IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁘ Peer-reviewed & Refereed journal ⁘ Vol. 11, Issue 6, June 2024

DOI: 10.17148/IARJSET.2024.11646

Smart Surveillance System Using Computer Vision

Prathik Achari[1], Sanika Dukle[2], Jayant Apte[3], A.U. Bapat[4]

Student, Department of Computer Engineering, Goa College of Engineering, Goa, India[1]

Student, Department of Computer Engineering, Goa College of Engineering, Goa, India[2]

Student, Department of Computer Engineering, Goa College of Engineering, Goa, India[3]

Professor, Department of Computer Engineering, Goa College of Engineering, Goa, India[4]

Abstract: In modern times, security and protection have become paramount concerns. This research paper introduces an innovative approach to fortify security in premises through a sophisticated smart surveillance system. Supported by a Django backend and empowered by various trained models, the system enables real-time monitoring of entries and exits, automated counting, and anomaly detection. It aims to revolutionize security and safety across diverse environments by automating tracking of individuals and objects, monitoring for suspicious activities, and early detection of fire hazards. The paper discusses the current state of technology in smart surveillance systems, covering advancements, challenges, and demerits. It outlines the objectives, motivation, and challenges of the project, emphasizing the importance of algorithm selection and fine-tuning, real-time processing, accuracy and reliability, adaptability to environmental changes, and addressing privacy concerns. The abstract underscores the project's commitment to delivering a comprehensive solution that leverages technology to enhance accuracy, efficiency, and security in premises.

Keywords: Computer Vision, Image Processing, YOLOv3-Tiny, Real-time object detection, Convolutional Neural Networks (CNNs), Non-Maximum Suppression (NMS), Structural Similarity Index (SSIM), Haar Cascade Classifier, classifier, Anomaly Detection, Object Recognition, Emergency Response, Fire Detection, Object Monitoring, Surveillance Technology.

I. INTRODUCTION

The intersection of computer vision and deep learning has revolutionized the capabilities of object detection and tracking systems. The ability to automatically identify and follow specific objects within image and video data has opened up expansive possibilities for applications ranging from security and surveillance to retail analytics and crowd management. This re- search paper delves into the analysis of an existing code implementation designed for" IN and OUT" detection, a system specializing in tracking the movement of individuals entering or exiting a designated area.

Monitoring the flow of people holds immense practical value across various scenarios. Security systems rely on such technology for real-time alerts and the identification of potentially anomalous behavior. In public spaces, including retail environments, transportation hubs, and event venues, crowd management can be vastly enhanced through the analysis of footfall patterns derived from in/out detection. These systems enable insights into customer behavior, optimization of space utilization, and the streamlining of resource allocation. Going further, automated tracking of individuals entering and exiting buildings can provide valuable data for smart building management, aiding in occupancy tracking and contributing to energy efficiency initiatives.

The code under investigation leverages a deep learning framework to detect objects within video frames, likely utilizing a variant of the renowned YOLO (You Only Look Once) object detection architecture. YOLO models are distinguished by their superior speed and real-time processing capabilities. Specifically, the configuration file suggests the use of YOLOv3-tiny, a more compact version of YOLO designed for efficiency and deployment on less powerful hardware. To track detected objects across frames and determine their directionality, the code implements a tracking mechanism within a continuous video processing loop. Tracking is a central component of this system, as simple object detection alone would not distinguish between an individual entering or leaving a defined area.

This paper embarks on a comprehensive exploration of the code's operational principles, examining the employed object detection techniques, motion tracking algorithms, and the system's potential real-world applications.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 11, Issue 6, June 2024

DOI: 10.17148/IARJSET.2024.11646

This foundation will support a subsequent in-depth analysis of the code's logic, specifically the mechanisms that enable its IN/OUT detection functionality. In addition to technical analysis, this paper will critically consider the strengths, potential limitations, and ethical dimensions of the system. We will conclude by outlining prospective future enhancements, paving the way for further development and optimization of such technology.

II. REQUIREMENTS

The successful implementation of the proposed system hinges upon meticulous adherence to both hardware and software prerequisites. These requirements are meticulously outlined to ensure seamless functionality and compatibility across diverse computing environments.

A. Software Requirements

To run the Smart Surveillance System software, the following are the required software components:

- Operating System: Windows/Linux/Mac
- Python 3
- OpenCV
- NumPy
- Integrated Development Environment (IDE): Visual Studio Code

B. Hardware Requirements

To run the Smart Surveillance System, the following hardware components are needed:

- Working PC
- Webcam with Drivers Installed
- Flashlight/LED (if using this at night)

III. METHODOLOGY

The scope of our project encompasses the development and implementation of a sophisticated smart CCTV system, aimed at transcending the conventional capabilities of traditional CCTV surveillance. By integrating cutting-edge technologies such as computer vision, machine learning, and real-time data analytics, our system offers an array of advanced features tailored to enhance surveillance efficacy and situational awareness. Below are the key features encapsulated within the scope of our project:

A. In and Out Count of People

By integrating motion detection and trajectory analysis capabilities, the system accurately counts and tracks the movement of individuals entering and exiting designated zones, facilitating crowd management and occupancy monitoring.

YOLOv3-Tiny Architecture:

The YOLO (You Only Look Once) family of object detectors are renowned for their single-stage design and real-time performance capabilities. YOLOv3- Tiny builds upon these foundations as a streamlined and exceptionally efficient variant of the full-fledged YOLOv3 model. Let's delve into its key architectural elements:

- Simplified Backbone: YOLOv3-Tiny employs a simplified backbone network inspired by DarkNet-19 for feature extraction. This backbone consists of a series of convolutional layers with interspersed max-pooling layers, designed to efficiently capture image features at various levels of abstraction. This reduction in network complexity directly translates to decreased computational requirements and sometimes faster inference times.

- Multi-Scale Predictions: Similar to YOLOv3, YOLOv3-Tiny predicts bounding boxes at three different scales. The network outputs feature maps at varying resolutions, allowing the model to detect both large and small objects more effectively. This multi-scale approach is essential in real-world scenarios where objects appear in a wide range of sizes.

- Anchor Boxes: Like other YOLO variants, YOLOv3- Tiny relies on the concept of anchor boxes – a set of predefined bounding boxes with varying aspect ratios and sizes. These anchor boxes serve as references to aid the model in predicting accurate object boundaries.

- Trade-offs: The emphasis on speed inevitably comes with some compromises. While remarkably efficient, YOLOv3-Tiny might exhibit a slight decline in accuracy, particularly in complex scenes with occluded objects or when detecting very small objects.

- Weights File (.weights): This file embodies the knowledge the model acquired during training. It contains the learned values for all the weights (parameters) within the convolutional layers of the neural network.

- Configuration File (.cfg): This file defines the model's architecture. It specifies the types of layers (convolutional, max-pooling, etc.), their arrangement, how they are connected, and other hyperparameters.

- Loading Process (OpenCV): OpenCV's cv2.dnn.readNet() or similar functions load the weights and configuration files, enabling you to instantiate the YOLOv3-Tiny model within your code.

- Non-Maximum Suppression (NMS)

Non-Maximum Suppression (NMS) is an indispensable post-processing step in object detection systems like YOLO, where the model generates a multitude of potential bounding boxes. The goal of NMS is to eliminate redundant detections and ensure that each true object in the image is represented by a single, well-localized bounding box. Here's a more detailed breakdown of the process:

Sorting by Confidence: The algorithm begins by sorting all the predicted bounding boxes according to their associated confidence scores. The confidence score reflects the model's belief in both the presence of an object within the box and the accuracy of its predicted location.

Selecting the Highest and Calculating Overlap: The box with the highest confidence score is selected as a potential " true" detection. The Intersection over Union (IoU) metric is then used to quantify the degree of overlap between this selected box and all other remaining bounding boxes. IoU calculates the area of intersection between two boxes divided by the area of their union.

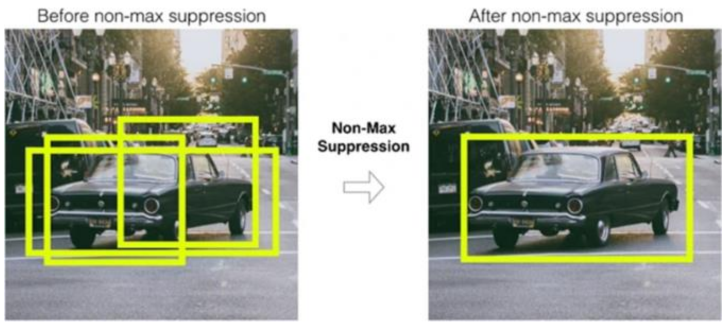

Fig. 1: The effect of Non-Max Suppression in Object Detection using YOLO

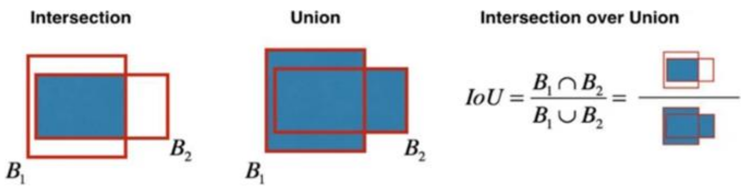

Fig. 2: Computational Schematic of Intersection over Union (IoU)

- Suppression Based on Threshold: Bounding boxes having an IoU with the selected box higher than a predetermined threshold are considered to be highly overlapping and, therefore, likely duplicates. These overlapping boxes are suppressed, essentially removing them from consideration.

- Why is NMS Important?

Non-Maximum Suppression (NMS) plays a vital role in object detection by addressing the issue of redundant detections. Without NMS, a single object could be surrounded by multiple overlapping bounding boxes, leading to inaccurate object counts and creating problems for tasks that rely on object detection results. NMS also enhances localization ac- curacy;

by suppressing less confident, overlapping boxes, it aids in pinpointing the most likely and well- defined bounding box for each detected object. Furthermore, NMS is essential for maintaining real-time performance. Its efficient process keeps the number of boxes that need further analysis manageable, which is particularly important in systems where speed is a critical factor.

B. Object Monitoring

Through sophisticated object detection and tracking algorithms, the system enables real-time monitoring and analysis of objects within the surveillance environment, aiding in the identification of suspicious activities and intrusions.

Object monitoring enhances security in smart surveillance systems by continuously analyzing specific object movement. Leveraging the Structural Similarity Index (SSIM), the system distinguishes significant scene changes, reducing false alarms and optimizing resource allocation. Real-time awareness is facilitated through visual monitoring, allowing quick decision-making. The adaptability to gradual scene changes ensures effectiveness in dynamic environments. Object monitoring contributes to efficient event logging, post-event analysis, and customizable monitoring strategies, enhancing overall surveillance system functionality.

Structural Similarity Index Metrics (SSIM):

The object monitoring process starts with initialization, where a threshold for the Structural Similarity Index (SSIM) change is set to detect significant differences. A reference frame is established for comparison, and an alert flag is configured to notify upon detecting abnormal behavior. In the main loop, each frame is captured and SSIM is calculated between the current frame and the reference frame. If the SSIM falls below the threshold, indicating a significant change, an alert is raised. The system displays the current frame for monitoring and updates the reference frame periodically to adapt to scene changes. Exit conditions include user input to exit the application and handling the end of the video stream.

The Structural Similarity Index (SSIM) quantifies the similarity between two images, assessing the perceived quality compared to a reference image. Commonly used in image processing and computer vision, SSIM aids in tasks like im- age quality assessment and comparison, providing a standardized measure for evaluating visual similarity.

The SSIM metric takes into account three key components of an image: luminance, contrast, and structure. The formula for SSIM is based on these components and is designed to produce a value be- tween -1 and 1, where: A value of 1 indicates perfect similarity (no distortion). A value of -1 indicates complete dissimilarity.

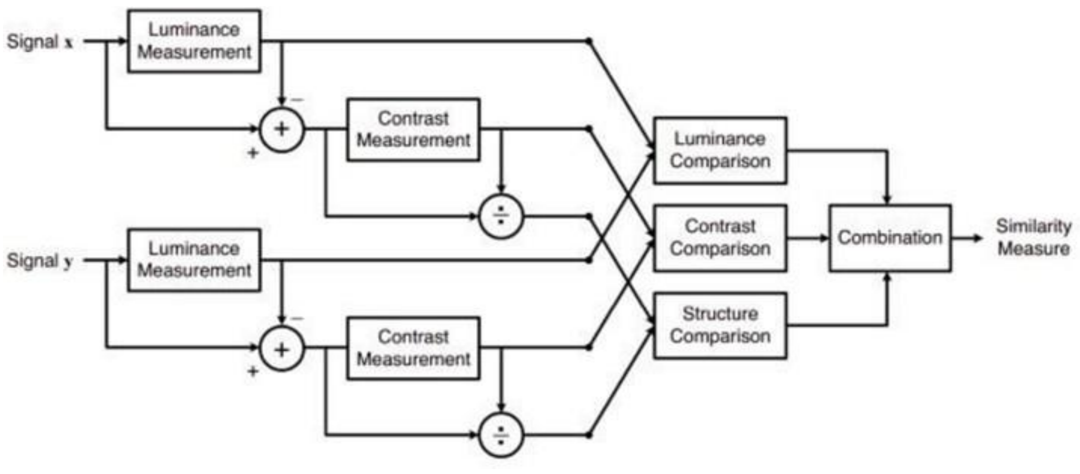

Fig. 3: Structural Similarity Index Metrics Model

C. Fire Detection

Leveraging computer vision, image processing techniques and Haar classifier the system can detect and alert authorities to the presence of flames within the surveillance area, facilitating prompt response and mitigating fire-related risks.

Fire detection is crucial across various domains, including surveillance systems and safety protocols, necessitating the application of advanced computer vision techniques for accurate and timely detection. In settings such as industrial environments or commercial establishments, the swift identification of fires is paramount for ensuring personnel safety, protecting valuable assets, and averting potential disasters.

DOI: 10.17148/IARJSET.2024.11646

Leveraging sophisticated algorithms like the Haar Cascade Classifier within surveillance systems brings several benefits, including rapid response capabilities, minimized downtime, and enhanced overall safety measures.

Haar Cascade Classifier:
The Haar Cascade Classifier is an object detection method based on Haar-like features, combined with a classifier that strengthens the cascade over time. This method has been successfully applied in numerous object detection applications, demonstrating its versatility and efficiency.

Haar Cascade is a feature-based object detection algorithm to detect objects from images. A cascade function is trained on lots of positive and negative images for detection. The algorithm does not require extensive computation and can run in real-time. Haar cascade uses the cascade function and cascading window. It tries to calculate features for every window and classify positive and negative. If the window could be a part of an object, then positive, else, negative.

To detect fire using our classifier, it requires a prior knowledge base to identify weather the object to be detected is fire. For this it is trained on a dataset using images. Positive images and Negative images are used for training purpose where, Positive images contain the object to be detected (Fire in our system) and Negative images contain objects other than the one we want to detect. We train these images and create a dataset file in .xml extension which will be later used by the Haar Cascade Classifier.

The rectangle on the left is a sample representation of an image with pixel values 0.0 to 1.0. The rectangle at the center is a haar kernel which has all the light pixels on the left and all the dark pixels on the right. The haar calculation is done by finding out the difference of the average of the pixel values at the darker region and the average of the pixel values at the lighter region. If the difference is close to 1, then there is an edge detected by the haar feature. These features on the image makes it easy to find out areas where there is a sudden change in the intensities of the pixels.

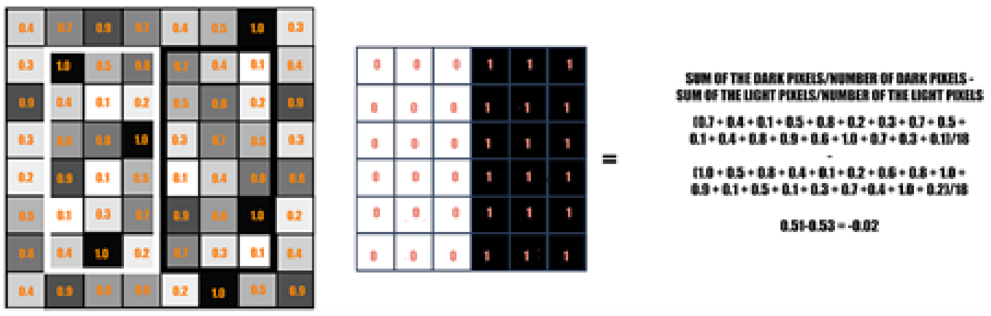

Fig. 4: Haar Cascade Classifier Calculation

IV. RESULTS

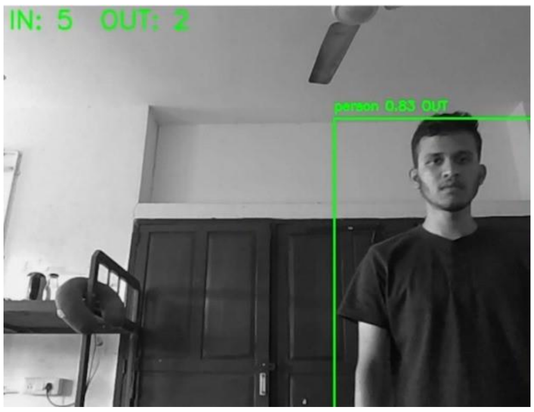

Fig. 5: In and Out Count of People

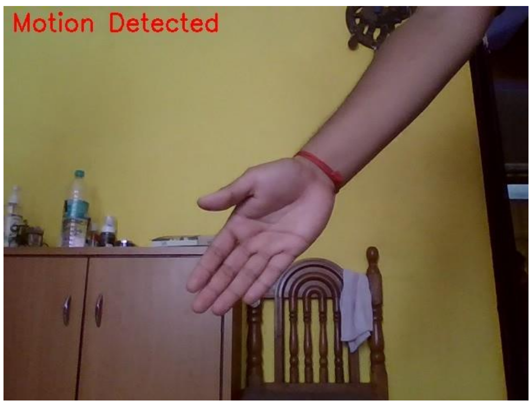

Fig. 6: Object Monitoring

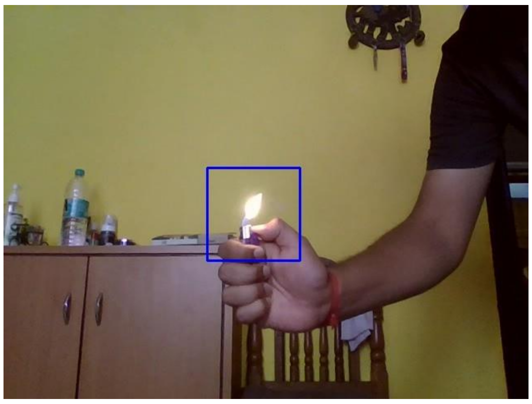

Fig. 7: Fire Detection

V. CONCLUSION

In conclusion, our research paper has explored the development and implementation of a comprehensive smart surveillance system endowed with advanced features tailored to enhance security, safety, and operational efficiency. Through the integration of functionalities such as in and out counting of people, object monitoring, and fire detection, our system rep- resents a significant advancement in the realm of surveillance technology. By leveraging cutting-edge computer vision techniques and deep learning algorithms, we have demonstrated the potential of our system to address critical challenges in diverse do- mains, from crowd management and asset protection to emergency response and disaster prevention.

The deployment of our smart surveillance system offers tangible benefits across various sectors, empowering businesses, organizations, and communities to proactively manage security threats, optimize resource utilization, and mitigate operational risks. Moreover, the scalability and adaptability of our system enable its seamless integration into existing infrastructure, facilitating widespread adoption and deployment across different environments and use cases.

VI. FUTURE SCOPE

The project lays a strong foundation for further advancements and enhancements in smart surveillance technology. Future iterations of the system can explore the integration of machine learning models to improve object recognition accuracy and adapt to dynamic environments. Additionally, the incorporation of edge computing capabilities can enhance real-time processing and reduce latency. The scalability of the system can be extended to cover larger areas and integrate with other IoT devices for a more comprehensive surveillance network.

Furthermore, the system can be enhanced with additional features such as facial recognition, anomaly detection, and predictive analytics for proactive security measures. Integration with cloud- based platforms can enable remote monitoring and data storage, facilitating accessibility and analysis. Collaboration with law enforcement agencies and emergency services can also be explored to ensure a swift response to potential security threats.

REFERENCES

[1]. Diwan T., Anirudh G. & Tembhurne J.V. “Object detection using YOLO: challenges, architectural successors, datasets and applications”. Multimed Tools Appl 82, 9243–9275 (2023). https://doi.org/10.1007/s11042-022-13644-y.

[2]. Kundu R., "YOLO: Algorithm for Object Detection Explained" Jan. 17, 2023. [Online]. Available: https://www.v7labs.com/blog/yolo-object-detection.

[3]. Redmon J, Divvala S, Girshick R, Farhadi A (2016) “You only look once: unified, real-time object detection”. In proceedings of the IEEE conference on computer vision and pattern recognition, pp 779-788.

[4]. Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P. “Image Quality Assessment: From Error Visibility to Structural Similarity”. IEEE Transactions on Image Processing, Vol. 13, No. 4, April 2004.

[5]. Lohith P.S., Shivasai G., Goud L.V., Lowkya C., Dr. Khan S. “Smart Surveillance Using Computer Vision”. International Journal of Innovative Science and Research Technology ISSN No:-2456-2165, Vol. 6, No. 6, June 2021.

[6]. Voila P., Jones M. "Rapid Object Detection using a Boosted Cascade of Simple Features". Accepted Conference on Computer Vision and Pattern Recognition 2001.
[7]. Behera G.S., "Face Detection with Haar Cascade" Dec 24, 2020 [Online]. Available: https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08.
[8]. Pannui N.S., Muradkar S., Chaurasia A. "IMAGE PROCESSING BASED FIRE DETECTION SYSTEM" Department of Electronics and Telecommunication Engineering, K.C. College of Engineering & Management studies & Research, Kopri, Thane(E), 400603.
[9]. Cao Z, Liao T, Song W, Chen Z, Li C (2021) Detecting the shuttlecock for a badminton robot: a YOLO based approach. Expert Syst Appl 164:113833. https://doi.org/10.1016/j.eswa.2020.113833.
[10]. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587.
[11]. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In proceedings of the IEEE conference on computer vision and pattern recognition, pp 770-778.
[12]. Jiao L, Zhang F, Liu F, Yang S, Li L, Feng Z, Qu R (2019) A survey of deep learning-based object detection. IEEE Access 7:128837–128868. https://doi.org/10.1109/ACCESS.2019.2939201.