object detection falls into two categories, single-stage and two-stage detectors. Although single-stage detectors are primarily designed for static images, they can also work well with videos. Two-stage detectors leverage temporal information contained in videos and can be used to improve detection accuracy.

The Faster R-CNN model is one of the most popular deep learning models for object detection, employing a two-stage process to detect objects in images. Other popular models include the Single Shot Multibox Detector (SSD) and You Only Look Once (YOLO)

This project utilizes YOLOv5 as an object detection model, as it is a popular single-stage model that achieves high precision even for videos. This problem of detecting guns in an image frame is previously applied by many researchers. This paper [1] applies Quasi-Recurrent Neural Networks (QRNNs) to extract spatiotemporal features for handgun detection. The model is built using 8000 annotated images and achieves a mAP (Mean Average Precision) of 90 at 56.8 fps. Similarly this project [2] uses 5000 well-chosen images, in which 16064 instances of guns and 9046 instances of persons are annotated. This model claims to achieve an AP (Average Precision) of 52.1.

C. My Contributions

In this project, I trained multiple pretrained YOLOv5 models on a custom dataset of images containing handguns. The custom dataset was created by augmenting images from the existing dataset from the research paper "TYolov5: A temporal YOLOv5 Detector based on Quasi-Recurrent Neural Networks for Real-Time handgun detection in Video".

After training the models on the custom dataset, I fine-tuned them to detect handguns and classify whether the person in the image is authorized or not. I used a technique called transfer learning to fine-tune the models. Transfer learning is a machine learning technique where a model that has been trained on one task is used as a starting point for training a model on a new task. In this case, I used the pretrained YOLOv5 models as a starting point for training the models to detect handguns and classify authorized and unauthorized persons.

After fine-tuning the models, I selected the best model and deployed it to HuggingFace Spaces. The demo for inference is available at this Link. This demo allows users to test

the model's accuracy in detecting handguns and classifying authorized versus unauthorized persons.

II. Approach

A. Algorithms used

1) YOLOv5 Model: YOLO, short for "You Only Look Once," is an object detection algorithm that was introduced by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi in 2015. It was pre-trained on the COCO dataset, which includes over 330,000 images of objects in 80 categories.

The algorithm breaks an image down into multiple regions and predicts probabilities and bounding boxes for each region. This approach has proven to be fast and accurate, and the Ultralytics team is currently working on improving it further.

The YOLOv5 model consists of three parts: the backbone, neck, and head. The backbone is a pre-trained network that extracts rich feature representations for images, reducing the spatial resolution and increasing the feature resolution. The neck extracts feature pyramids that enable the model to generalize well to objects of different sizes and scales. The head applies anchor boxes to the detected objects, renders classes, objectness scores, and bounding boxes.

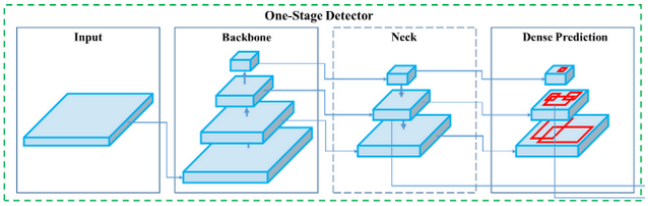

Fig. 1. YOLO Architecture

There are five versions of the YOLOv5 model, ranging from the nano model to the extra-large model. The CSP-Darknet53 serves as the backbone, with SPP and PANet in the model neck, and the head used in YOLOv4.

CSP, or Cross Stage Partial Network, partitions the feature map of the base layer into two parts and then merges them through a cross-stage hierarchy. SPP, or Spatial Partial Pooling, aggregates information from the input, providing a fixed-length output that increases the receptive field while segregating the most relevant context features. PANet, or Path Aggregation Network, is a feature pyramid network that improves information flow and helps localize pixels in mask prediction tasks.

The head is composed of three convolution layers that predict the location of bounding boxes (x, y, height, width), scores, and object classes. SiLU, or Sigmoid Linear Unit, is the activation function used in the hidden layers of the model. The basic sigmoid function is used in the output of the model. The class loss and objectness loss is computed using BCE (Binary Cross Entropy) and the bounding box location loss is computed using CIoU (Complete Intersection over Union).The total loss of the model is a weighted sum of the three losses: class loss, objectness loss, and bounding box location loss. [3].

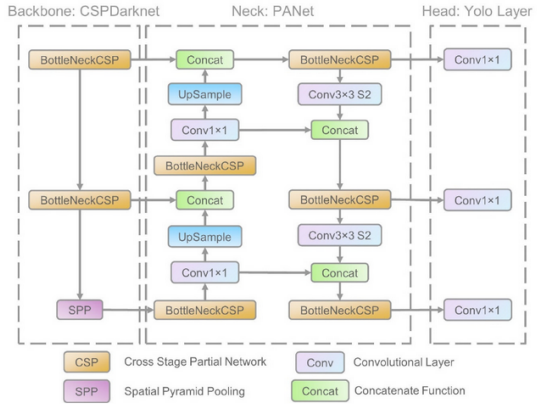

Fig. 2. YOLOv5 Architecture

The YOLOv5 model is a very light and efficient model when compared to other YOLO models.

$$TotalLoss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc}$$

2) Vision Transformer: The research paper titled "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," authored by Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, introduces the Vision Transformer (ViT) model. This model successfully applies the Transformer encoder, which is widely used in natural language processing tasks, to image recognition tasks. The ViT model achieves competitive results when compared with conventional convolutional architectures.

To use the Transformer encoder for image recognition, the attention mechanism is used to replace certain CNN components or applied along with the CNN network while keeping the overall structure in place. Each image is segmented into a sequence of fixed-size, non-overlapping patches, which are then linearly sent into the model. The absolute position embedding is fed into the resulting sequence of vectors. To use the Vision Transformer, all images must have the same size and resolution. Therefore, images are preprocessed using the VitImageProcessor to normalize them for the model. [4]

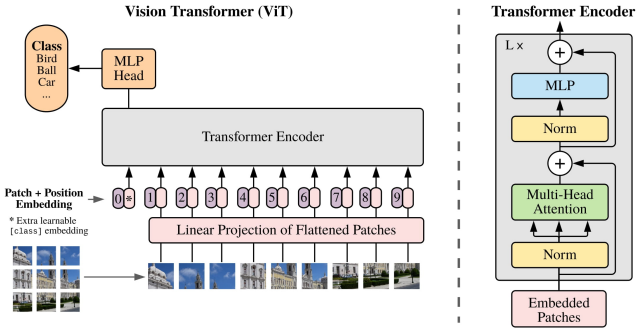

Fig. 3. ViT Architecture

B. My Implementation

In this project, I evaluated the efficiency of three versions of the YOLOv5 model on the same dataset in terms of computation and memory. The YOLOv5s is a small model with 7.2 million parameters, which achieves a fast detection speed of 98ms per image. The YOLOv5m is a medium-sized model with 21.2 million parameters, which balances detection accuracy and speed with a frame rate of 224ms per image. The YOLOv5l is a large model with 46.5 million parameters, which achieves higher detection accuracy but at the cost of slower detection speed, processing images at a frame rate of 430ms. Due to memory constraints, the first 15 layers of the model are frozen for all three versions. The current YOLOv5 model architecture consists of 157 layers. This number of freezed layers is found empirically by altering the values and computing the results.

For the image classification part, I used the googlevit-base-patch16-224 pretrained model, which has a base-sized architecture with a patch resolution of 16×16 and fine-tuning resolution of 224×224. This model is pretrained on ImageNet-21k, a collection of 14 million images and 21k classes, and is used as my base architecture without much modification. The training parameters of the model include a batch size of 16 and a learning rate of 2e-4.

C. External Resources

The project primarily relies on internal resources, with minimal external dependencies. The codebase draws heavily from the official documentation of Ultralytics YOLOv5 about training on custom data.The implementation of the VitImage-Classification is also taken from the official documentation provided by the HuggingFace Transformer. The contribution in my part is the fine-tuning process which is largely attributed to the careful structuring of the dataset, selecting the appropriate model, and identifying optimal hyperparameters like epochs, learing rate and the resolution of the images.

III. Experimental Protocol

A. Dataset

Two datasets were utilized in this project. The first dataset consists of approximately 5000 high-quality images that were extracted from various YouTube videos and series. This dataset includes screenshots from YouTube videos, surveillance footages, and random images of people holding guns. These images have varying dimensions and resolutions, and were uploaded to Roboflow, an online image annotating tool primarily used for computer vision applications. The annotations in the dataset were manually verified and incorrect labels and null values were removed. The images were then resized to 224 × 224 pixel images to reduce computation power and time needed for training the YOLOv5 model.

The second dataset was obtained from Google Images through SerpAPI, an online platform to download Google search results via an access key generated for an account. The search queries used to extract the images were 'police with armor', 'military with armor', 'police officers in tactical gear',

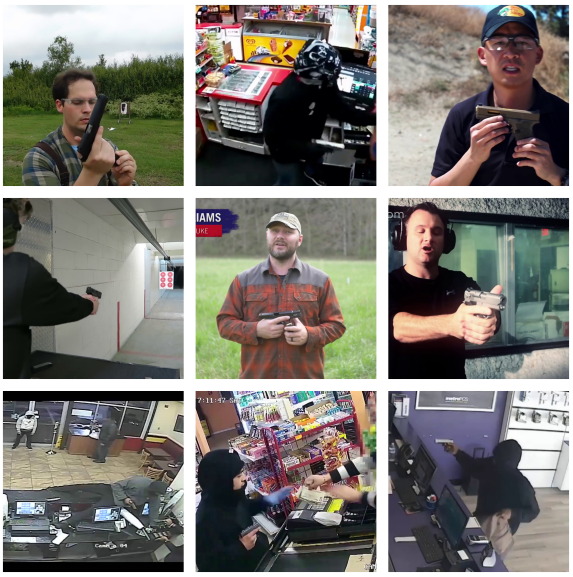

Fig. 4. Dataset for YOLOv5

'sheriff officers with armor', and 'FBI uniform with vest', each resulting in 100 images. Pedestrian images on the street were used for negative images, and about 900 valid images were used to create the dataset. These images were uploaded to Roboflow for preprocessing and data augmentation. They were also resized to 224 × 224 pixels to fit the Google/vit-base-patch16-224 model's input, which requires images in the form of 224 × 224 resolution.

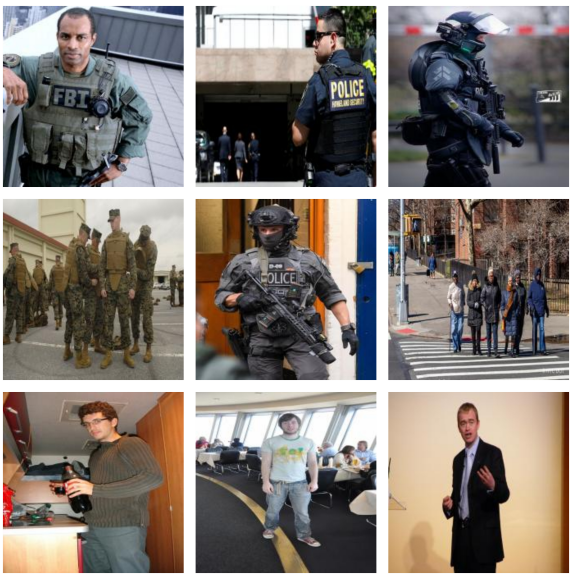

Fig. 5. Dataset for VitImageClassification Model

Link to Dataset 1 - Guns in an Image

Link to Dataset 2 - Police vs Public

B. Evaluation

The mean Average Precision (mAP) of the three YOLOv5 models is compared to determine the best performing model. The project's success is measured by the model's effectiveness in real-time detection and tracking of unauthorized individuals with firearms, and several parameters quantify its accuracy, such as Average Precision (AP), Mean Average Precision (mAP), F1-score, and Accuracy.

AP measures the model's accuracy in detecting objects within an image and can be calculated for each class or averaged over all classes. mAP is the average of AP scores across all classes, providing an overall measure of the model's performance. As this model is having only one class for detection the average precision for the detected gun class is the mean average precision. F1-score is the harmonic mean of precision and recall, useful in scenarios where both metrics are equally important. Accuracy measures the percentage of correctly identified objects out of all objects present in the image.

The project's success is determined by comparing the achieved mAP values to state-of-the-art technologies and analyzing how closely they match.

C. Computational Resources

The project was executed on the free-tier GPU T4 architecture available on Google Colab. This architecture has a RAM of around 8GB and memory of 50GB. However, this architecture is relatively basic, which limited its capability for training purposes. To address this issue, the Yolov5 layers were frozen and the image dataset was resized to 224 $\times$ 224 pixels. This was done to ensure computational efficiency during multiple trials on the system, without exceeding the free-tier quota.

IV. Results

A. Visualization

The results of the three models are listed down with various metrics like mAP for 50%, Precision and Recall. The red line represent the YOLOv5L (large model),The blue line represent the YOLOv5m (medium model),The green line represent the YOLOv5s (small model).

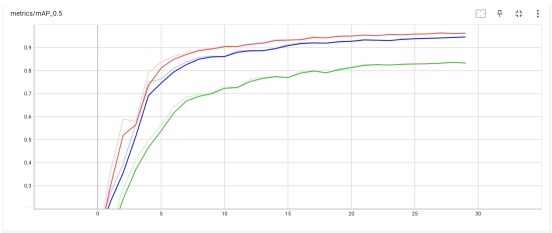

Fig. 6. Mean Average Precision

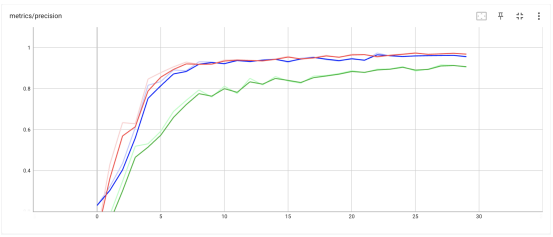

Fig. 7. Precision

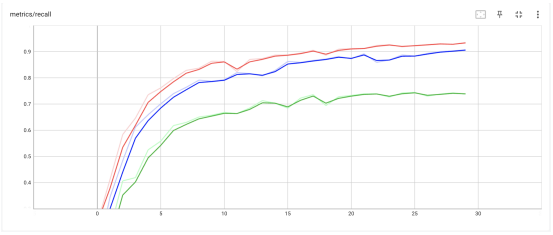

Fig. 8. Recall

The below are the loss graphs for the three models,

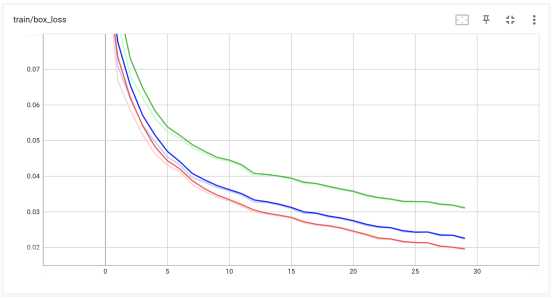

Fig. 9. Box Loss

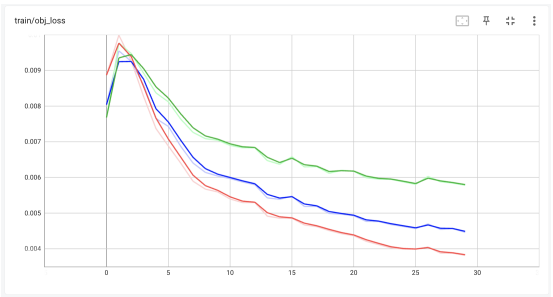

Fig. 10. Object Loss

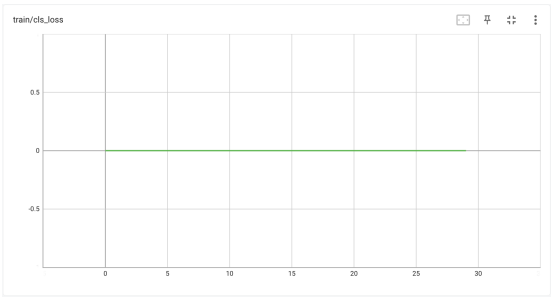

Fig. 11. Class Loss

The class loss here is just a flat line, because, there is only a single class present in the YOLOv5 model for detection

B. Figures & Tables

The F1 scores of the three models are as follows,

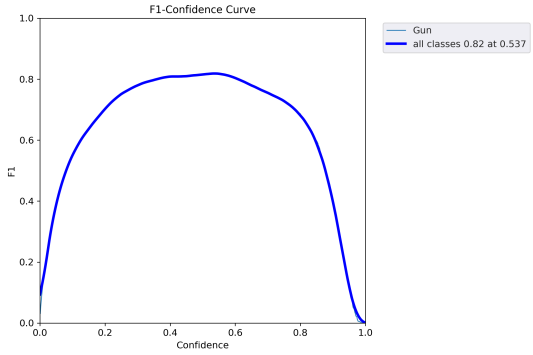

Fig. 12. Yolov5s F1 score

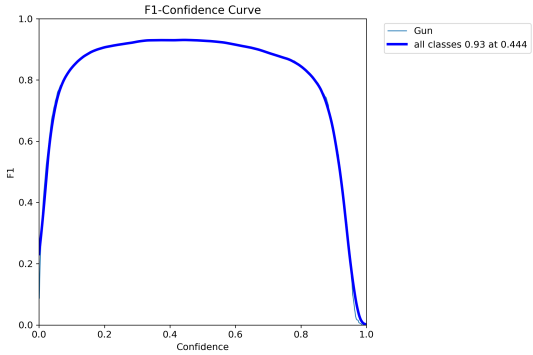

Fig. 13. Yolov5M F1 score

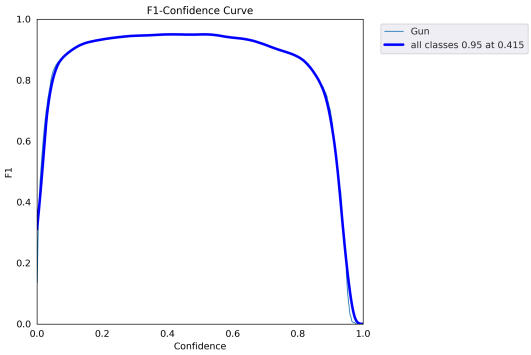

Fig. 14. Yolov5L F1 score

Model Used	**F1 Score**
Yolov5s	0.82
Yolov5m	0.93
Yolov5L	**0.95**

TABLE I
COMPARISON OF F1 SCORES

Based on the results of the PR curves from the output of three models,

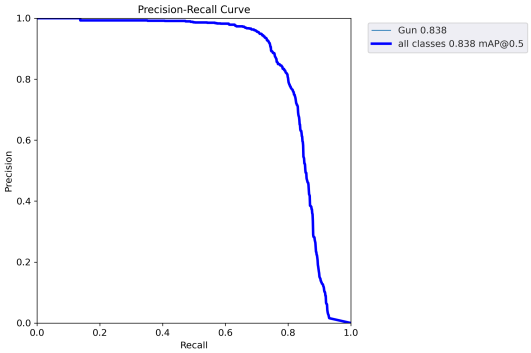

Fig. 15. Yolov5s PR Curve

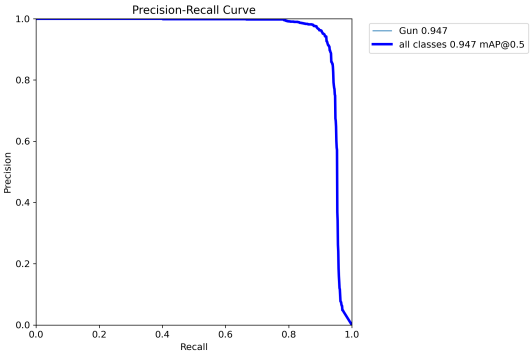

Fig. 16. Yolov5M PR Curve

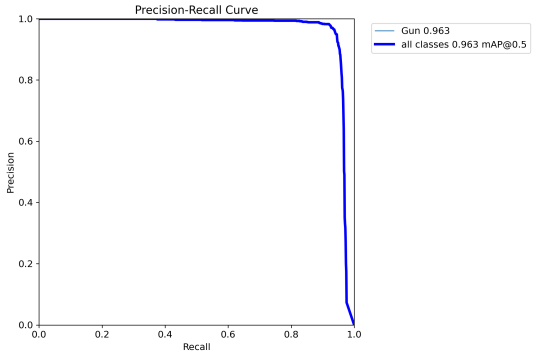

Fig. 17. Yolov5L PR Curve

Model Used	mAP_0.5
Yolov5s	83.8
Yolov5m	94.7
Yolov5L	**96.3**
TYolov5s	87.9
TYolov5s	90.1
TYolov5s	90

TABLE II
COMPARISON OF MAP SCORES

Training Results of VitImageClassification model are as follows,

Step	Training Loss	Validation Loss	Accuracy
10	0.418800	0.159327	0.959302
20	0.193200	0.189685	0.930233
30	0.041000	0.050845	0.988372
40	0.019700	0.074998	0.970930
50	0.015800	0.080269	0.970930

```
***** train metrics *****
  epoch                    =        4.0
  total_flos               = 57158671GF
  train_loss               =      0.133
  train_runtime            = 0:01:02.22
  train_samples_per_second =     12.728
  train_steps_per_second   =      0.836
```

Fig. 18. Image Classification Model-Training Results

Model is deployed in Hugging Face Spaces for testing

C. Interpretation

The results show that the YOLOv5L model achieved a higher mAP_0.5 value, indicating that it outperformed the other models. This improvement in mean average precision could be attributed to various factors, such as the architecture of the model and the number and resolution of the images used for training. Specifically, the low number of images and lower resolution may have caused the model to overfit to the training data, resulting in better performance on the test data.

While the Image Classifier performs well on the training set, its performance on the test dataset is poor, as observed from the trials conducted on the playground deployed on Hugging Face's Playground site. The model's performance may be attributed to bias in the dataset, specifically related to the black-colored armor. The classifier struggles to distinguish between a suit and an armor, which can result in incorrect predictions. Furthermore, the small size of the dataset may also have limited the model's ability to learn features effectively from the images. Further investigation into improving the dataset and model architecture may be necessary to improve the classifier's performance. One such improvement is done by manually thresholding the image using cv2's adaptive threshold function. On contrary, the thresholding resulted in a much lower accuracy, as many spatial features are lost during thresholding and that decreased the model's ability to classify.

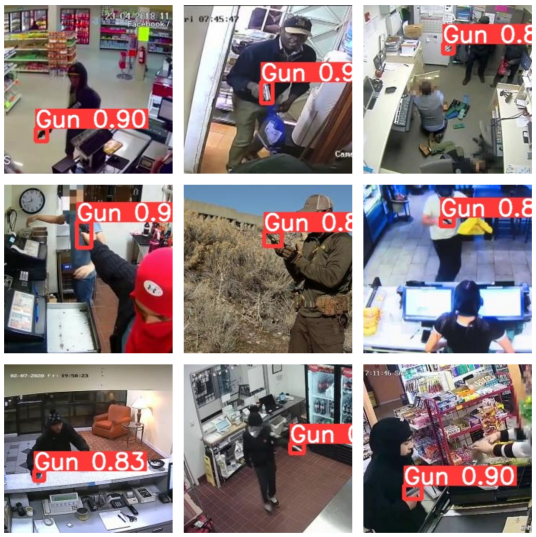

Fig. 19. Training Results

V. ANALYSIS

A. Advantages

One advantage of the model is that it performed well given the size of the dataset and available computation power. However, it is possible that the model's performance could be further improved by training on a larger dataset with higher resolution images and greater variation in the background.

To improve the image classification model, reducing bias in the dataset and annotating images with bounding boxes of specific features could be helpful. These bounding boxes may assist the model in focusing on relevant features of the image rather than processing the entire image as a whole. Additionally, exploring more advanced model architectures and training techniques may further enhance the model's performance.

B. Disadvantages

This model has certain limitations. Firstly, it can only detect hand guns as it was trained solely on that dataset, which restricts its utility in identifying other types of firearms or weapons. Additionally, the limited scope of the model may not be suitable for more comprehensive security or surveillance purposes.

In terms of the image classification model, the primary disadvantage lies in the bias present in the dataset. This bias can lead to incorrect predictions or an incomplete understanding of the features present in the images. Addressing this issue by improving the dataset and implementing more advanced techniques, such as data augmentation or transfer learning, may be necessary to overcome this limitation.

VI. Discussion

This project has provided valuable insights into the complexities involved in detecting objects accurately and consistently from images. It has highlighted how bias in the dataset can impact even pre-trained models, as well as emphasized the importance of carefully studying and preprocessing images to ensure optimal performance.

To improve the model in the future, incorporating images of rifles and other weapons could help broaden its scope and usefulness. Additionally, augmenting the image dataset with occlusions and other real-world scenarios could improve the model's ability to operate effectively in a range of environments. Continuing to explore more advanced techniques, such as transfer learning and neural architecture search, could also be beneficial in further enhancing the model's performance.

References

[1] Mario Alberto Duran-Vega, Miguel González-Mendoza, Leonardo Chang, and Cuauhtemoc Daniel Suarez-Ramirez. Tyolov5: A temporal yolov5 detector based on quasi-recurrent neural networks for real-time handgun detection in video. *CoRR*, abs/2111.08867, 2021.

[2] Gu Yongxiang, Liao Xingbin, and Qin Xiaolin. Youtube-gdd: A challenging gun detection dataset with rich contextual information. *arXiv preprint arXiv:2203.04129*, 2022.

[3] Cherifi Imane. Yolo v5 model architecture.

[4] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Zhicheng Yan, Masayoshi Tomizuka, Joseph Gonzalez, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020.