Night Vision Object Detection Using Machine Learning

BCA(DATA SCIENCE)

Supervisor : Mrs.S.SABARIA

Designation :Assistant professor

Student Name :

GUKAN.S

(221421601013)

SANJAY.G

(221421601048)

KAUSHIK.A.S

(221421601018)

DEPARTMENT OF COMPUTER APPLICATIONS

ABSTRACT

- The project focuses on enhancing road safety by detecting objects such as animals, pedestrians, and vehicles in low-light or nighttime driving conditions using a standard night-vision camera.
- Unlike conventional methods that rely on thermal cameras, the project uses a regular night-vision- enabled camera combined with machine learning techniques for real-time object detection.
- The developed system detects and classifies objects (animals, people, vehicles) within the camera's range, alerting the driver when any object is within a 150-meter range.
- It employs deep learning-based object detection models to process the camera feed and recognize the objects in real-time, offering enhanced driver awareness.
- The approach provides an efficient and cost-effective alternative to expensive thermal imaging solutions, aiming to improve road safety without the high costs typically associated with thermal cameras.

OBJECTIVE

- **Enhance Road Safety**: To improve road safety in low-light and nighttime driving conditions by detecting and classifying objects such as animals, pedestrians, and vehicles.
- **Cost-effective Alternative**: To create a more affordable and efficient alternative to traditional thermal cameras by using standard night-vision-enabled cameras combined with machine learning techniques.
- **Real-time Object Detection**: Develop a system that can perform object detection in real-time using video feeds from a vehicle's front-facing camera.
- **Alert System for Drivers**: Implement an alert system that notifies the driver when an object is within a 100-meter range, enhancing awareness of potential hazards.
- **Use of Deep Learning**: Leverage deep learning-based object detection models for precise and accurate identification and classification of objects, making the system robust and reliable.

PROBLEM IDENTIFIED

- Traditional vehicle sensors, including standard cameras, often fail to detect objects such as animals or pedestrians effectively in low-light or nighttime conditions, leading to potential accidents.
- Current solutions rely heavily on expensive thermal cameras for night vision, making them out of reach for many drivers and reducing accessibility for broader use.
- Many existing detection systems suffer from slow response times, which can be critical in preventing accidents, especially when objects are suddenly encountered within close range.
- Some existing systems do not provide real-time alerts or fail to alert drivers early enough to react, leaving gaps in critical driver awareness during low-visibility conditions.
- Existing systems often fail to efficiently identify and classify objects in challenging environments like poorly lit roads, thereby compromising driver safety and confidence.

Base paper Title: A novel framework for vehicle detection and tracking in night ware surveillance systems

Authors: NOUF ABDULLA HALMUJALLY1, ASIFA MEHMOOD QURESHI2, ABDUL WAHAB ALAZEB3,HAMEEDUR RAHMAN2,TOUSEEF SADIQ 4,(Graduate Student Member, IEEE),MOHAMMED ALONAZI5, ASAAD ALGARNI 6,ANDAHMAD JALAL.

Year: 2024

Objective: The primary objective is to develop a robust model that enhances the visibility and tracking accuracy of vehicles captured in aerial images during nighttime operations.

Limitation: The proposed method performs well for night time surveillance of road traffic.However, there are still some limitations of the model.

LITERATURE SURVEY

S. No	Title	Author	Year	Pros & Cons
1	A novel framework for vehicle detection and tracking in night ware surveillance systems	Almujally, N. A., Qureshi, A. M., Alazeb, A., Rahman, H., Sadiq, T., Alonazi, M., ... & Jalal, A.	2024	**Pros**: Innovative vehicle detection for night surveillance. **Cons**: Limited to night conditions and surveillance systems.
2	Object detection for night surveillance using Ssan dataset based modified Yolo algorithm in wireless communication	Murugan, R. A., & Sathyabama, B.	2023	**Pros**: Modified YOLO for night surveillance improves detection. **Cons**: Performance dependent on dataset quality.
3	Multiple pedestrian detection and tracking in night vision surveillance systems	Raza, A., Chelloug, S. A., Alatiyyah, M. H., Jalal, A., & Park, J.	2023	**Pros**: Effective for detecting and tracking pedestrians in night vision systems. **Cons**: Needs high computational power.
4	A Robust Framework for Traffic Object Detection using Intelligent Techniques	Nandhini, T. J., & Thinakaran, K.	2023	**Pros**: Robust framework for traffic object detection. **Cons**: May not handle complex or dynamic traffic conditions well.

LITERATURE SURVEY

S. No	Title	Author	Year	Pros & Cons
5	Object detection in autonomous vehicles under adverse weather: A review of traditional and deep learning approaches	Tahir, N. U. A., Zhang, Z., Asim, M., Chen, J., & ELAffendi, M.	2024	**Pros**: Comprehensive review of traditional and deep learning methods for vehicle detection. **Cons**: Does not propose novel solutions.
6	Improved metaheuristics with deep learning based object detector for intelligent control in autonomous vehicles	Alasmari, N., Alohali, M. A., Khalid, M., Almalki, N., Motwakel, A., Alsaid, M. I., ... & Alneil, A. A.	2023	**Pros**: Integrates metaheuristics with deep learning for intelligent vehicle control. **Cons**: Complex, might not be ideal for real-time applications.
7	SMART on-board multi-sensor obstacle detection system for improvement of rail transport safety	Ristić-Durrant, D., Haseeb, M. A., Banić, M., Stamenković, D., Simonović, M., & Nikolić, D.	2022	**Pros**: Enhances rail transport safety with a multi-sensor system. **Cons**: May require expensive sensors and setup.
8	Helmet detection using machine learning approach	Shenoy, M. A., Betrabet, P. R., & NS, K. R.	2022	**Pros**: Efficient helmet detection for safety applications. **Cons**: Limited scope, focusing only on helmet detection.

LITERATURE SURVEY

S. No	Title	Author	Year	Pros & Cons
9	Smart assistive system for visually impaired people obstruction avoidance through object detection and classification	Masud, U., Saeed, T., Malaikah, H. M., Islam, F. U., & Abbas, G.	2022	**Pros**: Assistive system for visually impaired using object detection. **Cons**: Requires constant sensor updates for optimal performance.
10	Artificial intelligence based object detection and tracking for a small underwater robot	Lee, M. F. R., & Chen, Y. C.	2023	**Pros**: Uses AI for object detection and tracking in underwater robotics. **Cons**: Specific to small underwater robots, limiting its broader application.

EXISTING SYSTEM

- Most approaches rely on thermal cameras for detecting vehicles, animals, and pedestrians.
- While effective, thermal imaging solutions are expensive and not widely available.
- Many systems use conventional detection techniques, which can be computationally heavy.
- Existing methods struggle in complex and dynamic settings.
- Most systems work well only under specific conditions or with particular types of cameras.
- Constraints in computational efficiency limit real-time processing capabilities.

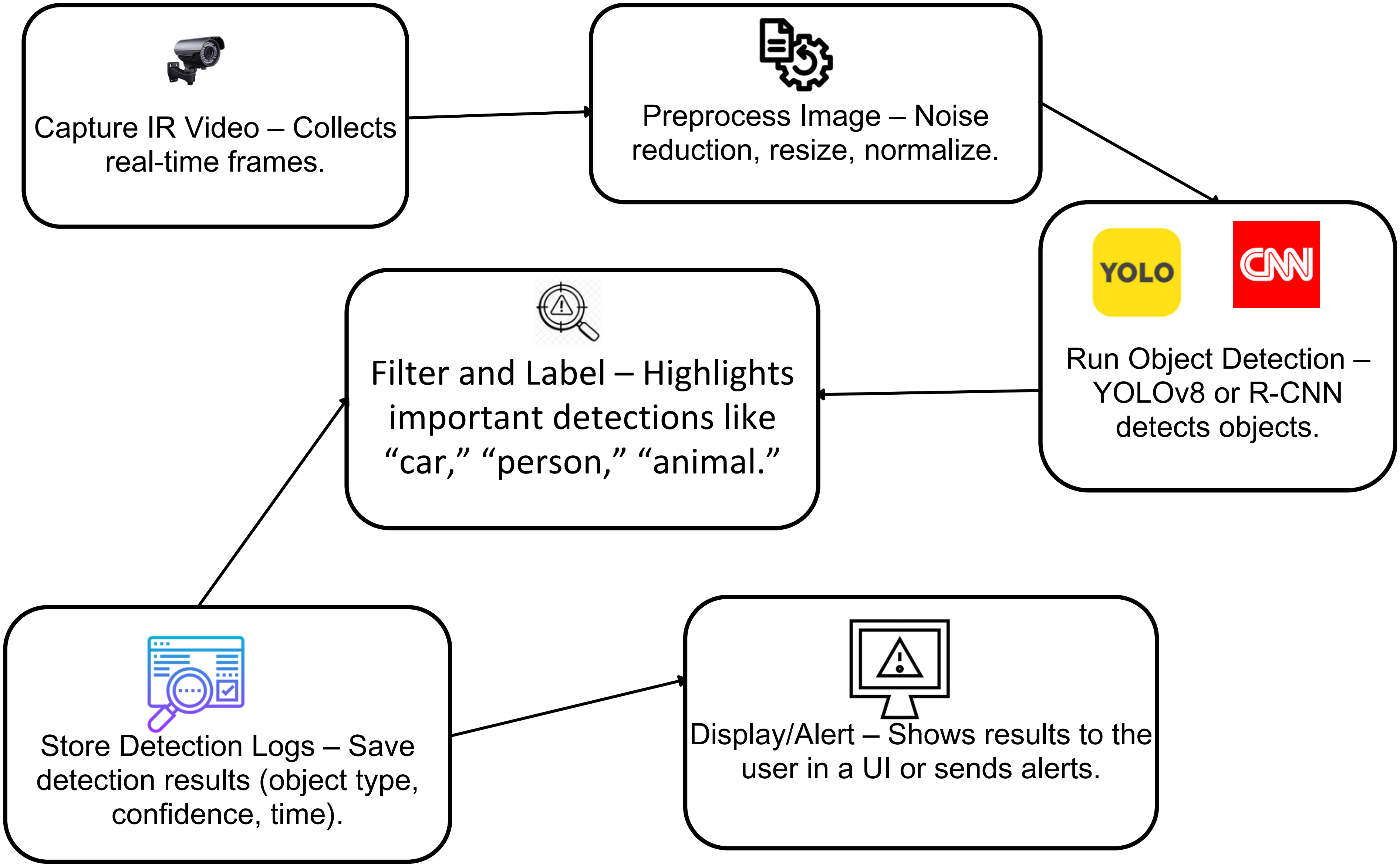
PROPOSED WORK

- The proposed work aims to overcome limitations of current methods.
- Uses standard night-vision-enabled cameras combined with deep learning-based object detection models.
- More cost-effective, accessible, and capable of real-time detection.
- Classifies objects such as vehicles, pedestrians, and animals in low-light conditions.
- Provides an alert when any object is within a 150-meter range of the vehicle.
- Enhances driver awareness and road safety using advanced machine learning techniques.
- Offers a more efficient and affordable solution compared to thermal imaging systems.

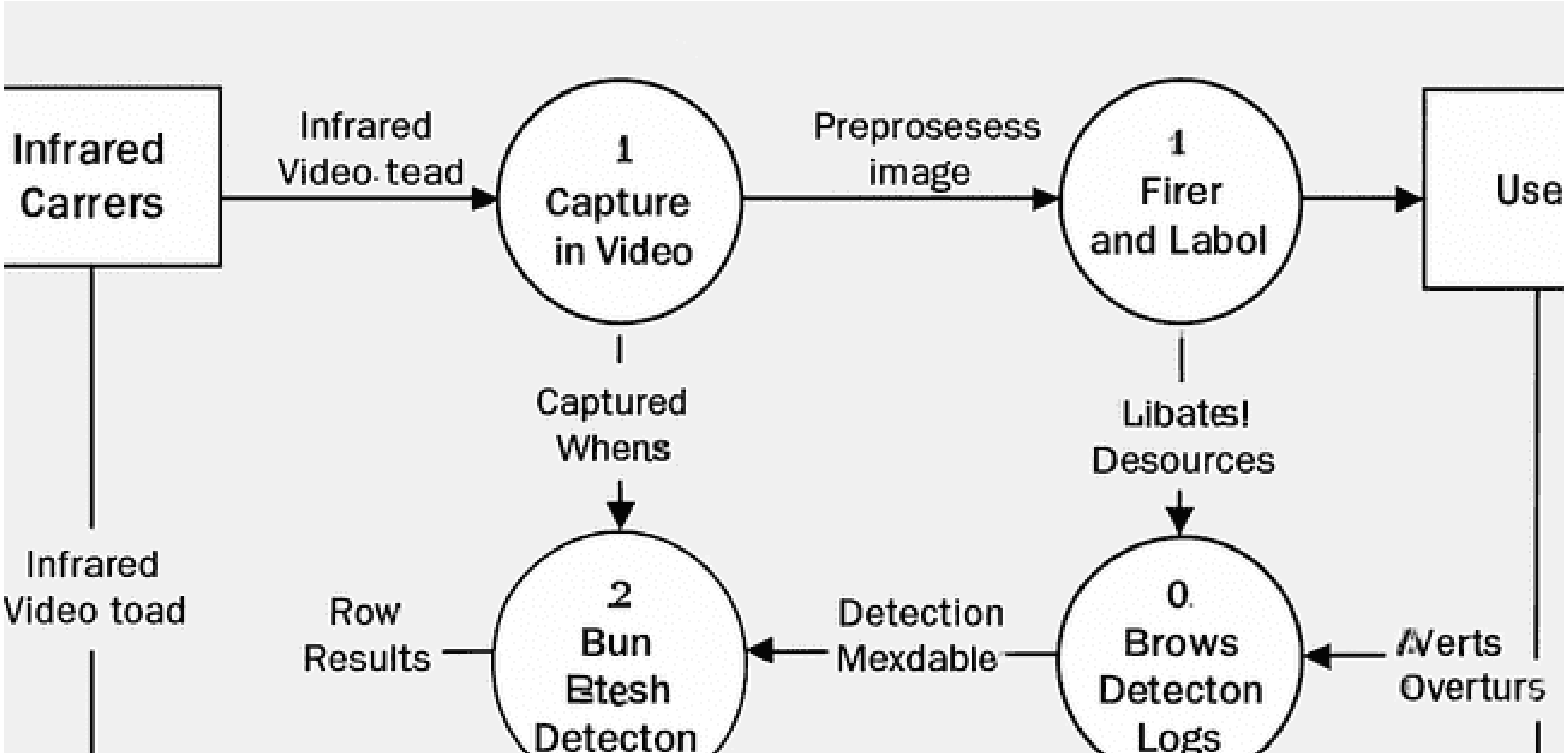
METHODOLOGY

1. **Data Collection**: Mention how you gathered the data. For night vision object detection, you might be using infrared camera footage, along with a dataset of labeled objects (e.g., cars, people, animals) in low-light conditions.
2. **Preprocessing**: Describe any preprocessing steps. This could involve image enhancement, noise reduction, or converting images to grayscale or a different format suitable for your model.
3. **Model Selection**: Explain the models you chose (e.g., YOLOv8 and R-CNN) and why they are suitable for your task. YOLOv8 might be good for real-time object detection, while R-CNN could provide better accuracy in detecting smaller or more complex objects.
4. **Training**: Explain how you trained your models (hyperparameters, batch size, number of epochs, etc.).
5. **Evaluation**: Mention how you evaluated your model performance, such as through accuracy, precision, recall, F1 score, or custom metrics related to your specific use case.

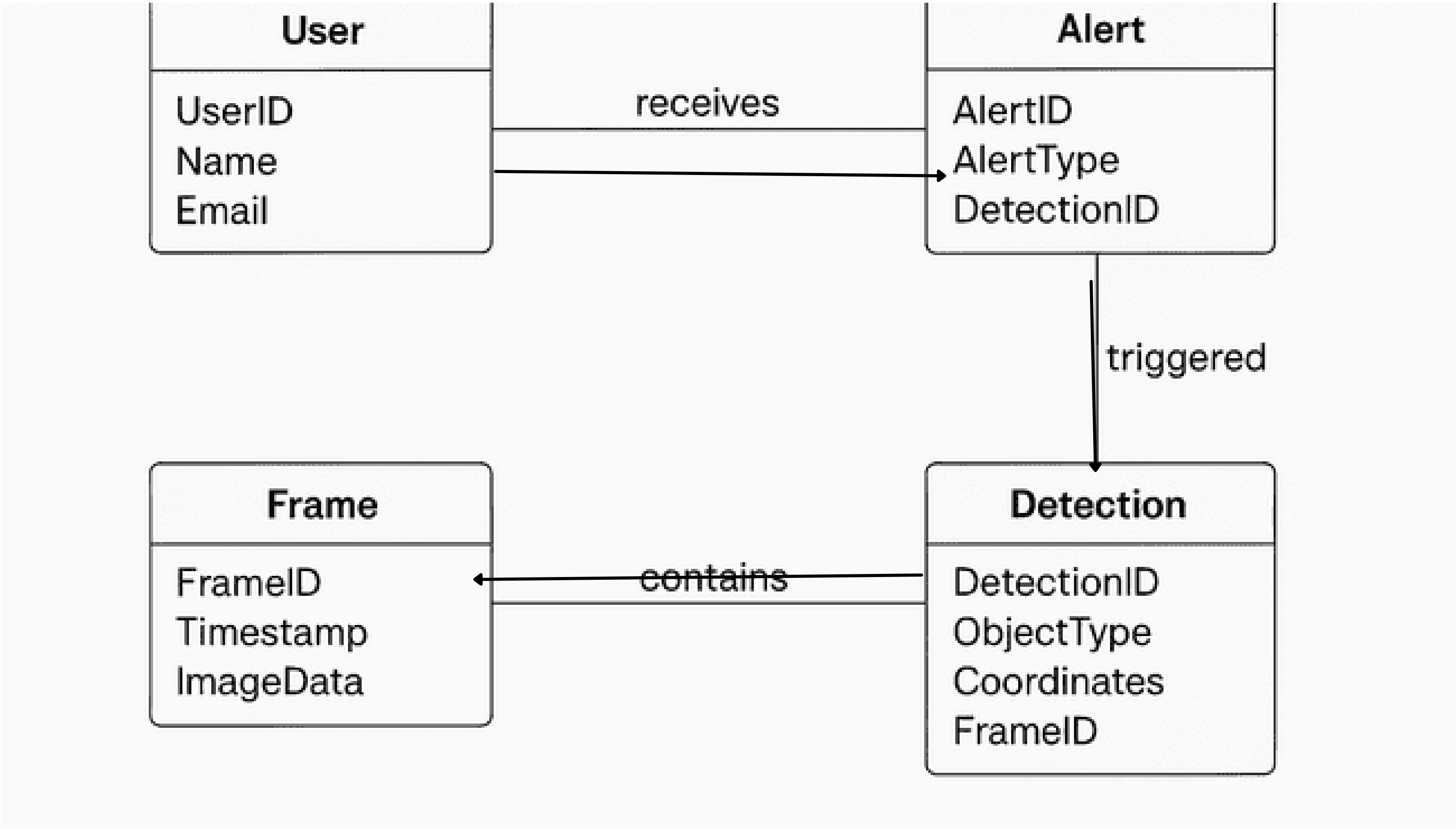
ARCHITECTURE DIAGRAM



B.S. Abdur Rahman Crescent Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956
GST Road, Vandalur, Chennai 600 048



ER DIAGRAM


User
UserID
Name
Email
receives
Alert
AlertID
AlertType
DetectionID
triggered
Frame
FrameID
Timestamp
ImageData
contains
Detection
DetectionID
ObjectType
Coordinates
FrameID

MODULES

1. **Data Collection and Preprocessing Process (EDA)**
2. **Object Detection Model**
3. **Distance Estimation**
4. **Alert System & Output Handling**

Module 1: Data Collection and Preprocessing Process (EDA)

- **Data Collection**: Source datasets from open platforms like Kaggle and Roboflow, including images of animals, pedestrians, and vehicles for night-time object detection.
- **Data Cleaning**: Remove unnecessary rows and columns, handle missing data, and eliminate duplicates to refine the dataset for better quality.
- **Data Preprocessing**: Normalize data, resize images, apply image augmentation (e.g., rotations, flips) to improve model performance.
- **Exploratory Data Analysis (EDA)**: Visualize dataset features, distributions, and class balance (e.g., object counts, bounding box sizes) to understand data characteristics.
- **Data Splitting**: Split the data into training, validation, and testing sets to prepare for model training and evaluation.

B.S. Abdur Rahman Crescent Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956
GST Road, Vandalur, Chennai 600 048

Module 2: Object Detection Model

- **Model Selection**: Choose a suitable deep learning model (e.g., **YOLOv8**, Faster R-CNN) for object detection tasks.
- **Training the Model**: Load the preprocessed data and train the selected model, using techniques like data augmentation and hyperparameter tuning.
- **Model Evaluation**: Use a test dataset to evaluate the model's performance using accuracy metrics like mean average precision (mAP).
- **Fine-tuning**: Adjust model hyperparameters, such as learning rate, batch size, and epochs, to optimize detection accuracy.
- **Model Export**: Save the trained model for future use in production (e.g., .h5 for Keras, .pth for PyTorch).

Module 3: Distance Estimation

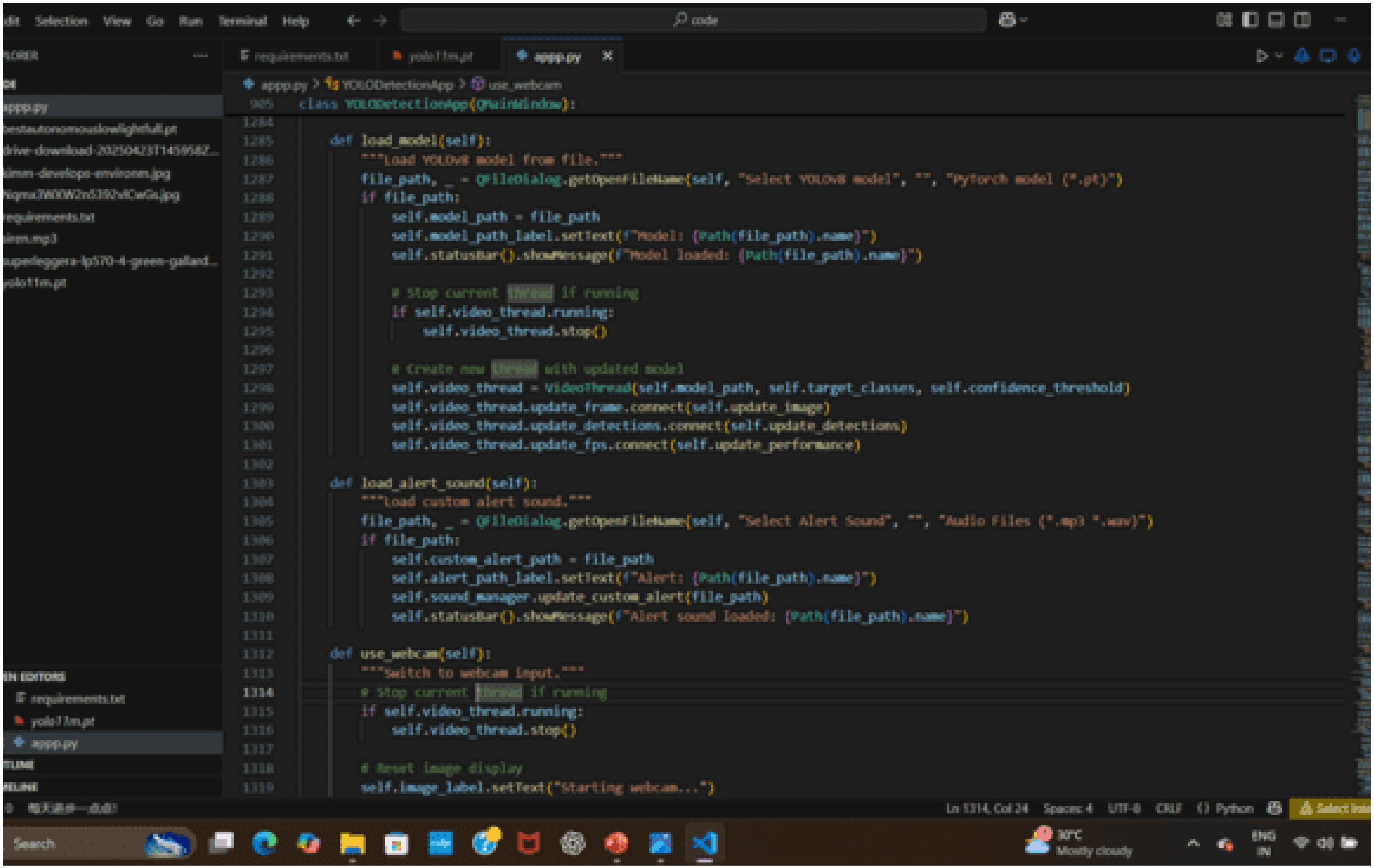
- **Model Integration**: Integrate the trained model into the backend using TensorFlow or PyTorch.
- **Inference Engine**: Develop an API (e.g., FastAPI or Flask) to handle model inference requests.
- **Data Processing**: Preprocess input images and format model outputs (bounding boxes, labels, etc.).
- **Real-time Processing**: Implement real-time object detection with distance-based alerts (e.g., within a 150-meter range).
- **Optimization**: Optimize model inference speed for real-time detection.

B.S. Abdur Rahman Crescent Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956
GST Road, Vandalur, Chennai 600 048

Module 4 :Alert System & Output Handling

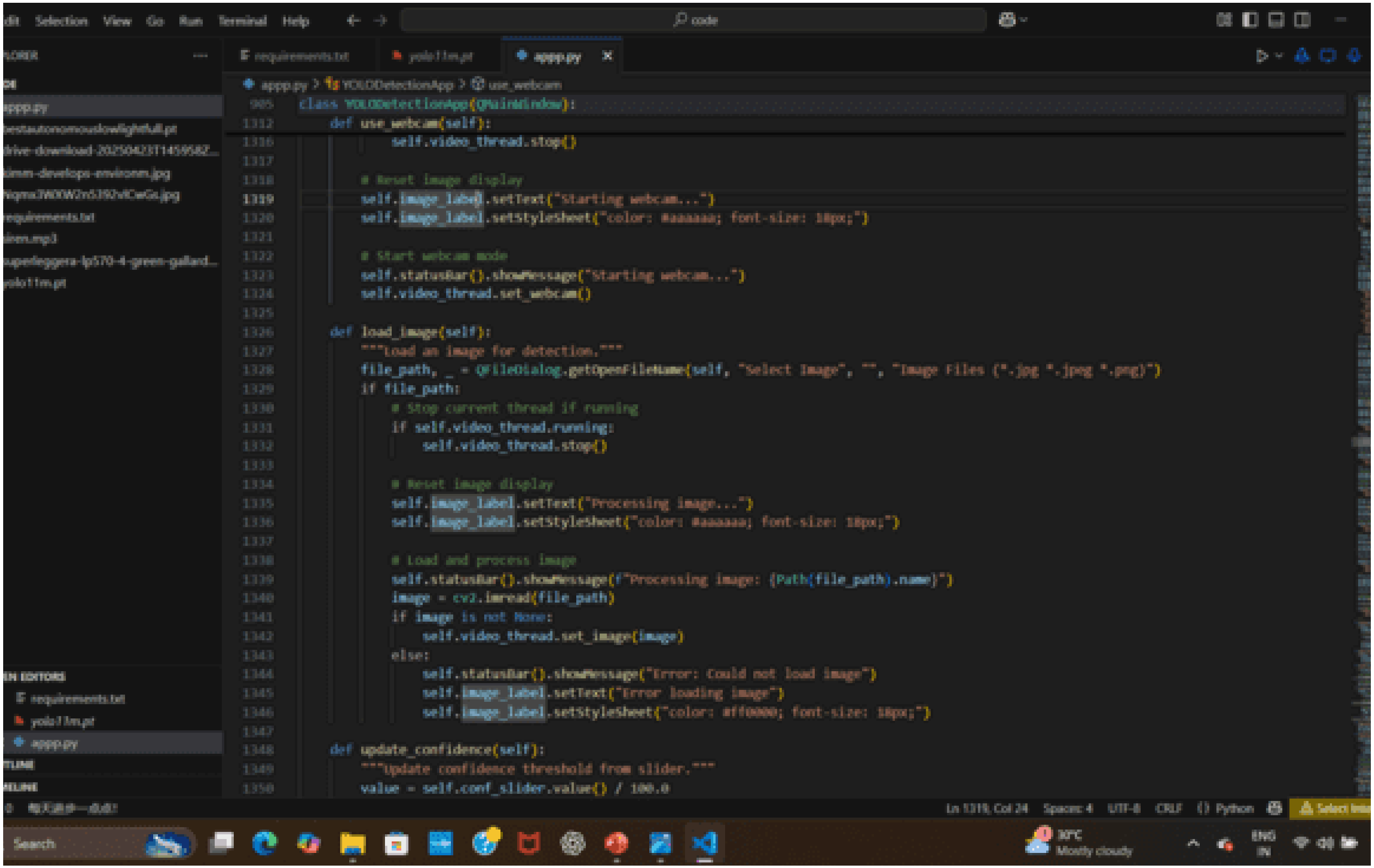
- **User Interface**: Build a Streamlit UI for users to upload images and interact with the model.
- **Prediction Display**: Display detection results, including object labels and bounding boxes.
- **Live Alerts**: Provide real-time alerts based on detected objects.
- **Integration with Backend**: Send user-uploaded images to the backend for inference and receive predictions.
- **Deployment**: Host the Streamlit app on cloud platforms (Heroku, AWS, etc.) for public access.

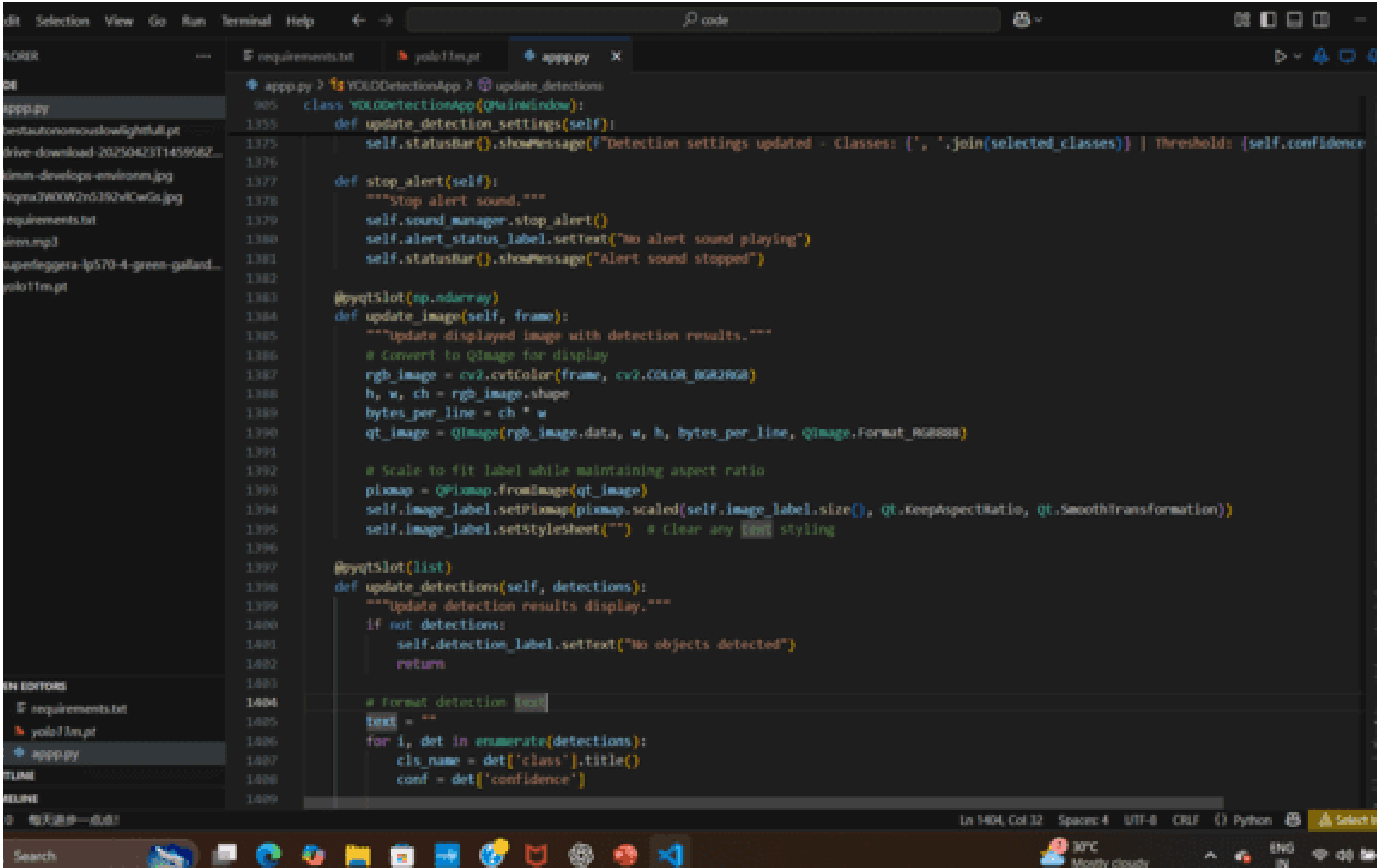
```python
class VideoThread(QThread):
    def run(self):
                    time.sleep(0.01)

                if cap.isOpened():
                    cap.release()

            elif self.mode == "image" and self.image is not None:
                start_time = time.time()
                processed_frame, detections = self.process_frame(self.image)
                process_time = time.time() - start_time

                self.update_frame.emit(processed_frame)
                self.update_detections.emit(detections)
                # For images, we'll use FPS signal to show processing time
                self.update_fps.emit(process_time)

                # After processing a single image, stop the thread
                self.running = False

    def process_frame(self, frame):
        """Process a single frame for object detection."""
        # Convert frame to RGB for YOLO
        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        # Run YOLOv8 inference
        results = self.model(frame_rgb, verbose=False)

        # Process results
        detections = []
        for r in results:
            boxes = r.boxes
            for box in boxes:
                # Get detection confidence
                conf = float(box.conf[0])
```



class YOLODetectionApp(QMainWindow):
def load_model(self):
"""Load YOLOv8 model from file."""
file_path, _ = QFileDialog.getOpenFileName(self, "Select YOLOv8 model", "", "PyTorch model (*.pt)")
if file_path:
self.model_path = file_path
self.model_path_label.setText(f"Model: {Path(file_path).name}")
self.statusBar().showMessage(f"Model loaded: {Path(file_path).name}")
# Stop current thread if running
if self.video_thread.running:
self.video_thread.stop()
# Create new thread with updated model
self.video_thread = VideoThread(self.model_path, self.target_classes, self.confidence_threshold)
self.video_thread.update_frame.connect(self.update_image)
self.video_thread.update_detections.connect(self.update_detections)
self.video_thread.update_fps.connect(self.update_performance)
def load_alert_sound(self):
"""Load custom alert sound."""
file_path, _ = QFileDialog.getOpenFileName(self, "Select Alert Sound", "", "Audio Files (*.mp3 *.wav)")
if file_path:
self.custom_alert_path = file_path
self.alert_path_label.setText(f"Alert: {Path(file_path).name}")
self.sound_manager.update_custom_alert(file_path)
self.statusBar().showMessage(f"Alert sound loaded: {Path(file_path).name}")
def use_webcam(self):
"""Switch to webcam input."""
# Stop current thread if running
if self.video_thread.running:
self.video_thread.stop()
# Reset image display
self.image_label.setText("Starting webcam...")

B.S. Abdur Rahman Crescent Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956
GST Road, Vandalur, Chennai 600 048



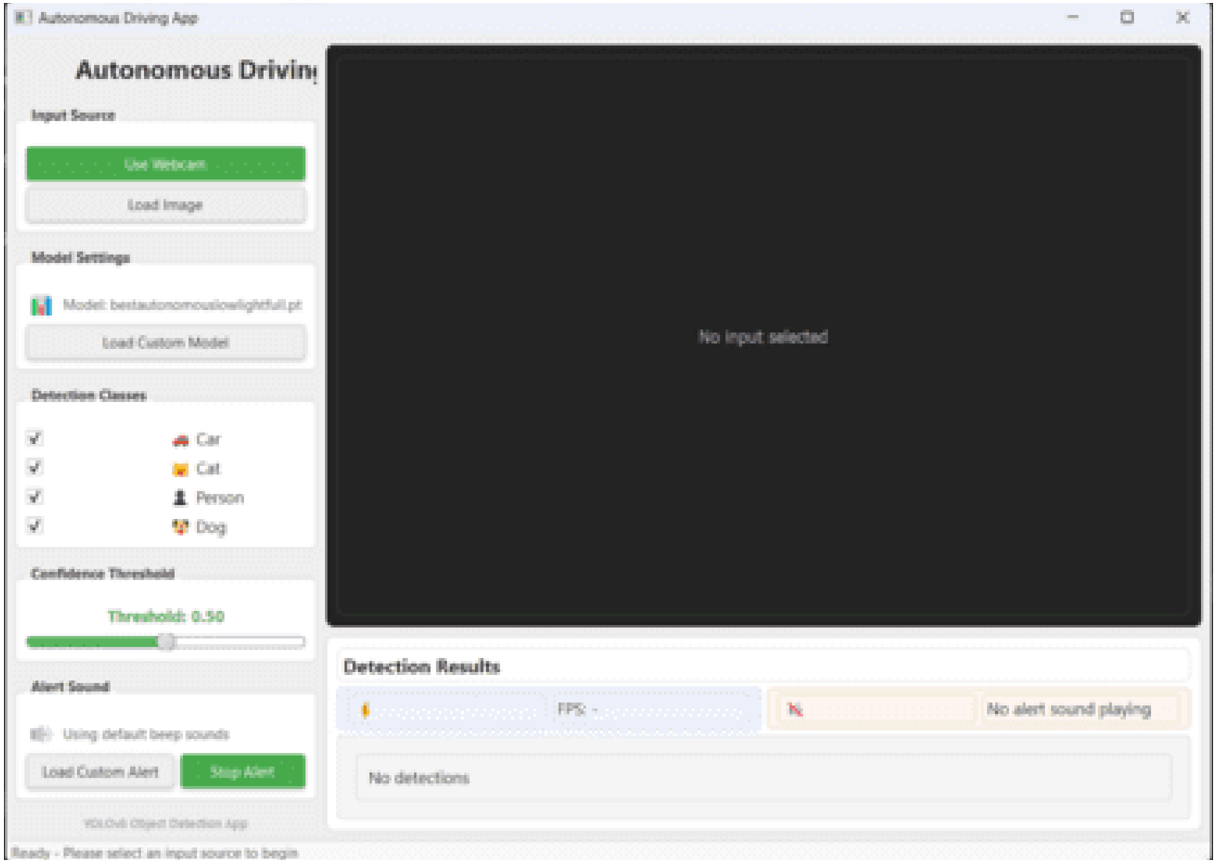


requirements.txt
yolo11m.pt
appp.py
appp.py > YOLODetectionApp > update_detections
class YOLODetectionApp(QMainWindow):
    def update_detection_settings(self):
        self.statusBar().showMessage(f"Detection settings updated - Classes: {', '.join(selected_classes)} | Threshold: {self.confidence
    def stop_alert(self):
        """Stop alert sound."""
        self.sound_manager.stop_alert()
        self.alert_status_label.setText("No alert sound playing")
        self.statusBar().showMessage("Alert sound stopped")
    @pyqtSlot(np.ndarray)
    def update_image(self, frame):
        """Update displayed image with detection results."""
        # Convert to QImage for display
        rgb_image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        h, w, ch = rgb_image.shape
        bytes_per_line = ch * w
        qt_image = QImage(rgb_image.data, w, h, bytes_per_line, QImage.Format_RGB888)
        # Scale to fit label while maintaining aspect ratio
        pixmap = QPixmap.fromImage(qt_image)
        self.image_label.setPixmap(pixmap.scaled(self.image_label.size(), Qt.KeepAspectRatio, Qt.SmoothTransformation))
        self.image_label.setStyleSheet("") # Clear any text styling
    @pyqtSlot(list)
    def update_detections(self, detections):
        """Update detection results display."""
        if not detections:
            self.detection_label.setText("No objects detected")
            return
        # Format detection text
        text = ""
        for i, det in enumerate(detections):
            cls_name = det['class'].title()
            conf = det['confidence']
Ln 1404, Col 32 Spaces: 4 UTF-8 CRLF Python

RESULT WITH ANALYSIS

STEP 1:

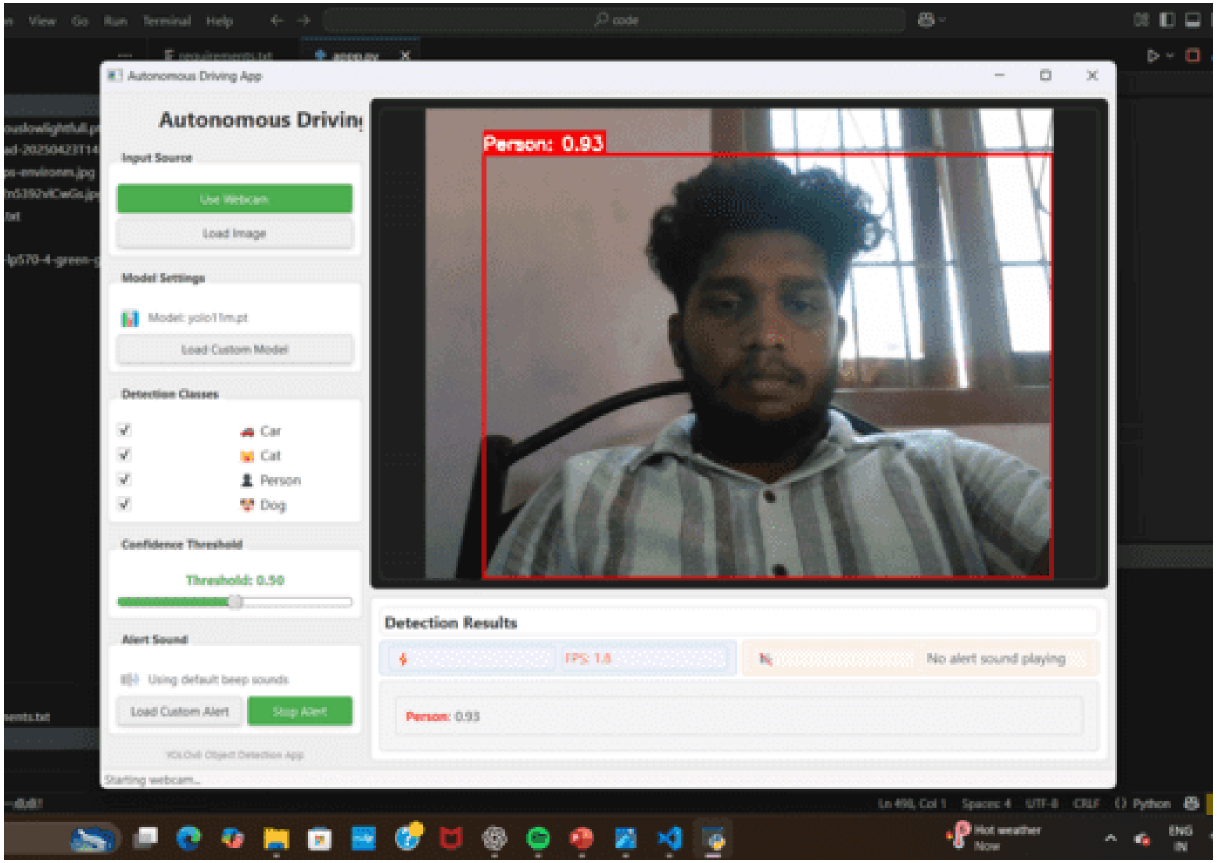
Run the code and it will open the webpage for user inter face



RESULT WITH ANALYSIS

STEP 2:

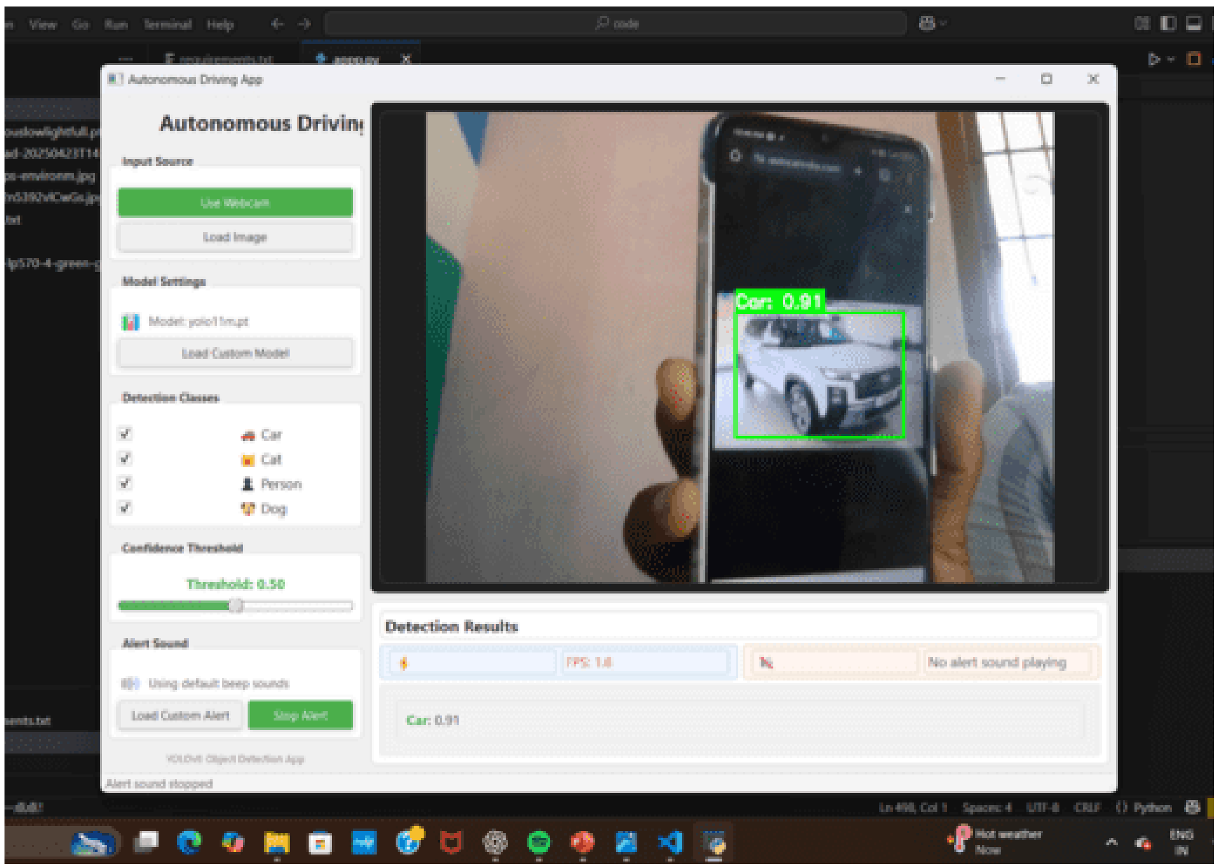
Click the use webpage to detect the object and in this set we detect the person



RESULT WITH ANALYSIS

STEP 3:

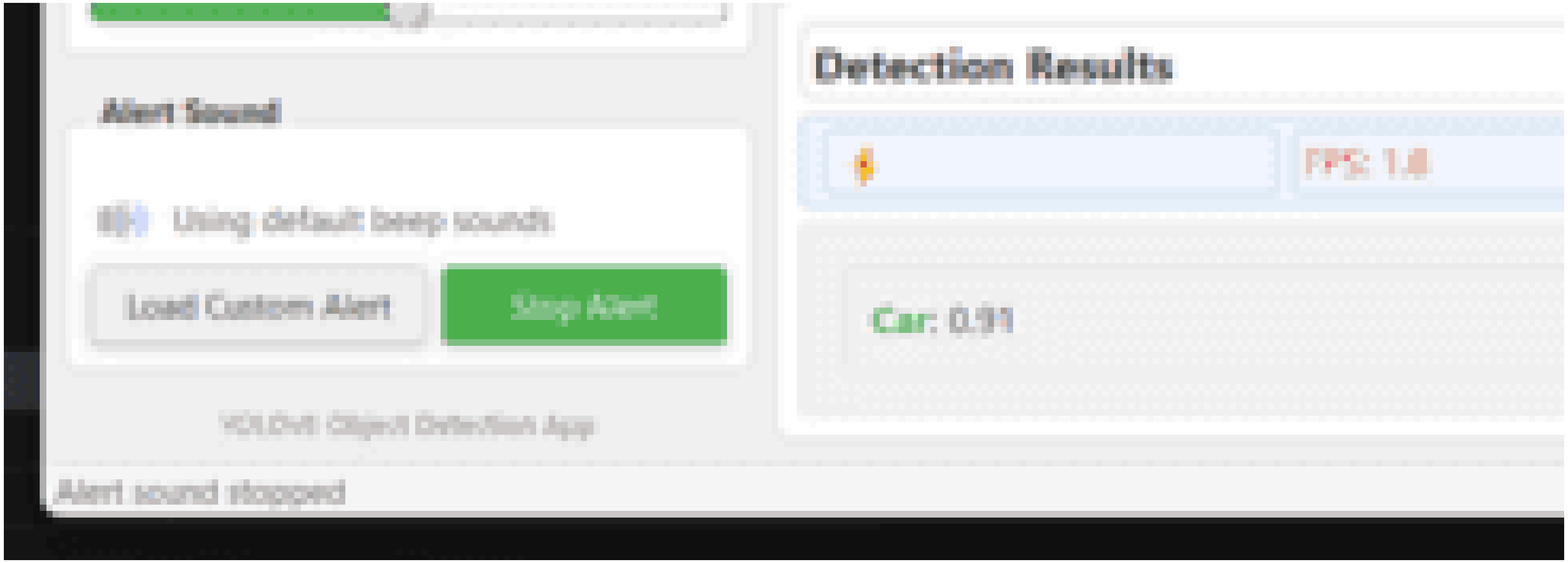
Click the use webpage to detect the object and in this detect the car etc.,, it can detect the dogs,cat,cow uing this for road safety.



RESULT WITH ANALYSIS

STEP 4:

In this we can able to stop the alert sound because of any detraction will get for the driver so we keep it if we can need on otherwise off



Detailed Algorithm for each module

YOLOv8 :

- YOLOv8 is the latest version of the YOLO (You Only Look Once) object detection algorithm. It is designed for real-time object detection, classification, and segmentation with improved accuracy, speed, and efficiency.
- Optimized for edge devices,object detection, classification, and segmentation.
- Good accuracy.

R-CNN (Region-based Convolutional Neural Network):

- It is an advanced deep learning model for object detection. It improves on previous versions (R-CNN, Fast R-CNN), making it faster and more efficient.

  - R-CNN is a high-accuracy object detection model.

  - Best for tasks where accuracy is more important than speed.

B.S. Abdur Rahman Crescent Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956
GST Road, Vandalur, Chennai 600 048

Conference:

Attended International Conference on Contemporary Trends in Advanced Computing Technologies (ICCTACT 2024)” ORGANIZED BY New Prince Shri Bhavani Arts and Science College on (15.03.2025)

GUKAN.S

SANJAY.G

KAUSHIK.A.S

Technology Certification

- NAME'S : Gukan.S , Sanjay.G , Kaushik.A.S
- COURSE COMPLETED IN Image Processing And Computer Vision With Python & Opencv DATE : In Udemy App On (5.3.2025)

udemy

CERTIFICATE OF COMPLETION

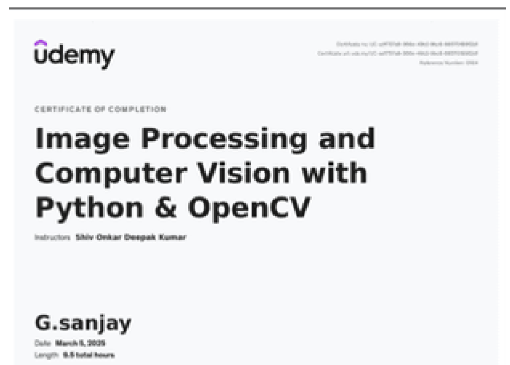
Image Processing and Computer Vision with Python & OpenCV

Instructors **Shiv Onkar Deepak Kumar**

Gukan S

Date **March 5, 2025**

Length **9.5 total hours**



udemy

CERTIFICATE OF COMPLETION

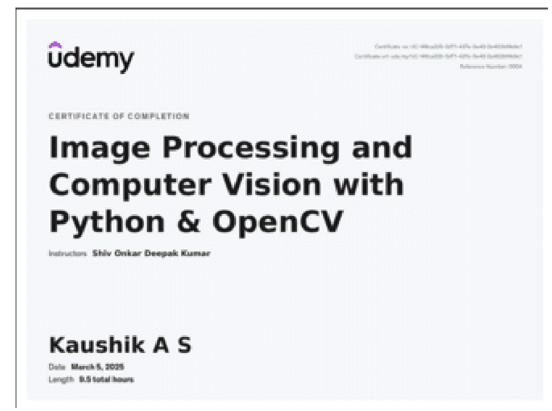
Image Processing and Computer Vision with Python & OpenCV

Instructors **Shiv Onkar Deepak Kumar**

G.sanjay

Date **March 5, 2025**

Length **9.5 total hours**



udemy

CERTIFICATE OF COMPLETION

Image Processing and Computer Vision with Python & OpenCV

Instructors **Shiv Onkar Deepak Kumar**

Kaushik A S

Date **March 5, 2025**

Length **9.5 total hours**

CONCLUSION

- The proposed system effectively enhances the detection of objects, such as pedestrians, animals, and vehicles, in low-light and nighttime conditions using a machine learning model, ensuring better road safety.
- By utilizing standard night-vision-enabled cameras rather than expensive thermal imaging cameras, the project offers a cost-effective alternative that makes object detection accessible for a wider range of users.
- The system is designed to process images in real-time, providing instant feedback to drivers, which is crucial for improving awareness and enabling faster reactions to potential hazards.
- The integration of the backend model with a Streamlit web application ensures that the system is easy to use, allowing users to simply upload images and receive predictions with clear, visual outputs.
- The project can be further enhanced by expanding its capabilities to handle video input, improving detection accuracy in more challenging environments, and integrating additional sensors for more robust real-time performance.

FUTURE SCOPE

1. **Improved Detection**: You could explore improving the accuracy and robustness of your detection system by adding more training data or integrating more advanced techniques like Transfer Learning or Ensemble Models.
2. **Real-time Processing**: Discuss the potential for enhancing the real-time performance of the system, perhaps through model optimization (like using smaller models or applying quantization techniques).
3. **Integration with Other Technologies**: You might consider integrating the system with a larger application, such as a security system that automatically sends alerts when dangerous objects (like a person or car) are detected.
4. **Scalability**: Consider how your system could scale, such as supporting more cameras or operating in different environments (e.g., daytime, different weather conditions).
5. **Enhanced User Interface**: If your system has a user interface, you could develop it further to include more detailed visualizations, real-time monitoring, or historical logs.

REFERENCES

1. Almujally, N. A., Qureshi, A. M., Alazeb, A., Rahman, H., Sadiq, T., Alonazi, M., ... & Jalal, A. (2024). A novel framework for vehicle detection and tracking in night ware surveillance systems. *Ieee Access*.
2. Murugan, R. A., & Sathyabama, B. (2023). Object detection for night surveillance using Ssan dataset based modified Yolo algorithm in wireless communication. *Wireless Personal Communications*, *128*(3), 1813-1826.
3. Raza, A., Chelloug, S. A., Alatiyyah, M. H., Jalal, A., & Park, J. (2023, January). Multiple pedestrian detection and tracking in night vision surveillance systems. In CMC (Vol. 75, pp. 3275-3289).
4. Nandhini, T. J., & Thinakaran, K. (2023, March). A Robust Framework for Traffic Object Detection using Intelligent Techniques. In *2023 9th International Conference on Electrical Energy Systems (ICEES)* (pp. 328-333). IEEE.
5. Tahir, N. U. A., Zhang, Z., Asim, M., Chen, J., & ELAffendi, M. (2024). Object detection in autonomous vehicles under adverse weather: A review of traditional and deep learning approaches. *Algorithms*, *17*(3), 103.
6. Alasmari, N., Alohali, M. A., Khalid, M., Almalki, N., Motwakel, A., Alsaid, M. I., ... & Alneil, A. A. (2023). Improved metaheuristics with deep learning based object detector for intelligent control in autonomous vehicles. *Computers and Electrical Engineering*, *108*, 108718.
7. Ristić-Durrant, D., Haseeb, M. A., Banić, M., Stamenković, D., Simonović, M., & Nikolić, D. (2022). SMART on-board multi-sensor obstacle detection system for improvement of rail transport safety. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, *236*(6), 623-636.
8. Shenoy, M. A., Betrabet, P. R., & NS, K. R. (2022, October). Helmet detection using machine learning approach. In *2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)* (pp. 1383-1388). IEEE.
9. Masud, U., Saeed, T., Malaikah, H. M., Islam, F. U., & Abbas, G. (2022). Smart assistive system for visually impaired people obstruction avoidance through object detection and classification. *IEEE access*, *10*, 13428-13441.
10. Lee, M. F. R., & Chen, Y. C. (2023). Artificial intelligence based object detection and tracking for a small underwater robot. *Processes*, *11*(2), 312.

Thank you!