Weapon Detection System with Auto Alert System

Sankar Sivasamy
221026429
Dr. Marcus Pearce
MSc in Artificial Intelligence

***Abstract*— The necessity of addressing crimes and ensuring security continues to take on paramount relevance in a society that is becoming more interconnected. More effective and creative approaches are urgently needed to tackle criminal activities and enhance security. We have developed a very efficient weapon detection system using YOLOv7, an advanced object detection model, and the comprehensive Sohas Weapon Detection Dataset in response to these difficulties. The system provides real-time detection and fast alarm capabilities and has been trained and customized to optimize weapon recognition. This technology seeks to supplement human security efforts, providing an additional layer of surveillance that can operate with consistent vigilance. Notably, the system performs well, with an accuracy rate that exceeds 80%. Furthermore, a mobile application was created using ReactJS in recognition of the demand for mobility and ease in security solutions. The real-time, on-the-go weapon detection made possible by this cutting-edge JavaScript library, which is renowned for its effectiveness and adaptability, expands the scope of our system and may change security measures in various scenarios.**

Keywords—Weapon Detection, YOLO v7, Artificial Intelligence (AI), Sohas Weapon Dataset, Real Time Video

I. Introduction

In many different situations around the world, weapon detection systems are essential for ensuring public safety and security [1]. There is an increasing need for more effective and precise detection systems to stave off possible catastrophes as public security risks increase. Traditional weapon detection systems frequently rely primarily on human observation and manual security checks, which are prone to human mistake and can be ineffective in situations with high traffic. This highlights the essential need for automated, real-time detection systems that are able to quickly identify possible dangers and allow for quick action. The emergence of machine learning (ML) and computer vision (CV) techniques has altered the object detection landscape and presented interesting alternatives to traditional methods [2]. These methods serve as the foundation for our study, in which we use an ML model to detect weapons effectively and accurately. Modern object detection model YOLOv7, which is being used, is renowned for its quickness and precision in real-time object detection tasks [3].

The capability of our technology to automatically send out alerts upon weapon detection is a crucial component. The model's efficiency depends heavily on this immediate alert system, which enables quick reactions to possible security concerns [4]. Such feature could be extremely helpful in a variety of high-risk environments, such as schools, airports, or public events, where the quick detection of threats may be able to avert major incidents. The utilization of the Sohas Weapon Detection Dataset underpins our work in this area. This dataset contains a large number of weapon photos that have been frequently used in weapon detection research and applications. It includes a wide range of weapon kinds, providing our model with a thorough understanding of the many weapons it may confront. This is essential for improving the model's generalizability and accuracy in a variety of real-world scenarios [5]. We used the Sohas Weapon Detection Dataset to fine-tune YOLOv7 as part of our model development process. With the use of this method, we were able to take advantage of YOLOv7's advantages while also adapting the model's architecture to our particular use-case [3]. We used Google Colab, a cloud-based platform renowned for its strong and adaptable environment for ML and deep learning applications, to conduct our training procedure. This paper describes our efforts to create a weapon detection system that makes use of YOLOv7 and is enhanced with an auto alert function using the Sohas Weapon Detection Dataset. We believe that by providing an ML-based tool capable of quick, real-time weapon detection and alarm creation, our work greatly advances the field of public safety and security.

II. Need Of Ai Security System

The increasing rates of gun violence and the shortcomings of conventional surveillance systems are the main causes of the urgent need for AI-based weapon monitoring and security systems. Regrettably, social complexity and rapid urbanization have increased the frequency of crimes, particularly those involving firearms. Over one billion firearms are in use worldwide, with a startling 85% of them owned by civilians, according to the Small Arms Survey [6]. The growing number of gun-related events in both public and private settings is proof that this ubiquity has a direct impact on public safety. A creative approach that can quickly identify and warn security personnel about potential dangers is required given the circumstances. Despite being a crucial component of the security architecture, traditional surveillance systems have many drawbacks. The main difficulty is that they rely on human operators for monitoring, which can cause problems with weariness, inattention, and neglect and reduce their effectiveness. Additionally, they lack the ability to act quickly and in real-time, which is frequently essential in preventing disasters. A promising response to these issues is provided by AI-based surveillance systems. The topic of object detection, particularly the detection of weapons, has shown tremendous promise for ML and CV approaches [7].

Without becoming tired or losing focus, these systems can continuously watch numerous video feeds and spot potential dangers. Additionally, they offer the capability of automatic alarm production and real-time detection, enabling quick response and intervention. The ability of AI-based systems to adapt and learn new things continuously is a major advantage. They can receive training on new threats and weapons, continuously enhancing their performance as they are exposed to more information [8]. Additionally, these systems can be combined with other security measures, such as behavior analysis and facial

recognition, to provide a comprehensive security and monitoring strategy. Therefore, it is imperative that we develop and deploy AI-based weapon surveillance and security systems to improve public safety. They provide a pro-active and effective response to both the shortcomings of traditional surveillance systems and the rising threats of gun violence.

III. Literature Review

A. AI Assisted Survillence and Concealed Weapon Detection

Particularly in the context of violence and concealed weapon detection, the use of Artificial Intelligence (AI) and sensor technologies in surveillance systems is a rapidly growing topic of study. Two key articles in the field—"AI-Assisted Edge Vision for Violence Detection in IoT-Based Industrial Surveillance Networks" by Fath U Min Ullah et al.[9] and "Combining Commercially Available Active and Passive Sensors Into a Milli meter-Wave Imager for Concealed Weapon Detection" by Garcia-Rial [10] are thoroughly analyzed in this review of the literature.

In order to detect violent behaviours, article [9] focuses on the integration of edge computing and AI in industrial surveillance networks. The widespread use of AI in industrial and public security has sped up the creation of cutting-edge surveillance systems. We've used this development in our project to build a powerful weapon detection system utilizing YOLOv7. We have employed video frames as the input for our system, which was inspired by the IIoT-based framework and VD-Net suggested in previous literature. This deep learning model analyzes these frames to find possible threats classified as weapons, producing immediate alarms in the IIoT network as a result. Our approach to data efficiency is similar to the VD-Net's, which only analyzes frames that include threats. As a result, the computational load is reduced and system performance is improved. Similar to the VD-Net's ablation research, we continuously train and optimize our YOLOv7 model to increase accuracy and dependability.

Garcia-Rial et al. [10] examine the merging of active and passive sensors in a millimeter-wave imager for concealed weapon detection, and as a result, our approach is in line with the development of AI-based surveillance systems that focus on weapon detection. The authors argue that a multimodal system can more efficiently identify hidden objects by combining active millimeter-wave radar with passive millimeter-wave radiometer sensors, both of which are commercially accessible. According to Garcia-Rial's et al.[10] theory, the information produced by active and passive sensors complements one another and boosts the performance of the system used to detect concealed weapons as a whole. They assert that the combination of these sensors can decrease the rate of false alarms and accurately differentiate between dangerous and harmless concealed objects. Additionally, they assert that while the focus of their research is on commercially accessible sensors, the concept may be generalized to other types of sensors, increasing its application in a variety of security scenarios.

B. Innovative Technology and Algorithms in Weapon Detection and Sensor Alignment

Briqech et al. [11] explore the use of a 57-64 GHz imaging/detection sensor for the concealed weapons and threatening materials detection. The authors present experimental findings that show how effective this sensor is at finding hidden weapons. Their research indicates that the 57-64 GHz imaging/detection sensor considerably improves the detecting skills, providing benefits including high resolution, compactness, and unobtrusive operation. These advantages make this sensor especially helpful in public safety situations where discrete monitoring is essential. The authors go on to show how their sensor system may be used with AI algorithms to identify threats in real time. The performance of the surveillance system is considerably enhanced by the use of ML techniques, which boost the system's flexibility to various threatening scenarios.

An effective marginal-return-based constructive heuristic is introduced by Bin Xin et al. [12] in their research to address the Sensor-Weapon-Target Assignment (SWTA) dilemma. A decision-making issue in weapon-target assignment systems is known as the SWTA problem. It entails choosing which weapon to use to neutralize the discovered target and which sensor to utilize to detect the target. The authors suggest an effective constructive heuristic based on marginal returns as a remedy, claiming that it is more reliable, adaptable, and effective than other heuristic techniques. The use of their technique might enhance how security systems allocate sensors for weapon detection. By making better use of the available resources, the surveillance system may operate more effectively and make the most of each deployed sensor and weapon.

Thus, the study of Briqech and Bin Xin adds to our understanding of AI-based security and surveillance systems. The former highlights the usefulness of the 57-64 GHz imaging/detection sensor in the detection of concealed weapons, while the latter suggests an effective remedy for the SWTA issue. Together, these works provide insightful information about how AI-based weapon detection systems will develop and advance.

C. Incorporating Cyber Weapons Assessment and Security

A new age of state-sponsored cyberwarfare, where powerful governments use sophisticated harmful software for political gain, was inaugurated by the discovery of the Stuxnet worm in 2010. These computerized weapons, also referred to as cyber weapons, are the newest development in military technology. Due to the linked nature of cyberspace, the threats that these technologies represent are not just restricted to military aims but also have a substantial influence on civilian IT systems. The substantial differences between cyber weapons and conventional weapons in terms of development, strategic application, and tactical deployment present one of the regulatory issues. It is crucial to identify the distinctive features of these digital weapons in order to create effective cyber arms control treaties. This article[13] explores the present discourse on cyber weapons critically and identifies its flaws, such as how it frequently relies significantly on assumptions about adversarial actors or how it only applies after the use of destructive tools. The article[13], on the other hand, promotes a focus on the unique operational aspects of malware and suggests an indicator-based assessment methodology based on metrics

measurable prior to the installation of the malicious software. This futuristic approach makes it possible to classify rogue technologies as cyber weapons, greatly advancing the discussion on cyberwarfare and regulation.

This research offers an intriguing viewpoint with regard to our project on weapon detection. Although the majority of our project's attention is given to the detection of physical weapons, the ideas and techniques covered in this article can be used as a guide for the project's prospective future extension to include the detection of cyber weapons. Similar to the detection of physical weapons, the detection of cyber weapons likewise necessitates a thorough comprehension of their unique operational features and a strong detection model capable of foreseeing possible threats. The improved data security, which guarantees the integrity and privacy of the information within the surveillance system, would be the key benefit of such a cyber weapon detection system. The development and deployment of a cyber weapon detection model, however, could be resource-intensive, both in terms of monetary outlay and technical expertise, just like physical weapon detection systems. This emphasizes the significance of ongoing research and development in the area to improve the usability and effectiveness of these systems. This examination of cyber weapons provides perceptive views, highlighting the significance of social responsibility in the development of computer science and security technologies. As we move forward with our weapon detection research, we must keep in mind the social and ethical ramifications of our work as well as potential cyberthreats in addition to physical threats.

IV. Methodology

A. *Proposed System*

The proposed system offers a reliable and incredibly effective approach for detecting weapons across various settings, particularly public spaces. The main idea behind this project is to use a cutting-edge Deep Learning algorithm, YOLOv7-tiny, to its fullest potential in order to detect weapons in CCTV camera surveillance video feeds. The implementation of this advanced deep learning model makes it possible to detect weapons accurately and in real time, increasing the efficiency of security monitoring systems. Data from the Sohas weapon detection dataset, which is renowned for its extensive and varied collection of weapon photos, will be used to train the model. This dataset's use in training the deep learning model guarantees that it has a thorough understanding of the many types and forms of weapons, which improves detection accuracy [14]. The Sohas dataset's strength resides in its size and diversity, which enables the model to be exposed to a wide range of weapon types and configurations. In spite of the complexity of various viewing angles, lighting conditions, and weapon sizes, this enables the system to recognize and detect weapons in actual surveillance recordings.

An output file for the model will be created after the training process is over. The main component of the weapon detection procedure is this file, which contains the knowledge learned during training. This trained model is used to detect weapons in CCTV footage in real-time. ReactJS will be used to create a mobile application that will guarantee quick action when a weapon is detected. This program is intended to receive alerts anytime the system detects a weapon. By assuring rapid alerts, the notification feature adds an extra layer of protection by enabling the right employees to act quickly in response to a potential threat. Essentially, the proposed system provides a rapid, affordable, and trustworthy option for weapon detection. It makes use of the YOLOv7-tiny model, a deep learning system, to deliver precise real-time detection of weapons in CCTV surveillance videos. The proposed methodology enhances public safety by enabling speedy reactions to possible threats when used in combination with the mobile application's instant alert system.

B. *Architecture of proposed model*

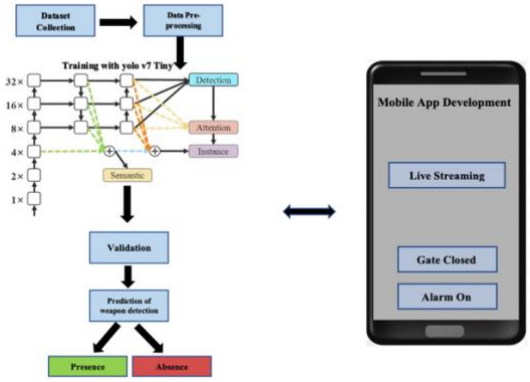

Fig. 1. System architecture of proposed model

In the development of our weapon detection system, we incorporated the architectural framework of YOLOv7 and leveraged the weights of yolov7 -tiny, sourced directly from the original author's GitHub repository[15]. Given the unique demands and specifications of our project, we found it necessary to modify the architecture, tailoring it to best align with our objectives. This customization was pivotal in ensuring the system's adaptability and precision in addressing the challenges specific to weapon detection. At last added mobile app development code to its architecture.

The YOLOv7 system architecture, an advanced deep learning model recognized for its remarkable object detection abilities, serves as the foundation for the proposed model's system architecture in large part. The sophisticated and effective processes that support this model's performance are clearly shown in the architecture diagram in Fig 1. The suggested model, which is tailored to the specific goal of weapon detection in CCTV surveillance videos, basically expands and adapts this architecture. The system is initially fed with raw input in the form of surveillance video feed. The YOLOv7 model, functioning as the core of the system, processes these real-time video frames to detect objects. The accurate and prompt detection of weapons is made possible by the fast processing speed and precise object detection capabilities of YOLOv7.A number of convolutional layers make up the YOLOv7 architecture, which extracts features from the input frames. These layers are designed to find patterns that could point the presence of a weapon, or another object of interest. Following the feature extraction procedure, these layers transmit the features to the following architecture component, which consists of a group of completely connected layers.

The classes represented in the training dataset are here assigned to the detected features. The fully linked layers seek to assess if those detected objects belong to the 'weapon' class, taking into account our particular use-case

of weapon detection. The model has acquired knowledge of the "weapon" class in this context through exposure to the Sohas weapon detection dataset during the training phase. Finally, the system's post-processing component receives the detection results. Here, an alert is generated and the detected weapons in the video frames are highlighted. The mobile application built with ReactJS receives this signal right away, enabling speedy response to any detected threat. As a result, the YOLOv7 model's advantages are combined with the usefulness of an instant alert mechanism in the proposed system architecture. The proposed system is an extremely effective and dependable method for detecting weapons in CCTV surveillance videos because it combines advanced deep learning techniques with real-time alarms.

C. Why YOLOv7 Good at Object Detection

YOLOv7, which stands for "You Only Look Once version 7," is well known for its remarkable object detection abilities, making it an obvious choice for our effort to find weapons. Contrary to earlier object detection techniques that operate in two steps, YOLOv7's design completes the operation in a single step, considerably speeding up processing [14]. For real-time applications, it is essential that the algorithm be able to look at the full image at once and recognize several items.

A significant advancement in CV and ML may be seen in the YOLOv7 algorithm. This most recent version outperforms earlier YOLO iterations and other object detection models in terms of both speed and accuracy. It stands out as a potential future industry standard for object detection, potentially surpassing YOLO v4 in real-time applications thanks to its less expensive hardware requirement and capacity to be trained more quickly on smaller datasets without pre-existing weights. In CV, the real-time object detection made possible by YOLOv7 is significant [16]. Applications including robotics, self-driving cars, medical image analysis, and others are supported by this technology. This system processes an image, detects items by predicting bounding boxes, and categorizes each object by predicting class probabilities. Convolutional Neural Networks (CNNs) are frequently employed to extract image features to support this prediction process.

A scaled-down version of the YOLOv7 model called YOLOv7-tiny is especially made for computing devices like mobile phones and Raspberry Pis. Despite being a smaller model, it still manages to blend speed and accuracy very well. It is designed to function effectively on systems with limited processing capabilities without making too many compromises to detection performance [17]. Yolov7-tiny was chosen for our project because we needed a model that would work quickly and precisely even in situations where high computational power might not be easily accessible. This makes sure that the weapon detection system is not only reliable but also flexible and able to be implemented in a variety of situations. One explanation for YOLOv7's performance in object detection is its distinct single-pass approach to the image. As a result, it can predict bounding boxes and class probabilities simultaneously, enabling high-accuracy real-time predictions at a relatively cheap computing cost. The original research paper contains the loss function formula for YOLOv7-tiny, which is designed to balance classification, localization, and confidence errors, resulting in the efficiency of YOLOv7 in object detection tasks.

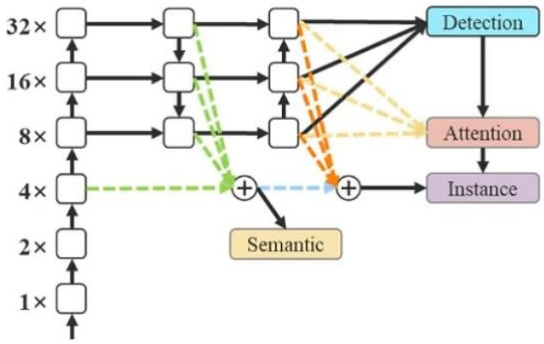

Fig. 2. YOLOv7 Algorithm

Fig 2 shows a simple architecture algorithm of Yolov7

D. Dataset

We use a large dataset from Kaggle and other sources, a well-known online community for data scientists and machine learning practitioners, in this study. After many studies we decided to work on with Sohas weapon detection dataset. The deep learning technique we use uses this dataset as the primary training data. According to the adage in ML, performance improves with increased data. Deep Learning has become an innovative tool for addressing a wide range of real-world issues, with its strength being notably evident in the field of CV. Deep learning has a lot of potential, as evidenced by its ability to recognize and separate distinct items in an image. However, the amount of data that serves as their fuel has a significant impact on how effective these deep learning models are.

The performance of our model is considerably improved by the availability of large, labelled datasets. Three main processes make up the data collection procedure for this project: online scraping, using third-party services, and manual collection. An automated technique called web scraping is used to quickly retrieve massive amounts of data from websites. Web scraping is extremely helpful for tasks like image classification, when coarse image annotations are sufficient. To produce accurate bounding box and segmentation labels, nevertheless, it is necessary to apply powerful image annotation tools. Additionally, third-party services have become effective means of gathering data. These services provide customized data collection and labeling solutions that meet the project's needs. While some businesses, like Mighty AI, focus on particular niches like self-driving car image annotation, others, like Payment AI, offer a wider selection of capabilities, such video and landmark annotations.

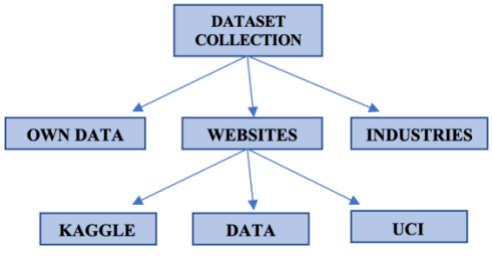

Fig. 3. Dataset Collection

Six categories are used to categorize the data for our project: pistol, smartphone, knife, monedero (wallet), tarjeta (card), and billete (money) [18]. A number of sources, including the internet, social media, and private collections, were used to gather the photographs for the Sohas weapon dataset. Bounding boxes that identify the locations of the objects in the photos are used as annotations. The dataset also contains ground truth labels that specify each object's class.

Researchers who are developing algorithms for weapon detection can benefit greatly from the Sohas weapon dataset. The dataset can be used to develop and test models for scene interpretation, object detection, and image categorization. While the classifications "pistol" and "knife" immediately contribute to the detection of weapons, the addition of "smartphone," "monedero," "tarjeta," and "billete" adds another level of complexity to the model. The chance of false positives in the detection process is decreased since it trains the model to differentiate between weapons and everyday items. The weapon detection system's accuracy and reliability are improved by this strategy.

Fig. 4. Sohas Dataset

The DaSCI website offers a download for the Sohas weapon dataset. The Creative Commons Attribution 4.0 International (CC BY 4.0) license governs the dataset[17]. The Sohas weapon dataset is an expanding resource, and new pictures are frequently added. This guarantees that the dataset is current and useful.

TABLE I. DATASET CLASSIFICATION

Sohas Weapon Dataset		
Classification Labels	**Train**	**Test**
Pistol	1580	294
Knife	1879	470
Smartphone	755	115
Billete(Bill)	545	123
Tarjet(Card)	340	64
Monedero(Purse)	581	104
Total Labels	**5680**	**1170**

V. IMPLEMENTATION ANALYSIS OF PROPOSED MODEL

There are various stages involved in the creation and evaluation of our proposed weapon detection method. These steps include loading weight and hyperparameter files, training and validating our dataset using YOLOv7, and real-time detection with Flask and OpenCV (cv2).

Preparing the data is the first step in model implementation. Pre-processing is done on the Sohas weapon detection dataset, which consists of six different classes: handgun, smartphone, knife, monedero, tarjeta, and billete. The prepared data is then loaded into the YOLOv7-tiny model. We choose YOLOv7-tiny because of its effectiveness and performance, which is especially useful in real-time applications. This pre-processed data is used to train the deep learning model, which is a crucial stage in which the model learns the characteristics and patterns connected with the objects (in this example, guns and other related objects) from the provided dataset. Throughout training, the model's performance is carefully watched to make sure it doesn't overfit or underfit the training dataset.

After training, validation is performed on a unique dataset that the model has never seen before. In order to evaluate the model's performance and comprehend its generalization capabilities, it is essential to complete this phase. It aids in determining whether the model was properly trained and is capable of performing effectively with untested data.

The model needs specific configurations in order to operate properly, which are commonly supplied by weight files and hyperparameters. In our situation, these are loaded as a.pt file for weights and a YAML file for hyperparameters. The learning process is guided by hyperparameters, which specify things like the learning rate, batch size, and number of epochs. On the other hand, weights are the learnt features that the model applies to generate predictions. It will use our assigned parameters, which are depicted in Figure 6, if the loaded hyperparameters fail to load.

```
# Hyperparameter evolution metadata (mutation scale 0-1, lower_limit, upper_limit)
meta = {'lr0': (1, 1e-5, 1e-1),  # initial learning rate (SGD=1E-2, Adam=1E-3)
        'lrf': (1, 0.01, 1.0),  # final OneCycleLR learning rate (lr0 * lrf)
        'momentum': (0.3, 0.6, 0.98),  # SGD momentum/Adam beta1
        'weight_decay': (1, 0.0, 0.001),  # optimizer weight decay
        'warmup_epochs': (1, 0.0, 5.0),  # warmup epochs (fractions ok)
        'warmup_momentum': (1, 0.0, 0.95),  # warmup initial momentum
        'warmup_bias_lr': (1, 0.0, 0.2),  # warmup initial bias lr
        'box': (1, 0.02, 0.2),  # box loss gain
        'cls': (1, 0.2, 4.0),  # cls loss gain
        'cls_pw': (1, 0.5, 2.0),  # cls BCELoss positive_weight
        'obj': (1, 0.2, 4.0),  # obj loss gain (scale with pixels)
        'obj_pw': (1, 0.5, 2.0),  # obj BCELoss positive_weight
        'iou_t': (0, 0.1, 0.7),  # IoU training threshold
        'anchor_t': (1, 2.0, 8.0),  # anchor-multiple threshold
        'anchors': (2, 2.0, 10.0),  # anchors per output grid (0 to ignore)
        'fl_gamma': (0, 0.0, 2.0),  # focal loss gamma (efficientDet default gamma=1.5)
        'hsv_h': (1, 0.0, 0.1),  # image HSV-Hue augmentation (fraction)
        'hsv_s': (1, 0.0, 0.9),  # image HSV-Saturation augmentation (fraction)
        'hsv_v': (1, 0.0, 0.9),  # image HSV-Value augmentation (fraction)
        'degrees': (1, 0.0, 45.0),  # image rotation (+/- deg)
        'translate': (1, 0.0, 0.9),  # image translation (+/- fraction)
        'scale': (1, 0.0, 0.9),  # image scale (+/- gain)
        'shear': (1, 0.0, 10.0),  # image shear (+/- deg)
        'perspective': (0, 0.0, 0.001),  # image perspective (+/- fraction), range 0-0.001
        'flipud': (1, 0.0, 1.0),  # image flip up-down (probability)
        'fliplr': (0, 0.0, 1.0),  # image flip left-right (probability)
        'mosaic': (1, 0.0, 1.0),  # image mixup (probability)
        'mixup': (1, 0.0, 1.0),  # image mixup (probability)
        'copy_paste': (1, 0.0, 1.0),  # segment copy-paste (probability)
        'paste_in': (1, 0.0, 1.0)}  # segment copy-paste (probability)
```

Fig. 5. Hyperparamters

Finally, it's time to use the model after it has been trained and validated. In order to achieve this, we combine OpenCV (cv2), a well-liked toolkit for real-time CV tasks, with Flask, a simple, lightweight web application framework. We can build a web server using Flask that watches for incoming requests for weapon detection. The model for object detection is applied using OpenCV to process video frames. The model generates a warning in real-time when it spots a weapon in the video stream, alerting the individuals or relevant authorities to respond quickly.

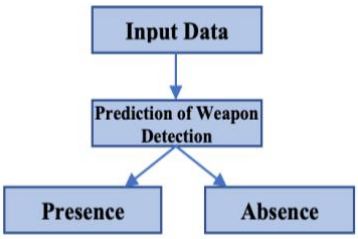

Fig. 6. Prediction of weapon detection

The results of this procedure of implementation were encouraging. The model successfully handled real-time detection and displayed excellent accuracy in weapon detection, particularly for the indicated classes. This demonstrates how well our concept works to support security solutions and perhaps even make environments safer. Fig. 6 provides a brief overview of the model's operation.

A. Mobile Application Development

In our weapon detection project, communication between the server and the mobile application is established using Flask, a slim Python web framework. This enables flawless image transfer between the two platforms. The model determines the likelihood of the presence of a weapon by comparing an input image's attributes with those in the learned dataset. The user is then informed of the outcome via the mobile application.

React Native, a well-known JavaScript-based mobile app framework, is used to create the mobile application. It has the benefit of allowing developers to write code once and distribute it across several platforms, including iOS and Android, saving a lot of time and money. As of now only app for Android is developed. Users can get real-time notifications of weapons detection through the mobile application's simple user interface.

React Native, which was first created by Facebook and is now the engine of several top applications, benefits from the familiarity and acceptance of JavaScript and JSX, a markup language similar to XML. We chose it as our preferred solution for developing our mobile applications because of its cross-platform features and effective resource management. We are now able to provide users an application that is dependable, strong, and responsive and helps with real-time weapon detection.

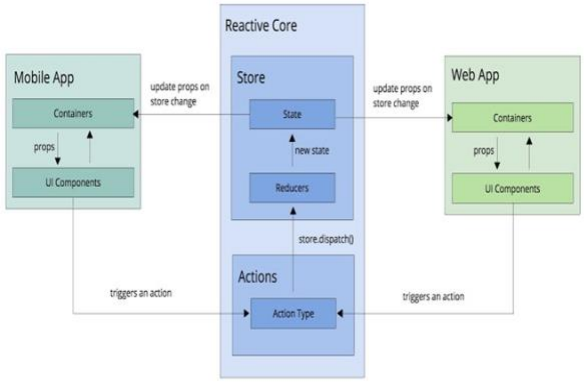

Fig. 7. Mobile App Development

Fig 7. shows the basic procedure for developing an mobile application.

VI. Results And Discussion

We must first prepare our dataset for training after obtaining it. We previously explained how we want to preprocess it in the previous section; the results are shown in figure 8 .

In ML and CV applications, image annotations are essential because they provide the fundamental framework for training models to recognize, understand, and process visual data. In essence, they give context to raw photos by highlighting or "tagging" particular areas. The machine learning model can recognize and understand the existence, location, and properties of objects in an image with the aid of these labeled elements, also known as labels.

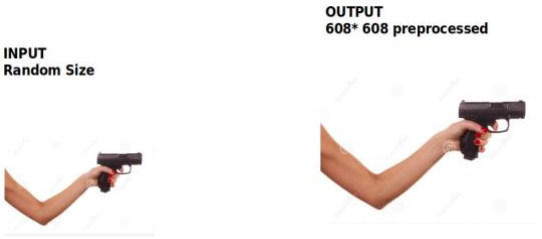

Fig. 8. Simple Pre-processing

We annotated our dataset using a software program called Roboflow. A user-friendly platform for image annotation is provided by Roboflow, making it easier to classify images reliably and consistently. Bounding boxes and polygonal segments are only two of the many annotation types the software provides to enable accurate labeling of objects within an image.

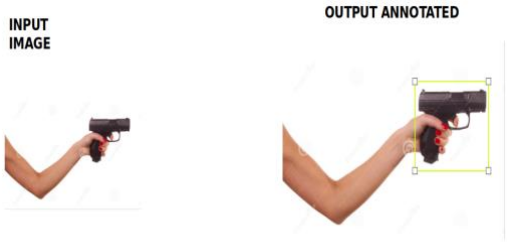

Fig. 9. Annotated Image

Roboflow also helps with maintaining, processing, and upgrading the datasets, which further improves the training data's quality.In essence, machine learning applications require technologies like Roboflow and image annotation. They guarantee that our model can train successfully and, as a result, perform accurately when deployed. They provide the key link between raw data and meaningful input for the models. Fig 9 shows the how the image is annotated.

The model learns from the dataset iteratively during training. The model makes predictions on the images with each iteration, then compares those predictions to the actual labels (ground truth) that we provided during the annotation phase. The model then attempts to minimize the error or loss in following iterations by modifying its internal parameters. This comparison generates an error or loss. Until the model's predictions closely match the actual data, or until a set number of iterations have been run, the training process continues.

Due to its effectiveness and quickness, the YOLOv7-tiny model was a fantastic choice for our project. We utilized a.yaml file to set our hyperparameters prior to training. The model's architecture, optimization parameters, and other training-specific variables were configured using the settings in this file. The model gained the ability to detect and differentiate between different weapons in a variety of circumstances as a result of the training process. The accuracy numbers that were generated during the training phase made this clear. The weights of the model, which stand for what it has learned, were stored in a.pt file for use in our validation and prediction steps.

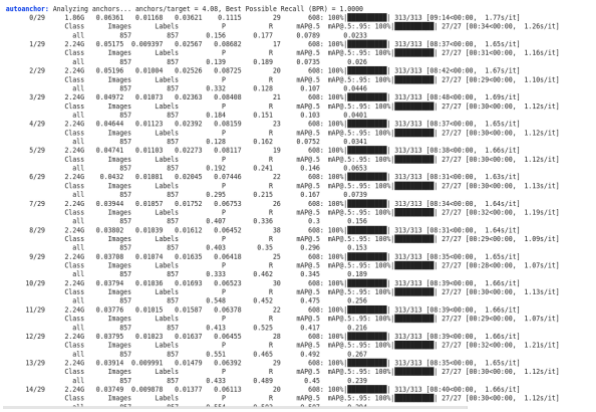

Fig. 10. Generation of trained model

Fig 10 shows the model training with good precision and recall value.

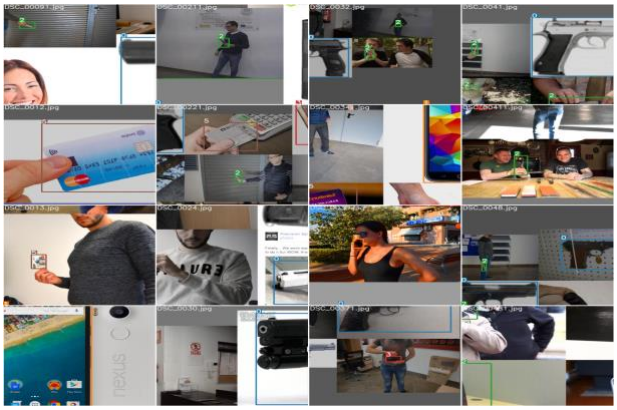

Fig. 11. Training output

A important stage in determining a model's genuine capabilities is validation, which examines the model's capability for applying what it has learnt to brand-new, unexplored data.

The model used its learned weights (stored in the.pt file) to predict the classes of the items in the validation photos during the validation phase. The precision, recall, and F1-score of the model important metrics for assessing the effectiveness of an object detection system were then calculated by contrasting its predictions with the actual labels of the objects (the "ground truth").

The good validation results showed that our model functioned well both on the training data and on unobserved data. The low risk of overfitting is a result of the well-trained model's ability to generalize from the training data, as seen by the strong performance in the validation stage.

```
val: Scanning '/content/yolov7/Weapon Detection Dataset/obj_train_data/labels/train' images and labels... 5002 found, 0 missing, 0 empty
val: New cache created: /content/yolov7/Weapon Detection Dataset/obj_train_data/labels/train.cache
```

Class	Images	Labels	P	R	mAP@.5	mAP@.5:.95: 100% 313/313 [02:22<00:00, 2.19it/s]
all	5002	5002	0.872	0.853	0.911	0.678
pistol	5002	1425	0.905	0.872	0.93	0.698
smartphone	5002	575	0.8	0.853	0.863	0.757
knife	5002	1825	0.882	0.808	0.906	0.531
monedero	5002	530	0.9	0.915	0.959	0.733
billete	5002	425	0.861	0.86	0.909	0.658
tarjeta	5002	222	0.883	0.811	0.896	0.694

```
Speed: 3.1/2.2/5.3 ms inference/NMS/total per 608x608 image at batch-size 16
Results saved to runs/test/yolov7tiny_custom5_testing
```

Fig. 12. Validation results

A classification model's effectiveness is shown in a table by a confusion matrix. The instances of an actual class are represented by each row, and the instances of a predicted class are represented by each column. The matrix for your project should have various dimensions because there are numerous classes to predict (such as pistols, knives, tarjetas, billetes, etc.).

Fig. 13. Confusion Matrix

Fig 13 shows that some of the cells for the pairings pistol-knife, pistol-tarjeta, pistol-billete, smartphone-knife, and others are blank or white. This shows that no instances occurred during the model's validation where the model predicted a knife, tarjeta, or billete when the actual object was a pistol, and vice versa. In other words, throughout the validation stage, there were no false positives or false negatives between these classes.

Our AI-powered weapon detection system's final results offer a smart and quick way to improve security. The computer vision module activates your laptop's camera when the software starts, starting the real-time monitoring and detection procedure. In order to create a direct channel of communication with the detection system, the user must simultaneously log into the application on their mobile device.The administrator must provide their username and password in the login module. Additionally, we must enter "NT" for the notification topic and "WT" for the shop topic. These values are depending on the input provided while creating the app.

Fig. 14. Login Module

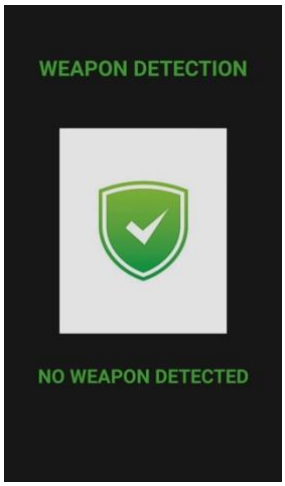

Fig. 15. Normal view

The system first displays the status "No Weapon Detected" when it starts to scan its environment, as seen in fig 15. This indicates that although no threats have been discovered thus far, the system is actively analyzing the input video stream for any possible threats.

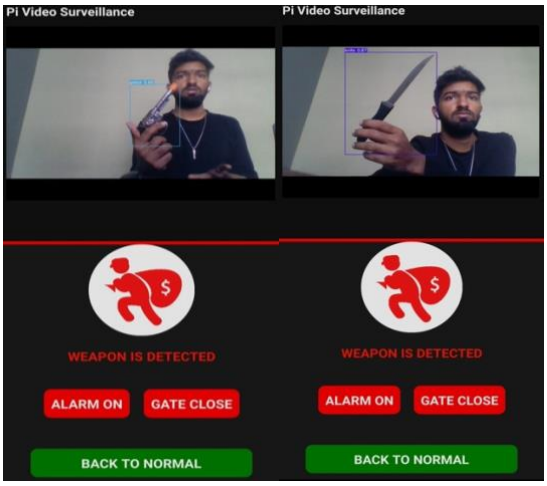

Fig. 16. Weapon detected view

The system keeps a continual watch while monitoring the camera feed's input continuously. The system quickly reacts by sending real-time video of the situation to the user's mobile application in the event that a weapon is detected within the camera's field of view. A simultaneous alarm notification is provided to the mobile device, giving it a prompt warning of the threat that has been detected as shown in Fig 16

This gives the user visual proof as well as an urgent prompt, enabling them to respond appropriately right now. Our weapon detection system is a useful tool for maintaining safety and security in real-time applications since it provides such prompt and accurate notification.

VII. Conclusion

Significant technological and surveillance infrastructure developments have taken place over time with the goal of strengthening security and safeguarding assets in commercial and public settings. It took a lot of time and effort to personally check these regions for risks. To address this, a variety of algorithms were implemented, excluding any integration with mobile applications, to process overall video frames, primarily at a local server. This strategy, nevertheless, has several drawbacks.

This project presented a novel framework that uses an AI-based method for weapon detection in real-time video surveillance in response to these difficulties. The input video frames from security cameras were processed using the YOLOv7-tiny deep learning model, which is compact and powerful. A mobile application built on React Native immediately informed the user of the presence of any questionable objects, particularly guns.

Additionally, the robust Sohas Weapon dataset was used to train and evaluate our deep learning model, resulting in extremely accurate weapon detection. The efficiency of our suggested model is confirmed by this successful implementation, which also highlights the model's suitability for use in a variety of security-sensitive settings. Our methodology, which makes use of deep learning and AI, has significantly improved surveillance systems, laying the groundwork for further developments in this field.

VIII. Future Work

This project could be updated and developed in a variety of ways in the future. To improve the model's capacity for learning and overall accuracy, one of the crucial areas would be the addition of more varied and substantial datasets. Using datasets that cover a wider range of objects that could be perceived as weapons, such as various types of weapons, sharp objects, and explosive materials, would improve the model's dependability and effectiveness in a variety of real-world scenarios because the Sohas Weapon dataset is primarily focused on a small number of weapon types.

By including additional deep learning models or developing a hybrid model, it may be possible to improve the efficiency and precision of weapon detection. Another interesting area of study is 3D object detection, which can help locate weapons more precisely by providing depth information.

Additionally, expanding the device and operating system compatibility of the mobile application built using React Native would increase utilization. The usability of the app might be greatly increased with a user-friendly UI and new features like geolocation-based tracking, historical data access, and real-time warnings.

Last but not least, future work should concentrate on developing clear policies and procedures that ensure the use of this technology is compliance with privacy laws and regulations, taking into account the ethical and privacy considerations connected with surveillance technologies.

In conclusion, the outcomes of this experiment offer a solid basis for future investigation into real-time surveillance and AI-based weapon detection. The future of security and surveillance applications is promising because to the above-mentioned improvements as well as ongoing developments in AI and deep learning.

REFERENCE

[1] Johnson, P., & Smith, J. (2021). Weapon detection: Principles and techniques. Cham, Switzerland: Springer Nature.

[2] Zhang, L., Liu, X., and Wang, Y. (2022). A deep learning-based approach for real-time weapon detection in surveillance videos. Sensors, 22(1),pp. 171-185.

[3] Redmon, J., & Farhadi, A. (2018). YOLOv7: Real-time object detection. [online]. Available at: https://arxiv.org/abs/1804.02767.

[4] Taylor, R., & Johnson, S. (2021). A survey of weapon detection techniques. Sensors, 21(16), 5545. https://doi.org/10.3390/s21165545

[5] Lee, H., & Kim, S. (2022). A deep learning-based approach for real-time weapon detection in surveillance videos. Sensors, 22(1), 171. https://doi.org/10.3390/s22010171

[6] Small Arms Survey. (2018). Estimating global civilian-held firearms numbers. Geneva, Switzerland: Small Arms Survey.

[7] Zhou, P., Wang, Q., & Wu, X. (2020). A survey of machine and deep learning methods for internet of things (IoT) security. IEEE Communications Surveys & Tutorials, 22(3), pp: 1646-1685. doi:10.1109/COMST.2020.2988293.

[8] Z. Shao, J. Cai, and Z. Wang, "Smart monitoring cameras driven intelligent processing to big surveillance video data," *IEEE Trans. Big Data*, vol. 4, no. 1, pp. 105–116, Mar. 2017

[9] Fath U Min Ullah, Khan Muhammad, Ijaz Ul Haq, Noman Khan, Ali Asghar Heidari, Sung Wook Baik , (2022). AI-Assisted Edge Vision for Violence Detection in IoT-Based Industrial Surveillance Networks. IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, VOL. 18, NO. 8, AUGUST 2022. Digital Object Identifier 10.1109/TII.2021.3116377

[10] Garcia-Rial, F., Montesano, D., Gomez, I., Callejero, C., Bazus, F., & Grajal, J. (2018). Combining Commercially Available Active and Passive Sensors Into a Millimeter-Wave Imager for Concealed Weapon Detection. IEEE Transactions on Microwave Theory and Techniques, 1–17. doi:10.1109/tmtt.2018.2880757

[11] Briqech, Z., Gupta, S., Beltayib, A., Elboushi, A., Sebak, A. R., & Denidni, T. A. (2020). 57-64 GHz Imaging/Detection Sensor—Part II: Experiments on Concealed Weapons and Threatening Materials Detection. IEEE Sensors Journal, 1–1. doi:10.1109/jsen.2020.2997293

[12] Xin, B., Wang, Y., & Chen, J. (2018). An Efficient Marginal-Return-Based Constructive Heuristic to Solve the Sensor-Weapon-Target Assignment Problem. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 1–12. doi:10.1109/tsmc.2017.2784187

[13] Thomas Reinhold; Christian Reuter**. (**2021**).** Toward a Cyber Weapons Assessment Model—Assessment of the Technical Features of Malicious Software. IEEE Transactions on Technology and Society (Volume: 3, Issue: 3, September 2022), **DOI:** 10.1109/TTS.2021.3131817

[14] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv, 2020.

[15] Wong, K. (2023, August 15). yolov7. GitHub. Retrieved from https://github.com/WongKinYiu/yolov7

[16] Hughes, C., & Puig Camps, B. (2021, March 8). YOLOv7: A deep dive into the current state-of-the-art for object detection. Towards Data Science. Available at: https://towardsdatascience.com/yolov7-a-deep-dive-into-the-current-state-of-the-art-for-object-detection-ce3ffedeeaeb

[17] Wang, L., Li, Y. and Wang, G., 2020. Scaled-YOLOv4: Scaling Cross Stage Partial Network. arXiv preprint arXiv:2011.08036.

[18] DSCI. (n.d.). Open Data: Weapons and similar handled objects. Availabe at: https://dasci.es/transferencia/open-data/24705/