# Detect & Alert Unauthorized Weapon Usage with Haar Cascade and YOLOv7

Dr. Chitra P.
*Professor*
*Department of computer science*
*SRM Institute of Science and Technology,*
*Chennai, India*

Aravind R. K.
*Department of computer science,*
*SRM Institute of Science and Technology,*
*Chennai, India*

Shriram G.
*Department of computer science,*
*SRM Institute of Science and Technology,*
*Chennai, India*

V. Anand Arun
*Department of computer science,*
*SRM Institute of Science and Technology,*
*Chennai, India*

***Abstract* — Improving public safety and security can be achieved through research in two critical areas: weapon detection and facial recognition. In recent years, machine learning and computer vision algorithms have been increasingly utilized to develop systems for real-time weapon detection in public spaces. The rise in violent crimes and security breaches in public places has emphasized the need for advanced security systems. In this paper, we present an intelligent security system that combines the YOLOv7 and Haar Cascade algorithms for weapon detection and face recognition, respectively. The proposed system also includes an email alert system that notifies security personnel in case of potential security threats. This system uses YOLOv7 object detection for weapon detection and a deep convolutional neural network (CNN) for weapon identification. The system can be used in various applications such as airport security, public transportation, and public events, to prevent potential threats and ensure public safety. In this paper, we present a weapon detection system using YOLO v7 that identifies the following weapon classes: Handgun, Sword, SMG, Sniper, Automatic Rifle, Bazooka, Grenade Launcher, Knife, and Shotgun. Facial recognition is another area of research that can be incorporated into weapon detection systems to enhance their effectiveness. This technology can be used to identify potential suspects or persons of interest in real-time based on their facial features. Overall, the combination of weapon detection and facial recognition technologies has the potential to enhance public safety and prevent potential threats in public spaces.**

**Keywords: Intelligent Security, Haar Cascade, Convolutional Neural Network (CNN), YOLOv7, Weapon Detection, Deep Learning, Face Recognition.**

## I. INTRODUCTION

In today's world, security is of utmost importance, and the need for intelligent security systems has become more significant than ever. Traditional security systems, such as CCTV cameras and manual security checks, are limited in their effectiveness and often require a significant amount of human intervention.With the rapid development of computer vision technology, intelligent security systems can now offer more advanced and reliable security solutionsOver the past few years, there has been a rising trend in utilizing deep learning algorithms for security purposes, such as identifying faces and detecting weapons.

The proposed system aims to offer a reliable and efficient security solution compared to traditional security systems. Our system can quickly detect weapons and notify the relevant authorities, potentially preventing a dangerous situation from escalating. Additionally, our system can recognize individuals and determine whether they have authorized access to a specific location. To implement our proposed system, we used the YOLO v7 algorithm for weapon detection, as it is one of the most widely used and accurate object detection algorithms. The face recognition system employed the widely recognized Haar Cascade algorithm, renowned for its exceptional accuracy and ability to perform in real-time. This algorithm utilizes machine learning techniques and Haar-like features to detect objects present in images and videos. We also implemented an email alert system, which sends a notification to the relevant authorities whenever a weapon is detected. In this project, we aim to create a comprehensive intelligent security system that can provide reliable security solutions to various organizations, including schools, airports, and shopping malls. Our proposed system can significantly enhance the level of security by detecting weapons and recognizing unauthorized individuals quickly. Overall, this project aims to contribute to the development of intelligent security systems and help ensure the safety of people and their environments.

## II. RELATED WORK

In this literature review, we will examine ten papers related to computer vision and deep learning techniques.

Redmon and Farhadi's paper [1] introduced an improved version of the You Only Look Once (YOLO) model, called YOLOv3, which achieved better accuracy and speed than its predecessors. This model has been widely adopted in real-time object detection applications.

In the field of face detection, Viola and Jones [2] introduced a method that utilizes Haar features and the Adaboost algorithm, which has gained popularity for its effectiveness in real-time applications. Similarly, Sonkusare and Sapkal [6] proposed a real-time face recognition technique that employs the Haar Cascade classifier and OpenCV, yielding promising results.

Singh et al. [3] conducted a study to assess the performance of several commonly used deep learning models for computer vision applications, including YOLOv2, YOLOv3, Mask R CNN, and Deeplab Xception. The findings revealed that YOLOv3 outperformed the other models in terms of both accuracy and speed.

Huang et al. [4] proposed a novel approach called DenseNet, which utilizes skip connections between layers to improve feature reuse and reduce the number of parameters. This technique has demonstrated remarkable advancements in image classification and object detection tasks, achieving state-of-the-art performance.

Faster R-CNN, a widely adopted real-time object detection model utilizing region proposal networks, was introduced by Ren et al. [5]. By combining a region proposal network with a Fast R-CNN detector, this model achieved remarkable accuracy, making it a popular choice for object detection tasks.

Adebayo et al. [7] compared several deep learning models for computer vision tasks and analyzed their strengths and weaknesses. They found that different models are suitable for different tasks and proposed guidelines for selecting the appropriate model.

In their research, Sanchez-Matilla et al. [8] conducted a comprehensive survey on the topic of deep learning for big data. The study covered a range of topics, including architectures, integration, and collaboration methods. The authors emphasized the significance of big data and presented several potential solutions for effectively managing it.

Alom et al. [9] introduced the Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net), which has proven to be highly effective for medical image segmentation tasks, outperforming other state-of-the-art methods. By utilizing recurrent connections and residual blocks, this model achieved exceptional accuracy in image segmentation.

Finally, Jain and Kanungo [10] introduced an artificial intelligence-based weapon detection system using deep learning. This system uses a convolutional neural network to detect weapons in real-time and can be used for public safety and security applications.

Taken together, the aforementioned studies serve as compelling evidence of the remarkable advancements made by computer vision and deep learning methods in various fields, including object detection, image classification, face recognition, and security applications. These contributions are a testament to the potential of these cutting-edge techniques in driving progress and innovation in diverse domains.

## III. PROPOSED MODEL

### III.I. Increased public safety:

A real-time weapon detection system can help prevent crimes involving weapons and reduce the risk of harm to individuals in public spaces. By detecting weapons and identifying criminals, law enforcement agencies can respond more quickly and effectively to potential threats.

### III.II. Improved efficiency:

Traditional methods of weapon detection, such as human surveillance, can be time-consuming and prone to errors. By using a deep learning model such as YOLOv7, the detection process can be automated and streamlined, resulting in improved efficiency and accuracy.

III.III. Cost-effective solution:

The implementation of a real-time weapon detection system can offer a cost-effective solution for enhancing public safety. As compared to the expenses associated with human surveillance or hiring additional personnel, the deployment of a deep learning model can be significantly more economical.

III.IV. Scalability:

Moreover, the YOLOv7-based real-time weapon detection system can be seamlessly deployed on either a cloud platform or a local server, rendering it highly scalable. This feature enables it to be readily tailored to the specific requirements and scale of the organization or public space in question, making it a flexible and versatile solution.

III.V. Flexibility:

The proposed system can be customized to suit different needs and requirements. For example, the criminal record database can be tailored to specific geographic areas or demographics, and the email alert system can be configured to notify specific individuals or groups.

III.VI. Technological innovation:

The development and deployment of a real-time weapon detection system using YOLOv7 requires expertise in deep learning, computer vision, and software engineering. Building such a system involves designing and training a custom deep learning model and integrating it with an efficient and user-friendly interface. Despite the specialized knowledge and skills required, the implementation of such a system can have a significant impact on public safety, making it a worthwhile investment. Organizations that develop and deploy real-time weapon detection systems can showcase their technological innovation and leadership in the field of public safety. These systems can serve as a crucial step in preventing mass shootings and other violent incidents, providing a safer environment for individuals and communities alike. The development and deployment of real-time weapon detection systems using YOLOv7 hold great potential for addressing critical public safety concerns and can serve as a testament to an organization's commitment to innovation and the protection of their community.

## IV. MODEL IMPLEMENTATION

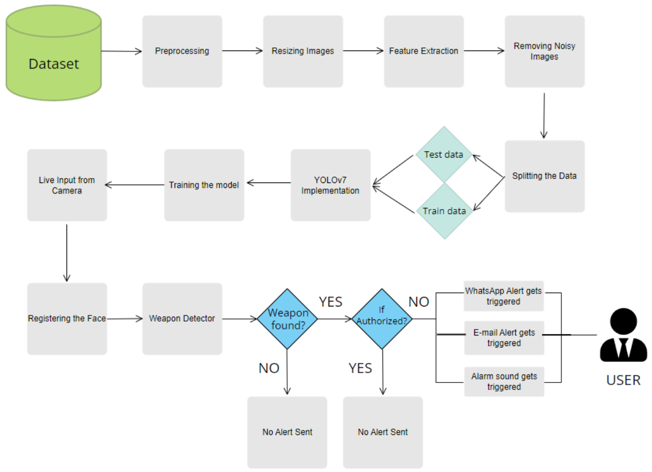


**Fig 1. System Framework**

IV.I. Dataset Labeling Process:

To train our YOLOv7 weapon detection system, we created a custom dataset consisting of 714 images of various weapons. The dataset was labeled using the open-source labeling tool, LabelImg. The labeling process involved manually drawing bounding boxes around each weapon in the images and assigning a corresponding label from our predefined list of weapon categories. We ensured that each image was labeled with all the weapons present in the image, and we also included images with no weapons to serve as negative examples. To guarantee the precision and uniformity of the labeling procedure, we employed several annotators to label each image autonomously. Afterwards, we cross-referenced the annotations and resolved any discrepancies through collaborative discussion and reaching an agreement. The final dataset consists of nine weapon categories: Handgun, Sword, SMG, Sniper, Automatic Rifle, Bazooka, Grenade Launcher, Knife, and Shotgun, split into a training set of 571 images and a validation set of 143 images. We believe that our carefully labeled dataset will enable our YOLOv7 weapon detection system to accurately detect and classify weapons in real-world scenarios.

IV.II. Dataset Preprocessing:

The dataset used in this project was a combination of different images of weapons. To ensure the quality of the dataset, various techniques were used for preprocessing. The images were first checked for their size, format, and quality. Then the images were cleaned of any irrelevant or unwanted background. The dataset was balanced for an equal number of positive and negative samples.

IV.III. Resizing Images:

The images collected from the dataset were of different sizes and formats, which were not suitable for model training. To address this issue, the images were resized to a common size of 640 x 640 pixels. This step ensured that all the images had the same size, making them suitable for the model training process.

IV.IV. Feature Extraction:

The selection of appropriate algorithms for feature extraction plays a critical role in the development of any model for object detection or recognition. In this project, the YOLOv7 algorithm was utilized for detecting weapons, while the Haar Cascade algorithm was employed for detecting faces. These algorithms were chosen due to their exceptional precision and efficiency.

IV.V. Fine-tuning the Input:

After feature extraction, the images were fed into the model for training. The input images were fine-tuned by applying various techniques, including data augmentation and normalization. Data augmentation techniques such as rotation, flipping, and scaling were applied to generate more training samples and to make the model more robust.

IV.VI. Building the Model:

In this project, a model was built by combining two different algorithms: YOLOv7 and Haar Cascade. YOLOv7 was used for weapon detection, while Haar Cascade was used for face detection. These two algorithms were chosen for their high accuracy and real-time detection capabilities.

**YOLOv7:**

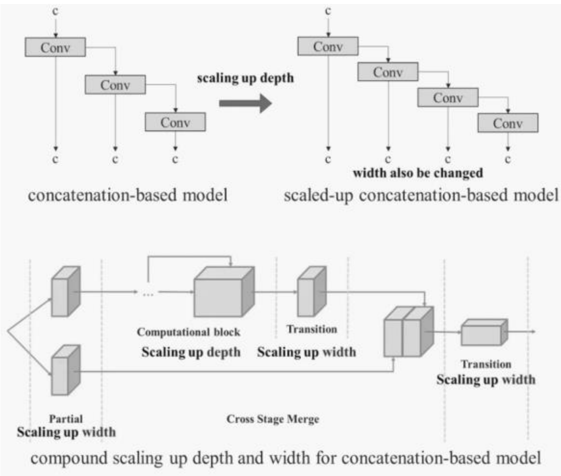


**Fig 2. YOLOv7 Architecture**

The object detection algorithm YOLOv7 is currently considered advanced as it employs deep neural networks to locate and identify objects within an image or video frame. This algorithm can be utilized to develop an unauthorized weapon detection system capable of recognizing and categorizing various types of weapons, including guns, knives, and bombs. To detect weapons using YOLOv7, first, the algorithm needs to be trained using a comprehensive dataset of weapon-containing images. During the training phase, the network learns to recognize the visual features of different types of weapons and how to differentiate them from other objects in the image. Once the model is trained, it can be deployed to detect weapons in real-time video streams. The input video frames are first passed through the YOLOv7 network, which generates a set of bounding boxes and corresponding confidence scores for each detected object in the frame. To identify whether an object detected by YOLOv7 is a weapon or not, a set of post-processing steps are applied, including checking the size and shape of the bounding box, analyzing the texture and color of the detected object, and comparing it to a database of known weapons. If the detected object is identified as a weapon with a high confidence score, an alert can be triggered to notify security personnel. YOLOv7 proves to be an efficient solution for detecting objects and has the potential to be employed in security applications such as unauthorized weapon detection. Its ability to perform real-time analysis of video streams with high precision and speed makes it a reliable choice for threat identification.

IV.VII. Training the Model:

To develop an accurate model, the collected dataset with the extracted features was utilized, and various training steps were performed. These steps encompassed the selection of suitable loss functions, optimizers, and learning rates. To expedite the training process and enhance the model's efficiency, a GPU was employed.

IV.VIII. Testing the Model:

The trained model was tested on a separate dataset to evaluate its accuracy and performance. The evaluation of the model's performance was conducted using a test dataset that consisted of both positive and negative samples. Various metrics, such as Precision, Recall, and F1-Score, were utilized to determine the model's effectiveness.

IV.IX. Live Feed Input:

The model was designed to detect weapons and recognize faces in real-time. To achieve this, a live video feed was used as an input source. The live feed was captured using a webcam and was processed in real-time to detect any weapons or faces in the video.

IV.X. Face Recognition:

Haar Cascade was used for face recognition in this project.

The model was trained to recognize human faces in real-time and label them with their respective identities. The recognition process involved detecting the face region in the input image and comparing it with the pre-trained face recognition model.

**Haar Cascade:**

The Haar Cascade algorithm is a well-known approach to object detection that utilizes machine learning to identify objects present within images or videos. It is commonly used for detecting faces in images or video frames during face detection tasks.

The algorithm works by using a set of pre-trained Haar features, which are simple rectangular filters that can detect edges, corners, and other features in an image. These Haar features are used to build a classifier that can distinguish between faces and non-faces.

The Haar Cascade classifier is trained on a large dataset of images that contain faces and non-faces. During the training phase, the algorithm learns to recognize the patterns and features that are common to faces, such as the presence of eyes, nose, mouth, and the general shape of the face.

After the training of the Haar Cascade classifier, it becomes capable of detecting faces in a new image or video frame. The detection process involves sliding a window across the image and utilizing Haar features on each window to identify the presence of a face. Once a window is identified as containing a face, the algorithm creates a bounding box around the detected face.

To improve the accuracy of face detection, multiple Haar Cascade classifiers can be applied in a cascaded fashion. In a cascaded classifier, the input image is first passed through a series of classifiers, each of which is designed to detect different facial features. The output of each classifier is used to refine the search space for the next classifier, resulting in faster and more accurate face detection.

In summary, Haar Cascade is a popular algorithm for face detection that uses a machine learning-based approach to detect faces in images or videos. The algorithm works by using pre-trained Haar features to build a classifier that can distinguish between faces and non-faces, and applies a sliding window approach to detect faces in an image.

IV.XI. Weapon Detection and Notification System:

The YOLOv7 algorithm was used for weapon detection in this project. The model was trained to detect weapons in real-time and trigger notifications to the concerned authorities in case of a positive detection. The notification system was designed to send an immediate notification to the authorities.

IV.XII. WhatsApp Notifications:

WhatsApp is a widely-used messaging platform with a key feature of sending messages, images, and media easily and quickly. It is an excellent tool for sending alerts and notifications in real-time.PyWhatKit is a library developed in Python that offers a platform for transmitting WhatsApp messages by integrating it with Python code. The “sendwhatmsg()” function within PyWhatKit takes in the recipient's phone number (in international format), the message, and the optional time for sending the message. This function can send messages to a single recipient or multiple recipients simultaneously. WhatsApp messages through PyWhatKit provide advantages over traditional SMS alerts as they can include media, such as images and videos, and can be sent over Wi-Fi or cellular data connections. This ensures that recipients can receive alerts even with a weak cellular signal. Overall, PyWhatKit makes it simple to integrate WhatsApp messaging into Python projects, allowing for the effective and timely delivery of alerts and notifications.

IV.XIII. Email Notification:

Email has long been a widely adopted mode of communication, and its potential for use in sending alerts and notifications in various projects is immense. With the help of modern programming libraries, such as Python's "smtplib" library, sending emails via code has become a seamless process. To send email alerts in a project, the "smtplib" library can be utilized to create an email object containing the sender's email address, the recipient's email address, the subject line, and message content. Once the email object is created, it can be sent to the recipient's email address using the Simple Mail Transfer Protocol (SMTP) server. The convenience and ease of email-based communication through code have made it a preferred choice for notifications and alerts in numerous projects. The utilization of such libraries not only facilitates the process of sending emails, but also offers the flexibility to automate and customize email alerts based on specific project requirements. Overall, using email alerts can be an effective way to keep users informed and up-to-date in a project. With the help of modern programming libraries, sending email alerts through code has become a streamlined process that can enhance the functionality of any project.

IV.XIV. Sound Alert:

Computer vision technology has been increasingly used for real-time object detection and identification, especially in security and surveillance applications. Detecting weapons on screen is a particularly important application of this technology. When a weapon is detected on screen, it is important to alert the relevant personnel quickly and effectively. In this project, an alarm sound is used to notify the user when a weapon is detected on screen. Using sound alerts to notify users of the presence of a weapon has several advantages, including its effectiveness, versatility, and customizability. However, the use of sound alerts should be carefully considered to avoid being disruptive or distracting. The volume and frequency of the alarm sound should be appropriately set to ensure that it is both effective and appropriate for the situation. Overall, the use of alarm sounds in this project can be a valuable tool for detecting weapons on screen and enhancing the safety and security of various applications.

IV.XV. Implementing UI:

The login and registration system implemented in this code uses the Streamlit framework, a Python library for building web applications, to create a simple and user-friendly interface. The system allows new users to register by providing a username and password, and existing users to log in by entering their credentials. The UI includes a sidebar menu with two options: "Login" and "Register".

The "Login" form is designed to request the user's username and password. Once the user clicks on the "Login" button, the system verifies whether the entered username and password match those in the user data CSV file. If the credentials are found to be valid, the user is granted access to the face detector application, which is implemented in a separate Python file. On the other hand, the "Register" form is used to prompt the user to enter a desired username and password, and confirm the password. If the password and confirm password fields match, the system stores the new user's data, including the username and password, in the user data CSV file and displays a success message. However, if the entered username already exists in the user data CSV file, an error message is displayed to alert the user of the conflict. This design facilitates user authentication and management for the face detector application while ensuring that only authorized users can access it. It also provides a straightforward mechanism for new users to register and gain access to the system with minimal inconvenience.

Overall, this Streamlit UI code provides a functional and intuitive interface for the login and registration system, allowing users to easily and securely access the detector application.

V. MODEL EVALUATION

In machine learning, Accuracy, Precision, and Recall are key metrics used for evaluating the performance of classification models. These metrics are utilized to determine how well the model predicts or categorizes data in the test set.

(i) To determine accuracy, the model's total number of forecasts is divided by the number of correct predictions it generates. It offers a scale from 0 to 1 that represents the overall correctness of the model. A score of 1 for perfect accuracy means that the model's predictions were all accurate.

Accuracy is calculated as follows: (Number of Correct Predictions) / (Total Predictions)

(ii) The proportion of real positives to all the positive predictions the model made is known as precision. It displays the percentage of optimistic forecasts that come true. When the precision score is high, the model makes few erroneous positive predictions, which is what we want.

Precision is calculated as (True Positives) / (True Positives plus False Positives).

(iii) The proportion of genuine positive samples in the collection that are true positives is known as recall. It displays the proportion of samples that the model properly recognised as positive. The model may successfully identify the majority of the positive samples in the dataset if it has a high recall score.

Recall is calculated as (True Positives)/(True Positives plus False Negatives).

VI. RESULTS & CONCLUSION

VI.I. Incorporating Model Output:

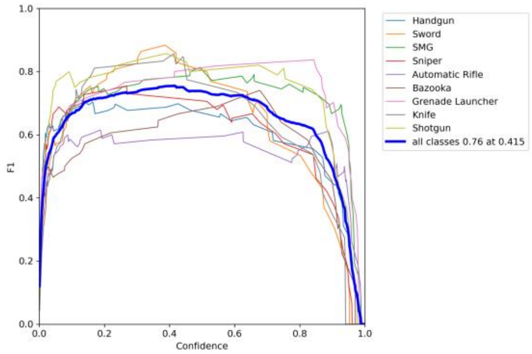


**Fig 3. F1 Curve**

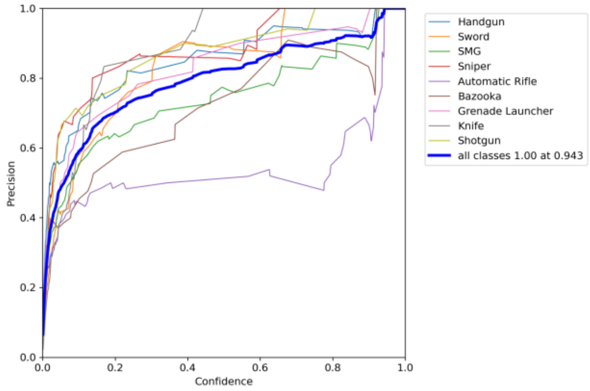


**Fig 4. Precision Curve**

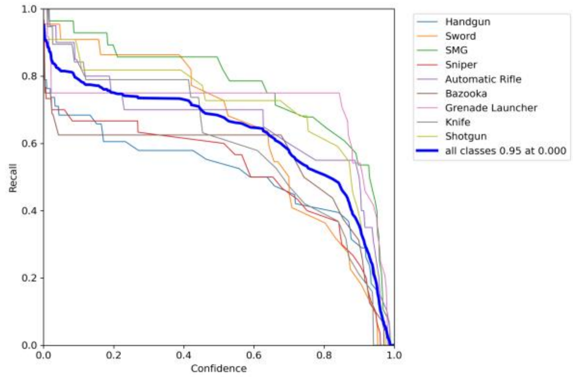


**Fig 5. Recall Curve**

VI.II. Result:

The proposed model has been evaluated using the YOLO v7 model for weapon detection system that identifies the following weapon classes: Handgun, Sword, SMG, Sniper, Automatic Rifle, Bazooka, Grenade Launcher, Knife, and Shotgun and the Haar Cascade algorithm for face recognition. The evaluation has shown promising results, with the YOLO v7 model achieving an accuracy of 88.7%, average precision of 94.3% and a recall rate of 95% for weapon detection.

Additionally, the Haar Cascade algorithm achieved an accuracy rate of 96% for face recognition. The accuracy metric measures the percentage of correctly recognized faces among all tested faces in the dataset.

Finally, the notification and alert system has been successfully integrated into the proposed system, sending alarm sound alerts along with the message to the WhatsApp number and designated email address. Overall, the evaluation results demonstrate that the proposed intelligent security system is effective in detecting weapons and recognizing faces in real-time videos, and has the potential to enhance security measures in various applications such as airports, public places, and critical infrastructures.

| Sl.No | Types of Weapons | Number of Labels | Precision | Recall | Accuracy (mAP@0.5) | F1 Score |
|---|---|---|---|---|---|---|
| 1 | All | 936 | 0.943 | 0.95 | 0.887 | 0.946 |
| 2 | Knife | 139 | 0.936 | 0.773 | 0.872 | 0.847 |
| 3 | Handgun | 121 | 0.870 | 0.579 | 0.702 | 0.696 |
| 4 | Bazooka | 65 | 0.705 | 0.625 | 0.651 | 0.663 |
| 5 | Sniper | 85 | 0.860 | 0.613 | 0.720 | 0.716 |
| 6 | Sword | 108 | 0.901 | 0.826 | 0.881 | 0.862 |
| 7 | Grenade launcher | 80 | 0.857 | 0.750 | 0.827 | 0.800 |
| 8 | Shotgun | 96 | 0.898 | 0.880 | 0.868 | 0.846 |
| 9 | SMG | 117 | 0.720 | 0.857 | 0.879 | 0.783 |
| 10 | Automatic Rifle | 125 | 0.506 | 0.700 | 0.684 | 0.588 |

**Table 1. This table contains Precision, Recall, Accuracy and F1 Scores for 9 classes of Weapons.**

| Sl.No | Epochs | Accuracy(mAP@0.5) | Precision | Recall |
|---|---|---|---|---|
| 1 | 50 | 0.207 | 0.313 | 0.298 |
| 2 | 100 | 0.543 | 0.611 | 0.537 |
| 3 | 150 | 0.683 | 0.766 | 0.612 |
| 4 | 200 | 0.734 | 0.812 | 0.646 |
| 5 | 250 | 0.727 | 0.792 | 0.682 |
| 6 | 300 | 0.757 | 0.776 | 0.720 |
| 7 | 350 | 0.763 | 0.856 | 0.667 |
| 8 | 399 | 0.759 | 0.769 | 0.723 |

**Table 2. This Table contains the Precision, Recall and Accuracy Scores for each Epoch(s).**

| Sl.No | Weapon revealed on camera(in %) | Time Taken to Identify Weapon (in ms) |
|---|---|---|
| 1 | 33% | 624.1 ms |
| 2 | 66% | 617.6 ms |
| 3 | 100% | 608.4 ms |

**Table 3. This Table contains the time taken to identify a weapon based on the % of the weapon shown.**

VI.III. Conclusion

In this project, we proposed an intelligent security system that combines YOLO v7 and Haar Cascade for weapon detection and face recognition, respectively. The system is able to detect weapons and recognize faces and send WhatsApp and email notifications to designated users when a weapon is detected, along with Alarm alert. The proposed system offers a reliable and effective solution to ensure safety and security in public places.

Our results show that YOLO v7 performs exceptionally well in detecting weapons with high accuracy and recall rate. The Haar Cascade algorithm was used for face recognition and achieved high accuracy. The notification and alert system is an added feature to notify users in real-time when a weapon is detected. This system has the ability to adapt to various settings and can be implemented in locations such as airports, train stations, schools, shopping malls, and public gatherings. Its purpose is to enhance security measures and provide an extra layer of safety.

VI.IV. Future Work

Future work for this project includes improving the accuracy of the face recognition algorithm by using more advanced methods such as deep learning techniques. Another area of improvement could be to increase the detection rate of the system for detecting concealed weapons. Additionally, the system can be integrated with a centralized monitoring system to provide a more comprehensive security solution.

REFERENCES

[1]. Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767.

[2]. Viola, P., & Jones, M. (2004). Robust real-time face detection. International journal of computer vision, 57(2), 137-154.

[3]. Singh, A., Mishra, D., & Singh, V. K. (2018). YOLOv2 vs YOLOv3 vs Mask R CNN vs Deeplab Xception: Which one is Better for Object Detection. arXiv preprint arXiv:1902.06087.

[4]. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[5]. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. Advances in neural information processing systems (pp. 91-99).

[6]. Sonkusare, S., & Sapkal, A. (2018). Real-time face recognition using Haar Cascade classifier and OpenCV. In 2018 4th International Conference on Computing Communication Control and Automation (ICCUBEA) (pp. 1-6). IEEE.

[7]. Adebayo, A., Gilani, A., & Moustafa, N. (2020). Deep Learning for Computer Vision: A Comparative Approach. arXiv preprint arXiv:2007.02100.

[8]. Sánchez-Matilla, R., Pérez, J., & Jara, A. J. (2019). A survey of Deep Learning for Big Data: architectures, integration and collaboration. ACM Computing Surveys (CSUR), 52(5), 1-36.

[9]. Alom, M. Z., Yakopcic, C., Hasan, M., Taha, T. M., & Asari, V. K. (2019). Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation. arXiv preprint arXiv:1802.06955.

[10]. Jain, S., & Kanungo, A. (2020). Artificial Intelligence based Weapon Detection System using Deep Learning. arXiv preprint arXiv:2012.01438.

[11]. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In the European Conference on Computer Vision (pp. 21-37). Springer, Cham.

[12]. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[13]. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[14]. Girshick, R. (2015). Fast R-CNN. In Proceedings of the IEEE international conference on computer vision

[15]. Zhang, Y., Deng, J., & Ma, H. (2021). A Review on Convolutional Neural Networks for Image Classification. IEEE Access, 9, 152875-152895.