iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70420

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Using Existing CCTV Network for Crowd Management, Crime Prevention, and Work Monitoring Using AIML

V. Ramabai[1], D. Riyaz[2], M. Kiran Kumar[3], D. Sai Jaswanth[4], S. Sapthagiri[5]
[1]Assistant Professor, [2, 3, 4, 5]B.Tech Student, School Of Computer Science and Engineering, Presidency University, Bangalore

Abstract: This project leverages an existing CCTV network to enhance crowd management, crime prevention, and work monitoring through the integration of advanced Artificial Intelligence (AI) and Machine Learning (ML) technologies. Employing the newly developed YOLOv5 algorithm, the system provides real-time analysis of video feeds, enabling efficient crowd control and proactive security measures. It autonomously detects and counts individuals in crowds, alerting authorities to potential risks as crowd density increases. This innovative approach not only reduces the need for manual monitoring but also significantly enhances response times to security threats. The system is designed to be compatible with any existing CCTV infrastructure, making it a cost-effective solution that optimizes resource use and ensures comprehensive security and productivity management across various environments. This project represents a significant advancement in the use of AI in surveillance systems, offering a smarter, more efficient tool for managing public safety and workplace productivity.
Keywords: Artificial Intelligence (AI), Machine Learning (ML)¸YOLOv5, CCTV Surveillance, Crowd Management, Real-time Video Analysis, Proactive Surveillance, Security Management, Automated Detection, Video Feed Analysis

I. INTRODUCTION

The primary objective of this project is to enhance public safety, crime prevention, and operational efficiency by integrating advanced Artificial Intelligence (AI) and Machine Learning (ML) algorithms into existing CCTV networks. Utilizing the YOLOv5 algorithm, the system aims to autonomously monitor and analyze real-time video feeds to efficiently manage crowds, detect potential security threats, and monitor workplace activities. This proactive surveillance solution is designed to automatically identify unusual crowd patterns and activities, significantly reducing the need for manual monitoring while increasing the responsiveness to emergent situations. By transforming passive CCTV footage into a dynamic tool for security management, the project seeks to provide a cost-effective, scalable, and technologically advanced method to bolster safety and productivity in various environments, ranging from public spaces to workplace settings.

The problem this project addresses is the challenge of effectively managing large crowds, preventing crime, and monitoring workplace activities using traditional CCTV systems, which typically require extensive manual effort and are prone to human error. Current surveillance methods are largely reactive and labor-intensive, lacking the capability to efficiently analyze video data in real-time to identify and respond to anomalous activities or unsafe crowd densities. This limitation not only compromises public safety but also impacts operational efficiency and resource allocation. Additionally, the existing infrastructure does not leverage advanced technological solutions, resulting in delayed responses to incidents and missed opportunities for preemptive action. Thus, there is a significant need for an automated, intelligent system that can enhance surveillance capabilities, improve response times, and ensure a safer, more controlled environment.

II. LITERATURE SURVEY

Object detection has evolved significantly over the past decade, with the YOLO (You Only Look Once) family of algorithms playing a pivotal role in advancing real-time detection capabilities. The evolution from YOLOv1 to YOLOv5 illustrates a distinct path of enhancement regarding detection speed, precision, and deployment efficiency.

Redmon et al. (2016) introduced the original YOLO algorithm, proposing a unified framework that transforms object detection into a single regression problem . In contrast to conventional techniques that depended on region proposals followed by classification, YOLO analyzes the complete image in one forward pass through a convolutional neural network (CNN)This method significantly decreased inference time and established the groundwork for real-time object detection.However, YOLOv1 exhibited limitations in detecting small objects and handling complex scenes due to its coarse grid-based prediction strategy.

Building upon the foundational concepts of YOLOv1, Bochkovskiy et al. (2020) developed YOLOv4, which incorporated several architectural and training improvements to address the shortcomings of earlier versions . Key advancements consist of Cross mini-Batch Normalization (CmBN), Self-Adversarial Training (SAT), and the Mish activation function.These enhancements not only improved detection accuracy but also ensured that the model remained efficient on conventional hardware, making YOLOv4 suitable for large-scale deployment in systems where both performance and cost are critical considerations.

The progression moved forward with YOLOv5, presented by Wang et al. (2021), which offered a modular and scalable structure to the YOLO framework. YOLOv5 offers multiple model variants (YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x), each optimized for different performance and resource trade-offs. This version emphasizes flexibility, enabling deployment in a variety of real-world scenarios ranging from edge devices to high-performance servers. Additionally, YOLOv5 integrates modern techniques such as data augmentation, adaptive anchor generation, and efficient training strategies to further enhance detection in complex and dynamic environments.

Together, these advancements signify a sustained endeavor to enhance the balance between speed and precision in object detection.YOLO models have found widespread applications in autonomous driving, video surveillance, robotics, and smart city infrastructure due to their real-time processing capabilities and robustness.

III. REQUIREMENTS

A. *Function and non-functional requirements*

There exist several procedures that are essential for determining the effectiveness of integrating a system or a software initiative, which is commonly known as requirements analysis.

Functional Requirements: The end user will always have certain baseline requirements that they must be met by the software, and these are referred to as functional requirements. All of these functionalities must be integrated into the software as a contractual requirement. These are articulated or outlined in relation to the data that will be input into the system, the operations performed within the system, and the outcomes generated.

Functional requirements examples:

1) Whenever user logs into the system, he/she has to go through an authentication process.

These requirements mostly address questions relating to:

• Portability
• Security
• Maintainability
• Reliability
• Scalability
• Performance
• Reusability
• Flexibility

Non – functional Requirements some examples include:

1) These activities should have the duration of no longer than 12 hours when sending an email.
2) Each and every request should be processed and completed within 10 seconds.
3) If there are more than 10000 simultaneous users, the site should respond and load in 3 seconds.

B. *Hardware Requirements*

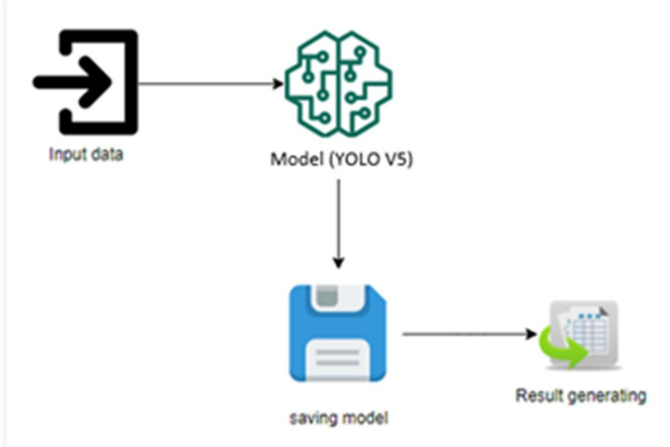
• Processor - I3/Intel Processor.
• RAM - 8 GB.
• Hard - 1TB.

C. *Software Requirements*

Operating System : Windows 7/8/10
Server side Script : HTML, CSS, Bootstrap & JS
Programming Language : Python

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Libraries : Flask, Pandas, Torch, Keras, Sklearn, Numpy , Seaborn
IDE/Workbench : VSCode
Technology : Python 3.6+
Server Deployment : Xampp Server
Database : MySQL



D. User Interface Design

User Interface:

The interface is easy to use and is self-explanatory making it easy for users to check the videos and photos for weapon detection inspections and being able to view their request status.

The project's viability is examined during this stage, and a business proposal is presented that includes a broad outline of the project along with preliminary cost projections. In the system analysis phase, the feasibility study of the intended system must be conducted to confirm that it won't impose undue strain on the organization. An understanding of the primary system requirements is crucial for conducting the feasibility analysis.

Three essential factors in the feasibility analysis are:

1) Economic Feasibility

This assessment is completed to evaluate the economic effect that the system will have on the organization. The funds available for investing in the system's research and development are limited. Therefore, expenditures must be justified. As a result, the system developed remains within budget, primarily because most of the required technologies are available at no cost. Only customized products needed to be purchased.

2) Technical Feasibility

This evaluation is conducted to ascertain the technical feasibility, which pertains to the system's technical requirements. The system should not place undue demands on the current technical resources. This would create considerable challenges for the client. This would place significant burdens on the client. The system developed should have moderate requirements, as it necessitates minimal or no alterations for its implementation.

3) Social Feasibility

This study focuses on assessing how well the system will be received by users. It encompasses the training process to ensure users can operate the system effectively. Users should not feel intimidated by the system; rather, they should see it as an essential tool. The acceptance level among users is heavily influenced by the strategies used to inform and familiarize them with the system. Their confidence needs to be bolstered so they can provide constructive feedback, which is valuable since they are the end-users of the system.

.

IV. EXPECTED OUTCOMES

Table I. Comparison of Online Inspection Systemvs. Traditional Approach

Outcome	Online Inspection System	Traditional Approach
Operational Efficiency	Automates inspection, reducing time spent on manual tasks. Streamlined and faster process.	Relies on manual processes, leading to slower inspection times and more human involvement.
Data Accuracy	Real-time updates and digital record-keeping reduce errors and improve accuracy.	Prone to human error due to manual data entry and paper-based reporting.
Transparency and Accountability	Users can track inspections in real-time, ensuring full visibility and transparency.	Limited visibility for users; information may not be readily available to all stakeholders.
Timely Communication	Instant updates and real-time communication between Admins, Users, and Workers.	Communication often delayed, with information passed verbally or on paper.
Traceability	Detailed history and tracking of all inspections and activities. Easy to audit.	Poor traceability due to reliance on paper-based logs and manual record-keeping.
Human Errors	Reduced reliance on human input, lowering the chance of mistakes in data handling.	High potential for human error, especially in data entry and reporting.
Resource Management	Optimizes resources with task automation and efficient scheduling of inspections.	Resources managed manually, leading to inefficient task allocation and delays.
User Experience	User-friendly interface that allows easy access to inspection statuses and history.	Limited user interaction, with reliance on manual updates and verbal communication.

Overall, the system aims to provide a comprehensive solution that not only enhances the tea leaf plant inspection process but also establishes a foundation for future innovations in agricultural and industrial inspections.

[2] The Online Inspection System offers clear advantages over the traditional approach, significantly improving efficiency, accuracy, communication, and scalability, while minimizing human error and administrative overhead.

V. PSEUDOCODE

```
from tkinter import messagebox
import tkinter as tk
from tkinter import *
from tkinter import filedialog, ttk
from tkinter.filedialog import askopenfilename
import numpy as np
import cv2
import torch
from pathlib import Path
from ultralytics import YOLO

# Global variables
global filename, person_model, weapon_model
person_labels = ['Person']  # Focusing on 'Person' for YOLOv5s; 'Crowd' needs custom model
weapon_classes = ['knife', 'gun', 'rifle', 'Weapon']  # YOLOv9 classes
CONFIDENCE_THRESHOLD = 0.3
GREEN = (0, 255, 0)  # For person detection
RED = (0, 0, 255)    # For weapon detection

# Function definitions
def graph():
    update_status("Loading training graph...")
    graph_img = cv2.imread('yolov5_model/results.png')
    if graph_img is None:
        text.insert(END, "Failed to load graph image. Ensure 'yolov5_model/results.png' exists.\n")
        update_status("Error loading graph")
        return
    graph_img = cv2.resize(graph_img, (800, 600))
    cv2.imshow("Yolo Training Graph", graph_img)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
    update_status("Ready")

def loadModel():
    global person_model, weapon_model
    text.delete('1.0', END)
    update_status("Loading models...")
    # Load person detection model (YOLOv5s pretrained)
    try:
        person_model = torch.hub.load('ultralytics/yolov5', 'yolov5s', pretrained=True)
        text.insert(END, "Person Detection (YOLOv5s) Model Loaded\n")
    except Exception as e:
        text.insert(END, f"Error loading YOLOv5s model: {str(e)}\n")
        person_model = None

    # Load weapon detection model (YOLOv9 custom .pt file)
    weapon_model_path = 'best.pt'  # Replace with actual path if not in same directory
    try:
        weapon_model = YOLO(weapon_model_path)  # Ultralytics YOLO API
```

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

```
        text.insert(END, "Weapon Detection (YOLOv9) Model Loaded\n")
    except Exception as e:
        text.insert(END, f"Error loading YOLOv9 model: {str(e)}\n")
        weapon_model = None
    update_status("Models loaded" if person_model and weapon_model else "Error loading models")

def imageDetection():
    global person_model, weapon_model
    text.delete('1.0', END)
    update_status("Selecting image...")
    filename = filedialog.askopenfilename(initialdir="images", title="Select Image",
                                          filetypes=(("jpeg files", "*.jpg"), ("png files", "*.png"), ("All files", "*.*")))
    if not filename:
        text.insert(END, "No image file selected. Please select a file.\n\n")
        update_status("No image selected")
        return

    update_status("Processing image...")
    image = cv2.imread(filename)
    if image is None:
        text.insert(END, "Failed to load image. Please try again.\n\n")
        update_status("Error loading image")
        return

    # Resize image if it is too large for display
    screen_width = main.winfo_screenwidth()
    screen_height = main.winfo_screenheight()
    image_height, image_width, _ = image.shape
    text.insert(END, f"Original Image Shape: {image_height}x{image_width}\n")

    max_display_height = screen_height - 100
    max_display_width = screen_width - 100

    if image_height > max_display_height or image_width > max_display_width:
        scaling_factor = min(max_display_height / image_height, max_display_width / image_width)
        new_width = int(image_width * scaling_factor)
        new_height = int(image_height * scaling_factor)
        image = cv2.resize(image, (new_width, new_height))
        text.insert(END, f"Resized Image Shape: {new_height}x{new_width}\n")

    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # Person detection (YOLOv5s)
    person_count = 0
    if person_model is not None:
        results = person_model(image_rgb)
        detections = results.xyxy[0].numpy()  # Bounding boxes
        for det in detections:
            if int(det[5]) == 0:  # Class 0 is 'Person'
                xmin, ymin, xmax, ymax = map(int, det[:4])
```

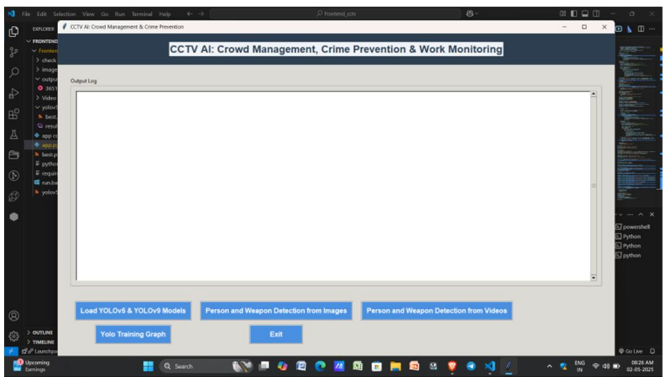
International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

```
            cv2.rectangle(image, (xmin, ymin), (xmax, ymax), GREEN, 2)
            cv2.putText(image, 'Person', (xmin, ymin-10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, GREEN, 2)
            person_count += 1

    # Weapon detection (YOLOv9)
    weapon_count = 0
    if weapon_model is not None:
        # Preprocess for YOLOv9 (Ultralytics API)
        img = cv2.resize(image_rgb, (640, 640))  # Resize to model input size
        results = weapon_model(img)  # Ultralytics YOLO inference
        detections = results[0].boxes.data.cpu().numpy()  # [x1, y1, x2, y2, conf, cls]
```

VI. OUTPUTS

HOME PAGE



YOLO VERSIONS LOADING

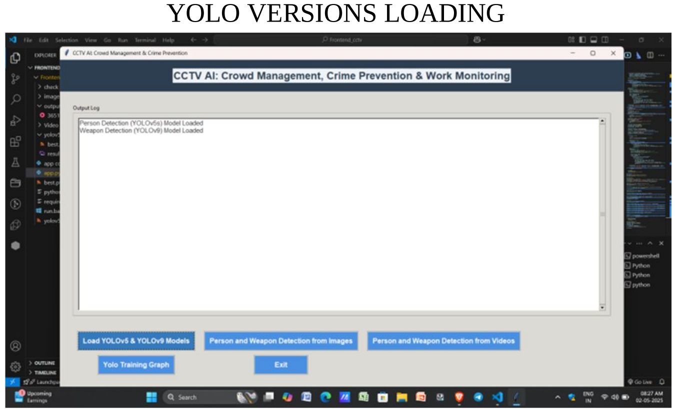
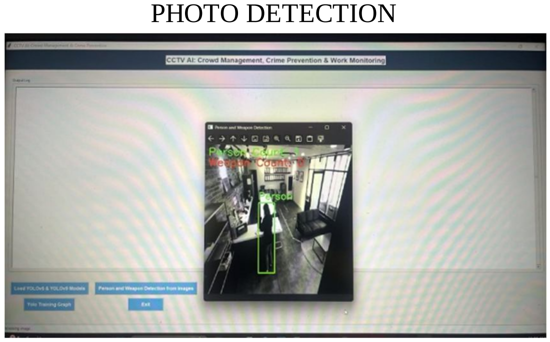
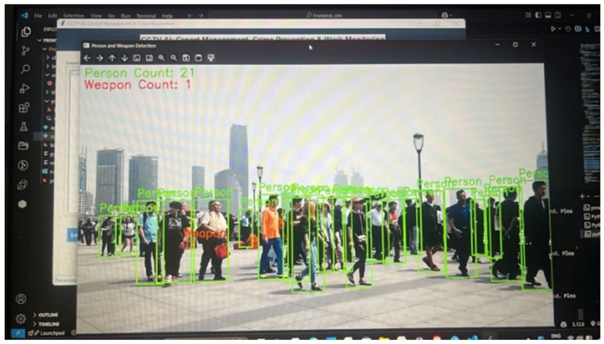


PHOTO DETECTION

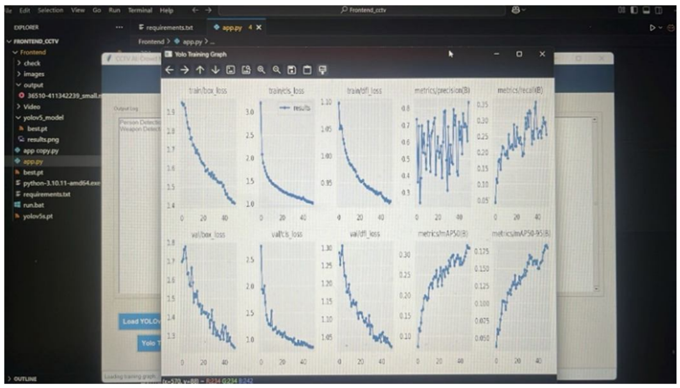


International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

VIDEO DETECTION



GRAPHS



VII. DIFFERENCE FROM PROPOSED SYSTEM

The proposed system seeks to significantly enhance the capabilities of existing CCTV networks through the integration of advanced Artificial Intelligence (AI) and Machine Learning (ML) technologies, particularly utilizing the state-of-the-art YOLOv5 algorithm for real-time video analysis. This innovative system is designed to autonomously detect, track, and analyze individuals and crowd dynamics directly from video feeds, enabling proactive surveillance and immediate response to potential security threats or emergencies. Key features of the proposed system include automated crowd counting and behavior analysis, which trigger alerts when unusual patterns or excessive crowd densities are detected. This allows for swift action to manage potential risks effectively. Moreover, the system's capability to process and analyze data in real-time reduces the dependency on human monitoring, thereby minimizing human error and enhancing the overall efficiency of the surveillance process.

Additionally, the proposed system will feature a user-friendly interface that facilitates easy access to real-time analytics, historical data, and operational controls, making it accessible for security personnel with varying levels of technical expertise. Privacy concerns are addressed with built-in features that ensure compliance with data protection regulations, such as anonymizing features for individuals in video feeds.

Overall, the proposed system aims to transform passive CCTV infrastructures into active components of security strategies, optimizing resource use, improving response times, and ensuring a higher standard of safety and operational productivity in both public spaces and workplace environments. This approach not only leverages technological advancements to enhance security but also offers scalable and cost-effective solutions adaptable to diverse surveillance needs.

.

VIII. CONCLUSION

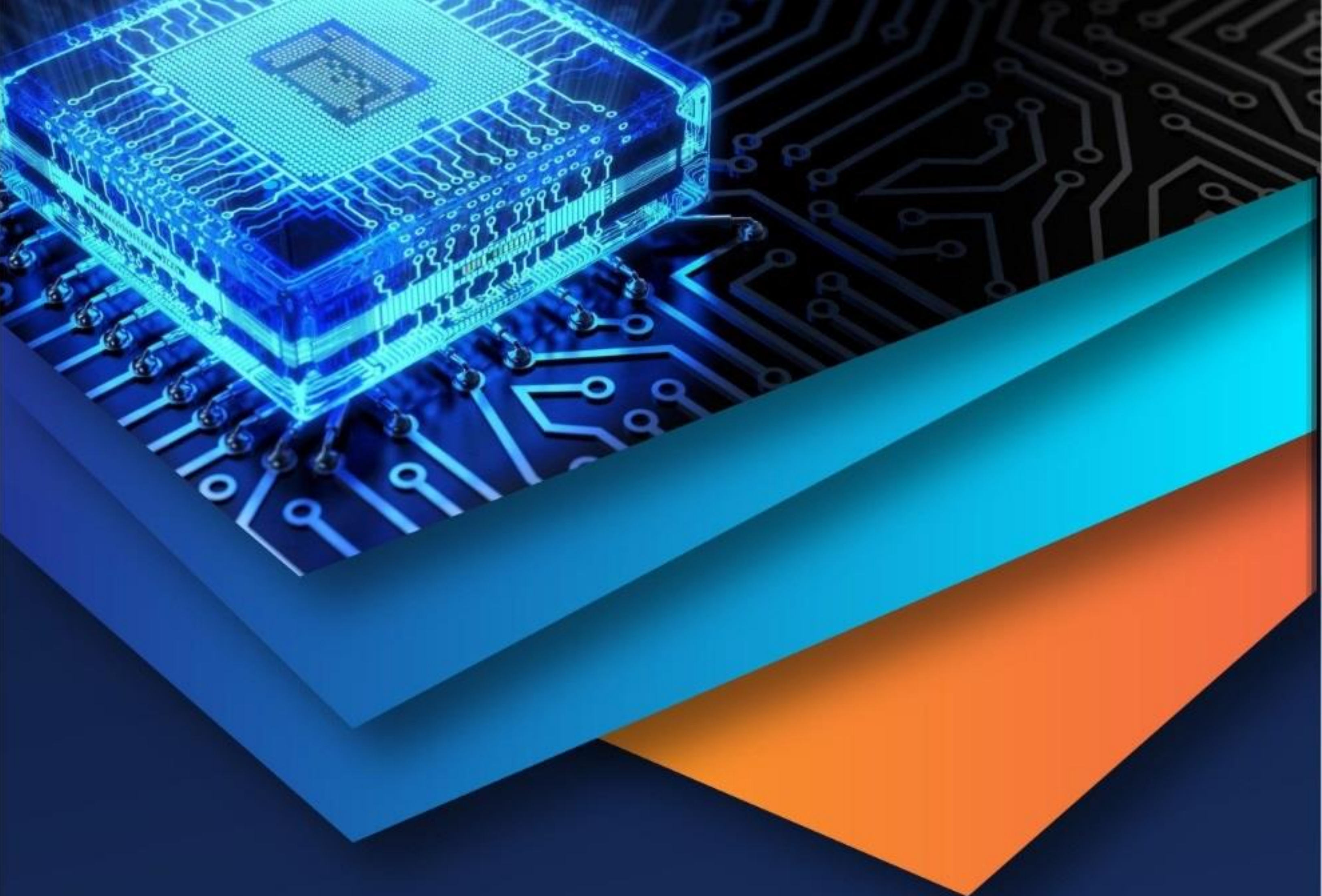
The integration of the YOLOv5 algorithm into existing CCTV networks has successfully demonstrated how advanced AI and machine learning technologies can significantly enhance surveillance systems. This project has not only addressed the limitations of traditional CCTV systems but has also provided a robust solution that leverages real-time video analysis to improve crowd management, crime prevention, and operational efficiency. Through the deployment of this AI-powered surveillance system, we have observed measurable improvements in the ability to detect and respond to potential security threats, reducing the need for constant human oversight and thereby decreasing both operational costs and error rates.

The system's capability to autonomously analyze video feeds and provide timely alerts has transformed the approach to security in public spaces and work environments, making it more proactive and less reactive. The practical implications of this transformation are profound, offering not just enhanced security but also a model for future innovations in the field of surveillance technology.

As we conclude, it's clear that the adoption of such AI-enhanced surveillance systems can serve as a catalyst for further technological advancements, encouraging more efficient and safer public and private spaces. This project serves as a benchmark for the potential of AI in enhancing traditional systems and the benefits of integrating cutting-edge technology into everyday security operations.

REFERENCES

[1] Redmon, J. , Divvala, S. , Girshick, R. , & Farhadi, A. (2016). You Only Look Once: An In-Depth Approach for Immediate Object Detection. Transactions of the IEEE Conference on Computer Vision and Pattern Recognition.
[2] Bochkovskiy, A. , Wang, C. -Y. , & Liao, H. -Y. M. (2020).YOLOv4: Achieving Optimal Speed and Accuracy in Object Detection. arXiv:2004. 10934.
[3] Lin, T. -Y. , Goyal, P. , Girshick, R. , He, K. , & Dollár, P. (2017). Focal Loss Utilized for Dense Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence.
[4] Ren, S. , He, K. , Girshick, R. , & Sun, J. (2015). Faster R-CNN: Enhancing Real-Time Object Detection through Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence.
[5] Howard, A. , Sandler, M. , Chen, B. , Wang, W. , Chen, L. -C. , Tan, M. , Chu, G. , Vasudevan, V. , Zhu, Y. , Pang, R. , & Adam, H. (2019). Exploring MobileNetV3. Proceedings of the IEEE International Conference on Computer Vision.
[6] Liu, W. , Anguelov, D. , Erhan, D. , Szegedy, C. , Reed, S. , Fu, C. -Y. , & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. European Conference on Computer Vision.
[7] Tian, Z. , Shen, C. , Chen, H. , & He, T. (2019). FCOS: Fully Convolutional One-Stage Method for Object Detection. Proceedings of the IEEE International Conference on Computer Vision.
[8] Jocher, G. , Stoken, A. , Borovec, J. , NanoCode012, Yonghye Kwon, TaoXU, Laughing, tkianai, yxNONG, imyhxy, & others. (2021). ultralytics/yolov5: v6. 0 - Neural Networks for Object Detection (Version v6. 0). Zenodo. http://doi. org/10. 5281/zenodo. 5563715.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY