IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

VisionX: An End-to-End Object Detection Platform Using YOLO and React

[1]Dulam Rohit Sai Anjan, [2]Puli Bhanu Kiran, [3]Kodi Karthikaya,[4]Velaga Rudra Bharath,

[1]Final Year B.Tech Student, [2]Final Year B.Tech Student, [3]Final Year B.Tech Student, [4]Final Year B.Tech Student,

[1,2,3,4]Department of CSE-AIML,

[1,2,3,4]Aditya College of Engineering & Technology(A), Surampalem, Andhra Pradesh, India

Abstract: Object detection is an important issue in computer vision. It enables computers to find objects in images and video streams automatically. This capability is crucial for many applications, including surveillance systems, self-driving cars, medical imaging, and automation. Traditionally, object detection relies on complex processes that involve proposing objects, extracting features, and classifying them. These processes can be resource-intensive. Recently, the field has progressed with the introduction of single-stage object detectors like YOLO (You Only Look Once). These detectors have significantly increased the speed of object detection while keeping accuracy high. This paper introduces a new object detection framework called VisionX. VisionX is a web-based system that uses deep learning object detectors and modern web technologies. Its goal is to create a unified framework for various object detection applications. The framework uses YOLOv8 for object detection. It supports multiple input sources, including images, video streams, and webcam feeds. VisionX is built with React for the frontend and Flask for the backend. It includes key features such as object model selection, dataset management, real-time detection, filtering objects based on confidence levels, and identifying suspicious objects for security purposes. The system processes input data through a series of steps, including data preprocessing, object detection with YOLOv8, filtering results, and visualizing the objects. Experimental results indicate that this object detection framework is effective in terms of accuracy and speed. It effectively connects machine learning object detectors with user applications. VisionX can be applied in various scenarios, such as surveillance systems, smart object monitoring, and AI-driven automation.

Index Terms - Object Detection, YOLOv8, Computer Vision, Deep Learning, Real-Time Detection, React, Flask, Web-Based System, Surveillance Systems, Machine Learning.

I. INTRODUCTION

This rapid increase in growth of digital data and the introduction of artificial intelligence technologies has resulted in increased need for the use of intelligent systems in the detection of images. The primary role of this system is object detection. Object detection helps in identifying and locating objects in images. Object detection has a variety of real-world applications.

Some of the uses include intelligent surveillance systems, self-driving cars, health diagnostics, traffic monitoring systems, and automation systems. With the increase in the need for real-time decision-making systems, the need for efficient object detection systems has become paramount. The traditional object detection systems are the Region-based Convolutional Neural Networks (R-CNN), Fast R-CNN, and Faster R-CNN. That choice uses multi-stage detection systems. Their detection accuracy is high. However, using these systems has experienced challenges due to increased computational complexity.

The use of a single-stage object detection system has been introduced to overcome the challenges faced with the conventional system.

The use of single-stage systems has culminated in the creation of the YOLO family of models. The YOLO family of models has attracted a lot of fame in the recent past due to the provision of high-speed detection systems coupled with a fair level of accuracy.

The YOLO family of models against the use, considers detecting objects as a regression problem. This completely rules out the requirement to have different stages in the system for proposal of the regions. It contributes to the overall system's efficiency.

The recent development in the system, including the YOLOv8 model, assists in enhancing the performance of the system, in a holistic manner, in the detection of objects. Its characteristics make it possible for the YOLO system to be used in real-time applications with different input streams such as videos. Despite the improvements in the system over the years, there are certain disadvantages present in the current object detection systems in terms of accessibility and usability. The current systems in the market require a lot of technical knowledge to use them. The present object detection system has not integrated various components of the system, including the dataset management system, the training system, the visualization system, and the object detection system. The poor integration between the various components in the system makes it difficult for users, more so beginners or non-experts, to make effective use of the varied technologies in the system. The need for an end-to-end object detection system was the basis for the paper's proposed system, which is VisionX- An End-to-End Object Detection Platform using YOLO and React. The proposed system is web-based and therefore helps simplify the entire process of object detection. Also, the proposed system should be able to use different input sources like images, videos, and live feeds from webcams. The system is on a client-server and React for dynamically presenting the user in the case of the frontend, combined with Flask for model inference and delivering the APIs in the backend. The object detection pipeline includes input preprocessing, YOLO-based inference, confidence thresholding, identification of suspicious objects, and visualization. The structured pipeline ensures efficient processing, accurate detection, and clear visualization of the results. The major contributions of this work include building a complete system for end-to-end object detection, combining the real-time object detection with an interactive interface, developing a rule-based alert system for identifying suspicious objects, and designing a modular system for scalability and extensibility.

Therefore, VisionX, through the utilization of deep learning and modern web technologies, provides an efficient solution for pushing object detection systems' practical use.

II. EXISTING SYSTEM VS PROPOSED SYSTEM

EXISTING SYSTEM

Traditional object detection system is mainly based on classical computer vision techniques or traditional deep learning techniques that rely on multi-stage processing. Object detection techniques like R-CNN, Fast R-CNN, and Faster R-CNN perform a series of steps to detect objects. These techniques are precise but computationally intensive and result in high processing latency. Therefore, these techniques don't work for real-time applications. Besides, apart from the techniques employed by the system, the existing object detection system is required to be integrated and is not user-friendly. Since object identification involves many processes, the user must perform all the steps manually. This is a huge demerit. Besides, existing object detection systems are unsuitable for video and webcam inputs. Besides, the existing object detection system is incompatible with video and webcam inputs. Existing object detection systems mainly focus on object detection and thus have no ability to interpret the results. For instance, if the system is employed to recognise objects such as knives and weapons, it cannot provide alerts and highlight the objects. Also, there is no proper visualization.

PROPOSED SYSTEM

The VisionX system is a way to detect objects. It uses the deep learning approach and the latest web technologies to overcome the limitations of systems. The VisionX system uses the YOLOv8 model to detect objects. This model is special because it can find and classify objects in one step. The YOLOv8 model is also very good at detecting objects in real-time with high accuracy.

The VisionX system has a user interface that's easy to use. It is built with the React web framework. This means that people can use the object detection system without needing to know a lot about technology. The VisionX system can use different types of input, like images, videos, and the webcam. The backend of the system is built with the Flask web framework. This allows the system to use REST API endpoints to process the input data. The VisionX system is designed in a way. It has parts, including: Preprocessing, YOLO detection, Confidence filtering, and Result generation. The VisionX system can also generate alerts when it detects objects. It uses the YOLO model to detect objects. When the system detects something

like a weapon it sends a message. The VisionX system has an alert system. It also has visualization tools, such as a bounding box, class label, confidence score, and confidence bar. The VisionX system is designed to be modular. This means that it can be extended to use models and tools in the future. The VisionX system uses the YOLO model to detect objects. The VisionX system is a way to detect objects because it is accurate and easy to use. The VisionX system has features that make it useful. The VisionX system is a tool for detecting objects.

III. RELATED WORK

Object detection is an important area of research in computer vision. Over the past twenty years there have been a lot of improvements in object detection techniques. Earlier techniques used image processing and machine learning approaches, like the Haar feature-based cascade classifier that was developed by Viola and Jones. These techniques were able to detect objects in time for specific applications, such as face recognition. However, object detection techniques were not able to work for other applications. The use of deep learning techniques in object detection has made a difference. Region-based Convolutional Neural Network (R-CNN) techniques have made improvements in object detection. This technique was developed by combining region proposal techniques with neural networks. Then Fast R-CNN and Faster R-CNN improved this technique by integrating region proposal networks. These techniques have improved the accuracy of object detection. Reduced the time it takes to run the algorithm.

To make object detection techniques efficient, single-stage object detection techniques have been developed. YOLO, which stands for You Only Look Once, is an object detection technique that is based on the single-stage framework. This technique was proposed by Redmon et al. It has changed the way we do object detection by turning the object detection problem into a regression problem. This technique predicts the class probability and bounding box coordinates directly from the image. This has made object detection more efficient by reducing the time it takes to run the algorithm. Then YOLOv3, YOLOv5, and YOLOv8 improved the performance of the YOLO framework by using network architectures with better feature extraction techniques and training mechanisms. Other single-stage object detection models include Single Shot MultiBox Detector (SSD and RetinaNet. Although these models have shown results, YOLO-based object detection is still the preferred choice because it is fast and simple. Recently, researchers have been working on integrating object detection with web-based platforms and real-world applications. They have explored ways to use deep learning-based object detection systems in surveillance, autonomous monitoring, and other real-world applications. However, the existing object detection systems do not have an interface. Also, they cannot be integrated with tools and platforms. Existing object detection systems cannot understand the context and make decisions. Although the system can detect objects, it does not understand the importance of the objects it detects. For example, if the system is used to detect objects and it identifies a weapon, the system will not be able to provide any alerts or information. This is because the system does not understand the importance of the objects. This is a problem with the existing object detection system. Unlike existing object detection systems, the proposed VisionX system is an integrated system that combines the benefits of YOLO-based object detection with web-based platforms. By supporting real-time object detection and an alert-based decision mechanism, the proposed VisionX system is a solution to the existing object detection system. The VisionX system is an improvement over the existing object detection systems because it can understand the context and make decisions. The VisionX system can detect objects. Provide alerts and information about the objects it detects.

This makes the VisionX system a useful and effective object detection system.

IV. METHODOLOGY

VisionX is a proposed system developed in a modular, scalable manner. The system can perform object detection in real-time through a web-based integrated system. The advantage of the system is that it uses deep learning detection as well as the interaction of both its frontend and backend. The entire system consists of several stages.

4.1 System Architecture Overview

The VisionX system was built using the Client / Server architecture to facilitate easy communication between user interaction and calculation components. The Client / Server architecture is divided into three main layers – Presentation Layer, Application Layer, and Model Layer. Presentation Layer developed using React and is responsible for the user interface, user interaction, and displaying the results of an object detection task. The Application Layer developed in Flask framework and serves as the back-end processing engine. The Model Layer consists of the YOLOv8 deep learning model used for the actual detection task.

4.2 Input Acquisition and Handling

The system has been developed in such a manner that it will be able to accommodate different sources of input. This will enable the system to be flexible and adaptable to different application requirements. For instance, the user will be able to provide the input in the form of static images, video files, and webcam. For the static image, the detection will take place on the input itself. However, in the case of video and webcam, the system will take the frames at regular intervals and process them one by one.

This will enable the system to function in near real-time conditions.

4.3 Data Preprocessing

Before proceeding with object detection, there are various preprocessing activities to be performed on the input data to make it compatible with the YOLO model. The input image is decoded, and it is converted to an appropriate format. The image is then resized to match the input size required for the model. This ensures uniformity in all input images.

Then, normalization is performed on the pixels to make it more numerically stable. The image is then converted to tensor data type using NumPy.

4.4 YOLO-Based Object Detection

The major detection mechanism employed by VisionX is based on the YOLOv8 model. This is based on a single-stage object detection approach. This is different from other object detection mechanisms, where localization and detection are done separately. In YOLOv8, localization and detection are done simultaneously. This is advantageous since it speeds up the detection process by reducing the computational cost.

The YOLOv8 model is based on a grid-based approach. This means that the image is divided into a grid, and predictions are made based on this grid. Each prediction is based on the probability that an object exists in a certain region of the image. This prediction is based on a confidence score that represents the probability and accuracy of the detection box.

4.5 Confidence Filtering

Once the model has been inferred, a post-processing step occurs. During the post-processing step, a confidence threshold is used to filter out the detections. This filtering ensures that false positives are eliminated. Only the detections that have a confidence value above the predefined threshold will be considered. This filtering step ensures the reliability of the system. It plays a crucial role in enhancing the precision of the system.

4.6 Suspicious Object Identification

In order to extend the system beyond object detection, VisionX uses a rule-based method for detecting suspicious objects. The system uses a predefined set of object categories that are related to potential threats. These objects may include weapons.

During the post-processing phase, object labels are compared to a predefined set. If there is a match and the object label satisfies the threshold condition, then the object is declared suspicious. This feature allows for contextual awareness within the system so that results can be interpreted accordingly.

4.7 Alert Generation

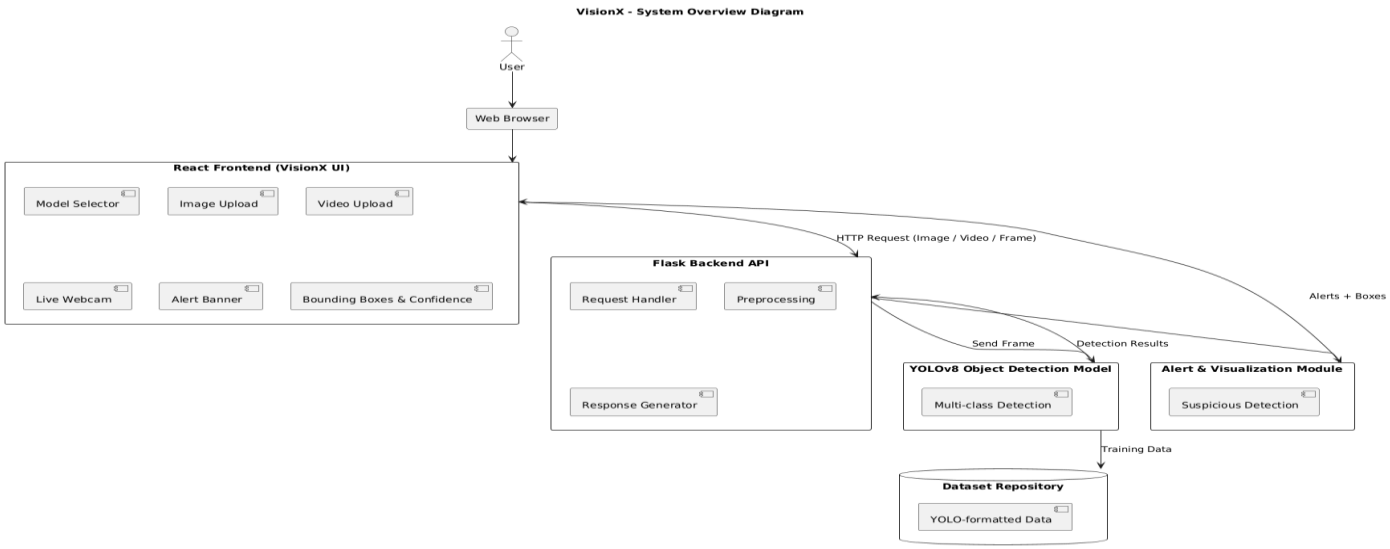
Once a suspicious object has been detected, the system initiates the process for the generation of the alert. Alerts are presented to the user via the frontend interface. These alerts are visually highlighted to ensure the user's prompt attention. This system would be helpful in applications where timely responses are necessary, such as in security surveillance. The alert system makes the system more practically useful by providing the user with insights based on the results.

4.8 Backend Processing and API Communication

The Flask backend is essentially the central processing unit, which enables the interface and the model to communicate effectively. This is done by defining RESTful APIs, which accept data from the interface, preprocess the data, use the YOLO model for processing, and produce results. These results are presented in JSON data format, which contains information about the detected objects, their confidence, and their coordinates. This data is standardized, which enables effective data exchange between different components of the system.

4.9 Result Visualization

For the visualization of the detection results, the React-based front-end takes care of the task. The front-end dynamically draws the bounding boxes on the images, video frames, or live video streams. The detected objects are marked with their corresponding classes and confidence levels. Other visual elements like confidence indicators and alert boxes may be added to the design for better interpretability. The use of dynamic rendering ensures the smoothness of the user interface as the new detection results arrive in real-time.

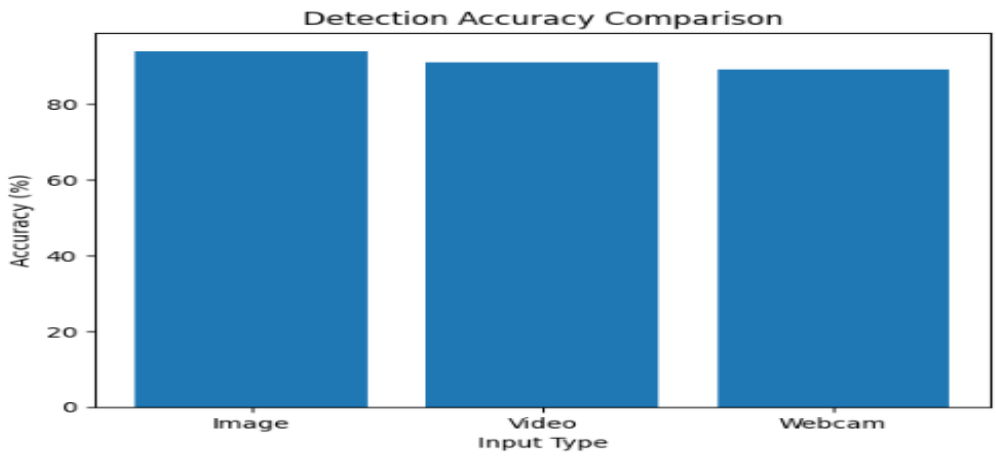


V. RESULTS AND DISCUSSION

5.1 Detection Accuracy

The performance of VisionX was tested with an image dataset containing images and video frames with various classes of objects, including general classes and security-related classes such as weapons. The performance of YOLOv8 was highly accurate, allowing for efficient identification of various classes of objects with high confidence. The use of confidence-based filtering was effective in increasing the accuracy of the system by removing weak detections. The system was highly accurate for visible objects, with acceptable performance for images with varied lighting conditions. The results show that the YOLO approach is efficient for real-time applications, providing both speed and accuracy.

Fig 5.1 represents the accuracy of VisionX for detecting various types of input images. The results show that VisionX has the highest accuracy for static images, followed by video and webcam images with minor variations.



5.2 Performance Analysis

The performance of the VisionX system was assessed in terms of processing time, latency, and system responsiveness in real-time. Since the YOLOv8 model is based on a single-stage architecture, the system is able to effectively process the data without any significant latency. It was observed that the processing time per image was relatively low on average, thereby ensuring faster processing even in continuous data streams like videos and webcam feed data. Moreover, the data transfer between the React frontend and the Flask backend was optimized by using lightweight APIs, ensuring effective data transfer without any latency.

Table 5.1 shows the performance metrics of the proposed system for different types of data. It is evident from the results that processing static images is faster compared to processing other types of data, while videos and webcam data have some latency due to continuous processing.

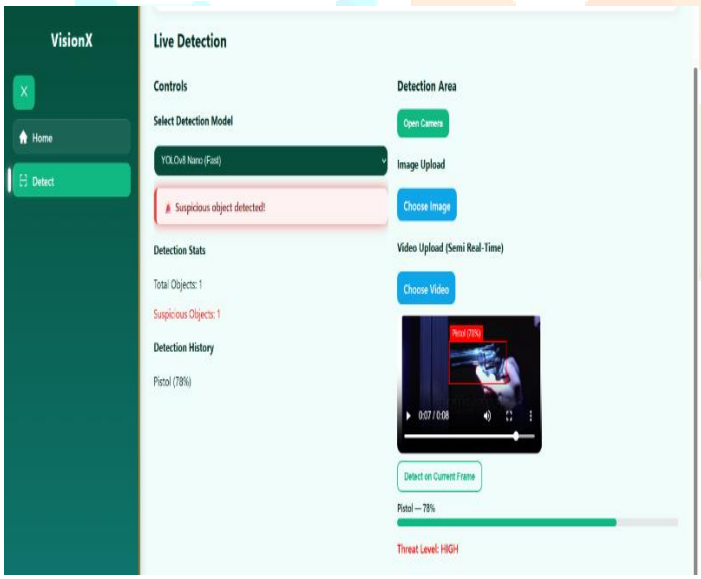
Input Type	Average Processing Time (ms)	Detection Accuracy (%)
Image	45	94
Video	60	91
Webcam	75	89

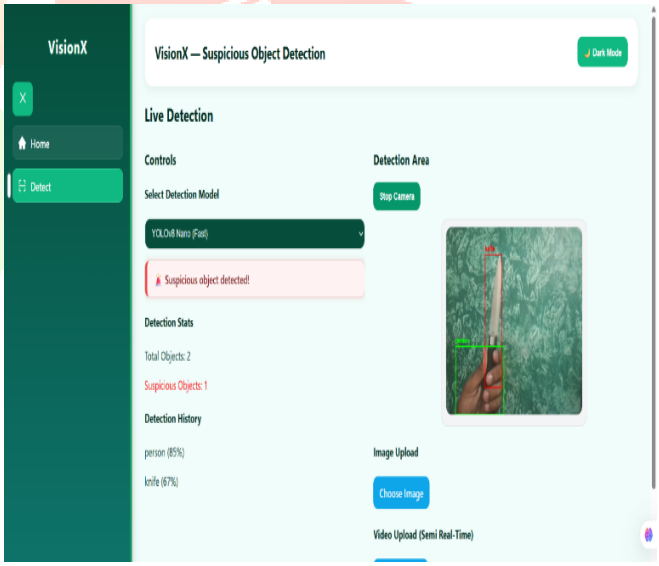
Table 5.1 Detection Performance Metrics

5.3 Detection Performance Metrics

The VisionX system facilitates real-time visualization of the output of the detection results in real time. This is achieved through an interactive and dynamic user interface. In this case, the system uses bounding boxes in highlighting the different objects in the image. This is done in conjunction with the class labels and the corresponding confidence scores. This enables the user to better understand the output of the object detection. It is also possible for the VisionX system to perform multi-object detection in a single frame. In addition, it ensures consistent quality in the visualization of the output from different sources, including images, videos, and webcams. The bounding boxes provided by the YOLOv8 model are accurate in terms of the placement of the boxes in relation to the different objects in the image. This ensures that there is high accuracy in terms of the output provided by the model. In addition, the output provided has a corresponding confidence score, which enables the user to have a better understanding of the output provided by the model. This is because it enables the user to have a better understanding of the output provided in real time.

Fig 5.2 shows some of the output provided by the VisionX system for different sources. This demonstrates the capability of the VisionX system in terms of providing consistent output for different sources, including images and videos.





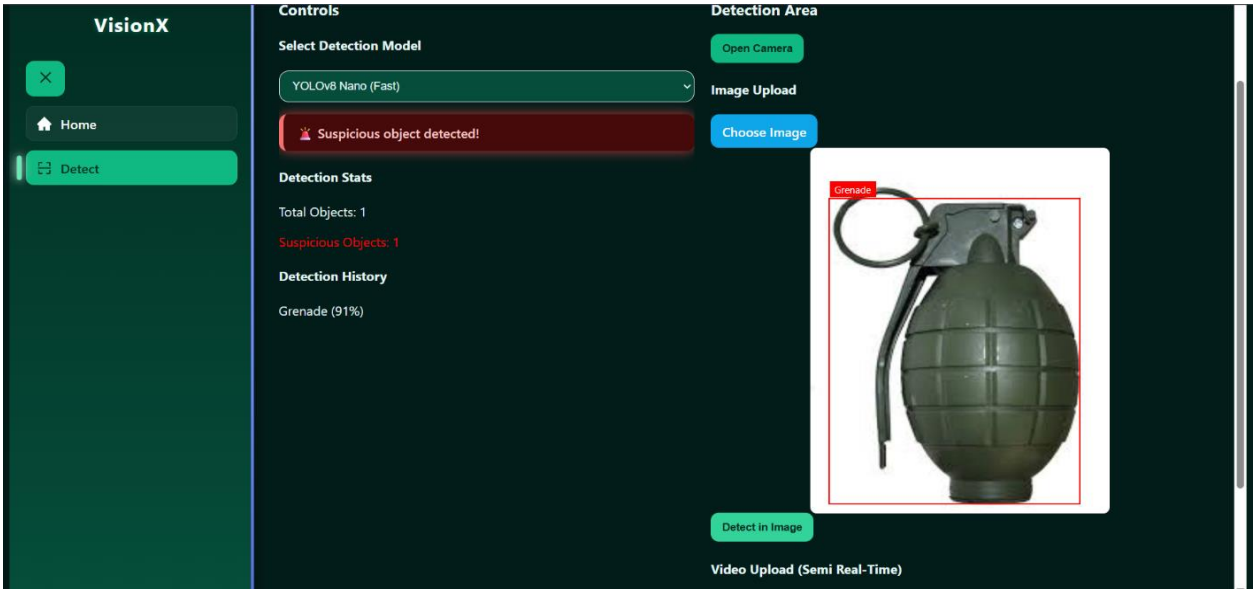


Fig 5.2 Detection Outputs for Image, Video, and Webcam Inputs

5.4 Suspicious Object Detection Results

The VisionX system has a rule-based system for detecting suspicious objects, thus increasing its application in security and surveillance scenarios. In this system, the labels of the identified objects are analyzed and matched against a set of defined labels associated with potential threats like weapons. Once a match is found and the confidence level is above a threshold, the object is marked as suspicious.

The alert system has been implemented for providing instant visual feedback to the user through the interface. In this system, alerts are provided in a manner that they are prominently displayed so that there is no room for missing important information. This makes the system an intelligent tool for object detection, thus increasing its application in surveillance scenarios.

Figure 5.3 shows the output of the alert system, in which suspicious objects are successfully identified and highlighted. This makes the system suitable for real-time application scenarios.

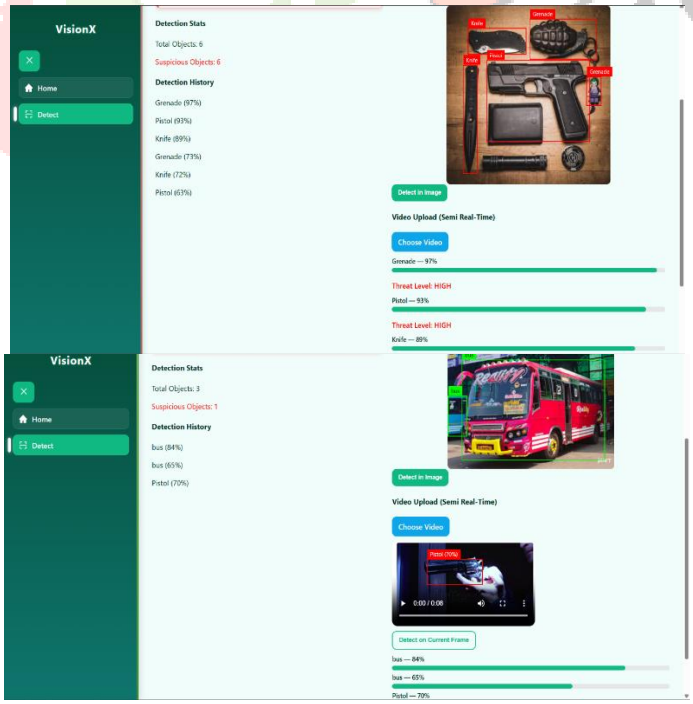
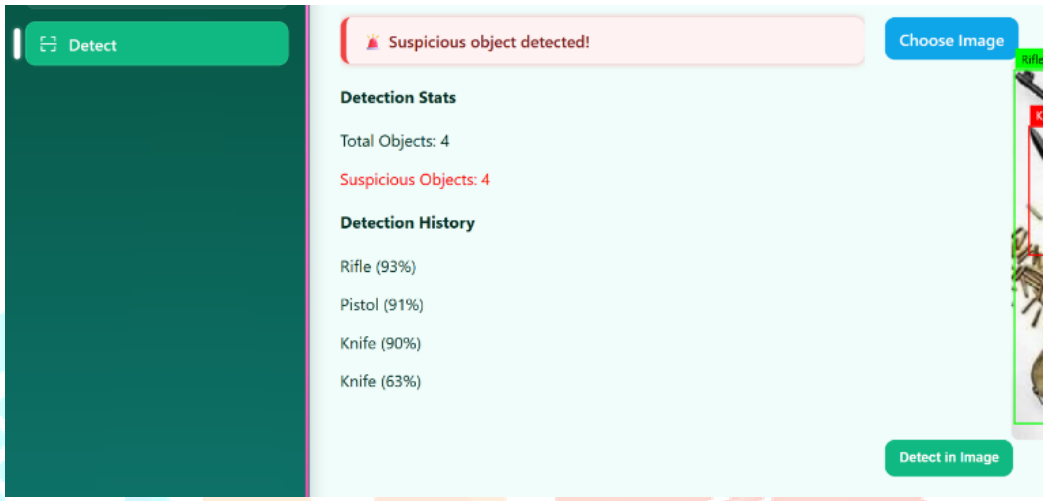


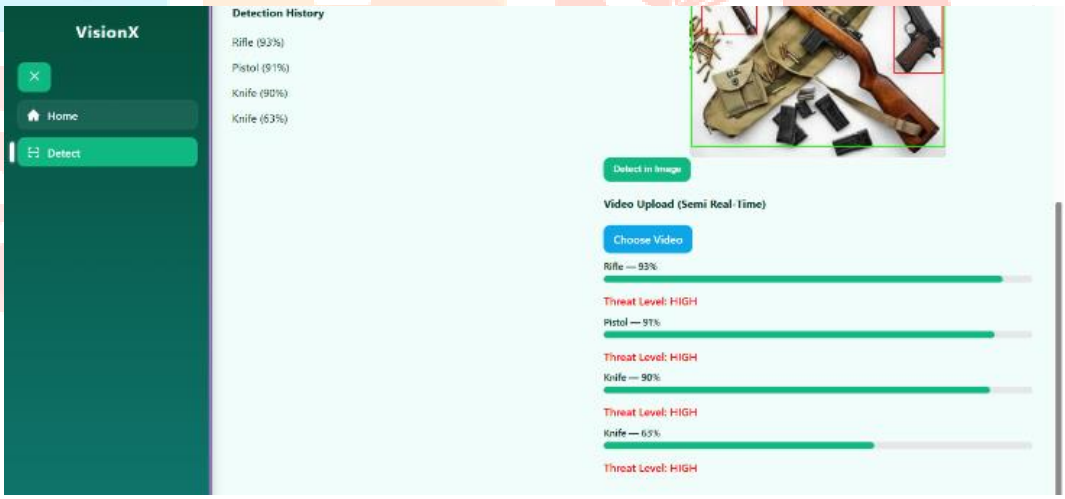
Fig 5.3 Suspicious Object Detection and Alert Generation

5.5 Overall System Evaluation

The performance of VisionX shows the effectiveness of the system as a tool for object detection in real-time. The use of YOLOv8 with the addition of a web interface shows the efficiency of the system in object detection. VisionX can be considered better than other object detection systems based on speed, efficiency, and the use of real-time technology.

The performance of the system shows effective handling of different input sources. The performance of the system is consistent for different use cases. This shows the effectiveness of the system in different conditions. Based on the performance of the system, it can be concluded that VisionX is effective for use in different applications. Although the performance of the system is effective, the accuracy of the results can be increased by training the model on large datasets. Based on the results, it can be concluded that VisionX is effective in finding the right balance between efficiency, accuracy, and usability.





VI. FUTURE SCOPE

There are clearly numerous options for future enhancements to VisionX, a system using deep learning and web technologies for real-time object detection.

One possibility for future enhancement of the system is by integrating more advanced architectures/models for object detection. The VisionX system implements the YOLOv8 architecture for object detection, which is extremely accurate and efficient in time/device resources to do so. There are also a wide variety of other advanced architectures/models for object detection, such as transformer-based architectures/models for object detection that may be usable/enhance object detection in the future.

In addition, the accuracy of the VisionX system may be further improved by training the system with increasingly large and varied datasets, thereby enabling representations to be built from increasingly large and varied datasets, including domain-specific datasets and dataset representations (including all of the various datasets used to create data representations) for improving object detection effectiveness and accuracy for a wide variety of applications, including security and surveillance applications; thus improving the ability with which the system may efficiently detect a wider array of higher risk objects, such as weapons or dangerous substances.

Currently, the VisionX system is running in a local environment, however; it could be further optimized for real-world scenario deployments through cloud-based enhancements. In this case, the VisionX System's backend could be deployed in one (or more) of the many cloud computing platforms such as AWS, Google Cloud Platform or Azure, allowing multiple simultaneous users to be served more efficiently by the VisionX system. Another potential extension of the VisionX system is the ability to support real-time multi-camera systems. If multiple real-time video streams may be processed by VisionX, it could ultimately be deployed for many different applications including smart city initiatives, video surveillance systems and industrial monitoring. This extension will require optimizing the VisionX system to appropriately process multiple real-time video streams.

An additional future development direction for the VisionX system would be the capability to support edge computing. VisionX could leverage several types of edge computing devices, such as NVIDIA Jetson devices, smartphones, etc., if deployed as an edge computing system. This will provide the VisionX system with a better overall performance than that experienced using a traditional system, by allowing for video processing at the device level as opposed to using the more traditional cloud architecture. This would be especially beneficial in cases where the device has limited or no network connectivity. VisionX can be extended to support video analytics features such as object tracking, motion detection, and activity recognition. Therefore, as an extension, the system will be able to analyze the video stream through time thereby improving its performance. For example, the system will be able to track suspicious objects in the video stream to improve its overall performance. Improvements in the system's alerting ability will also improve usability. For example, additional alerting capabilities such as email alerts, SMS alerts and push notifications could extend the ability of the system to send alerts to users even if they are not currently monitoring the user interface. Also, the system can be extended to support different types of devices, such as IoT devices and therefore will be able to respond to different events. Finally, performance optimization techniques such as quantisation, pruned models, and hardware acceleration enable the system to minimize the computational resources required for operation. It is important to include these techniques during the deployment of the system on devices with limited resources. For all of the above reasons, VisionX's ability to provide real-time object detection is quite robust, the future enhancement of VisionX should present a very good and continued opportunity for further enhancement of this capability.

VII. CONCLUSION

This research presents VisionX, which is an innovative End-to-End Object Detection system that merges Deep Learning and Modern Web Technology to enable Interactive Real-time Object Detection System for users. The developed VisionX system has been effectively developed using YOLOv8 Object Detector as the method for object detection, as well as utilizing ReactJS to build the Front End and Flask to build the Back End of the VisionX application, creating a seamless interaction between user inputs and object detections. Moreover, the proposed VisionX System integrates various sources of user input, including, but not limited to, images, videos, and webcams; therefore, it resolves the issues experienced with traditional object detection approaches due to their individualistic, non-integrated processing methods. Furthermore, a structured object detection pipeline is created and implemented within VisionX to ensure specific and accurate results while detecting objects. Additionally, an algorithmic-based approach is utilized within the system to detect suspicious objects and therefore, would have increased utility when applied to security based applications. Experimental results indicate that the VisionX has been developed to achieve an efficient balance between the accuracy and efficiency of providing object detection results. Finally, as the VisionX System has been developed to handle multiple types of input data in real-time, thus increasing the efficiency for practical applications. Additionally, due to the modular structure of the VisionX System, it is designed to allow for improvements as technology evolves in future work.

Overall, VisionX has proven to be a highly efficient object detection system that provides users with an interactive and real-time object detection application. This new system will serve as the link from the theoretical model of deep learning to the applied world. VisionX’s ability to integrate an object detection, visualization, and alert generation system will enable various uses, including but not limited to security, tracking, and automation solutions.

VIII. REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
Available: https://arxiv.org/abs/1506.02640

[2] A. Bochkovskiy, C. Y. Wang and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.
Available: https://arxiv.org/abs/2004.10934.

[3] G. Jocher et al., "Ultralytics YOLOv8 Documentation," Ultralytics, 2023.
Available: https://docs.ultralytics.com

[4] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems (NeurIPS), vol. 32, 2019.
Available: https://arxiv.org/abs/1912.01703

[5] "React Documentation," Meta, 2024.
Available: https://react.dev

[6] "Flask Documentation," Pallets Projects, 2024.
Available: https://flask.palletsprojects.com

[7] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.