Journal of Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.755215

OBJECT DETECTION BASED SECURITY SURVEILLANCE SYSTEM

Shraddha S. More[1], Akash Patel[2], Heenal Patel[3], Sushma Kumbhar[4]

MORESHRADDHA30@GMAIL.COM[1] , AKASHPATEL1098@GMAIL.COM[2], HEENALPATEL1905@GMAIL.COM[3], SUSHMAKUMBHAR2002@GMAIL.COM[4]

[1,2,3,4,]DEPARTMENT OF INFORMATION TECHNOLOGY
[1,2,3,4]St. John College of Engineering and Management, Palghar, India

Abstract

OBJECTIVE: This project aims to enhance security and surveillance systems by integrating multiple detection models, including object detection, face detection, face recognition, and motion detection, into a unified framework. The primary objectives of the project are to identify potential threats, detect anomalous behavior, and provide access control through automated processes. **METHODS:** To achieve these objectives, they employ state-of-the-art models such as YOLO- NAS for object detection and face detection. Additionally, motion detection is implemented using background subtraction methods. These models run simultaneously on edge devices, reducing latency and enabling real-time processing. The metadata generated by each model is sent to the cloud for further analysis and automated tasks, such as notifications and identity verification. **FINDINGS:** After training the model, it achieves the result of an mAP of 50.75%, f1 score of 5.51%, precision of 2.88% and recall of 82.7% in 100 epochs and inference on a video is of around 10-12 fps. **NOVELTY:** The novelty of this project lies in its comprehensive approach to security enhancement, leveraging cutting-edge detection technologies and edge computing to create a smarter and more efficient surveillance system.

Keywords: Convolutional Neural Network (CNN), Object Detection, Security Surveillance, YOLO-NAS.

1. Introduction

The increasing demand in security and ease of use, there is a need for intelligent security system which must act upon itself when any unauthorized activity happen. Traditional camera fails in doing so, as it need human intervention to act on the unusual activity after doing manual reviewing of footage stored by camera which is susceptible to human error and cannot adequately address the scale and complexity of modern security challenges. But with commence of Intelligent security system, there is no need of doing that and it increases the effectiveness of security and is better in proactive security measures[1]. Our research paper addresses this critical need by proposing an intelligent security and surveillance system designed to revolutionize traditional camera setups. By integrating cutting-edge technologies such as object detection, face detection, face recognition, and motion detection, our system aims to provide comprehensive security coverage in real-time. Our system uses state of the art model called YOLO-NAS which is capable of task like object detection that enables our system to accurately identify and recognize objects within the camera's field of view[2]. Our system incorporates face detection and recognition capabilities using YOLO NAS, allowing for the identification of individuals entering the monitored area. This not only enhances security by

enabling the system to detect unauthorized personnel but also facilitates access control and identity verification processes. Our system also has features motion detection functionality, which enables the identification of unusual movements or activities within the surveillance area. By utilizing background subtraction methods, our system can detect motion in real-time, thereby alerting security personnel to potential threats or security breaches. A main aspect of our system is its deployment on edge devices, which significantly reduces latency and enhances real-time processing capabilities. By processing data locally on edge devices, our system minimizes reliance on cloud infrastructure and ensures rapid response times to security incidents.

YOLO NAS serves as a cornerstone for enhancing object and face detection capabilities. YOLO NAS offers unparalleled efficiency and speed, processing images in real-time with a single pass through a deep neural network architecture. Its exceptional accuracy and precision minimize false positives and negatives, ensuring reliable detection of objects and faces within surveillance footage. Furthermore, YOLO NAS's scalability, flexibility, and adaptability make it suitable for deployment in diverse environments, from small office spaces to large campuses[3]. Crucially, its compatibility with edge computing aligns perfectly with our project's objectives, enabling local processing on edge devices to minimize latency and enhance real-time processing capabilities. By leveraging the strengths of YOLO NAS, our system can effectively identify and recognize objects and faces, significantly enhancing security measures and mitigating potential threats. Our research paper presents a novel approach to intelligent security and surveillance, addressing the shortcomings of traditional camera systems through the integration of advanced technologies and edge computing. Through the adoption of YOLO NAS for object and face detection tasks, our system offers unparalleled accuracy and efficiency, thereby enhancing overall security measures in diverse environments[2][3].

2. Literature Survey

Dang-Khoa, *et al.* [4] proposed a study on border surveillance systems using edge computing, improving upon cloud-based approaches. It introduced Border Edge, a lightweight human detection model for Raspberry Pi 4, offered high accuracy and low memory usage. However, the study lacked real-world testing, analysis of different devices, scalability discussion, model comparison, and ethical considerations.

Narina Thakur *et al.* [5] evaluated object detection algorithms on the MOT20 dataset and a custom UAV dataset, highlighting the Yolov5 model's superiority with 61% precision and 44% F-measure. Emphasis was on pedestrian detection for surveillance. Practical implications included improving security and disaster management.

Praahas Amin *et al*. [6] introduced an object detection system employing the YOLO algorithm, Python programming, and a COCO dataset. It successfully detected and classified objects but lacked specific performance metrics, comparative analysis with other algorithms, and discussions on computational requirements and dataset biases, warranting further investigation.

Gautam S *et al.* [7] offered a practical method that uses face images and the YOLOv8 deep learning model to assess children for autism spectrum disorder (ASD). Clinical findings of the

variations in facial characteristics between photos with and without ASD are supported by this. By gathering sensitive spatial and contextual information with fewer parameters, the work demonstrated the promise of deep learning in ASD screening, and the versatility of YOLOv8 for classification is demonstrated.

The study conducted by Deepak T. Mane *et al.* [8] using the YOLOv8 approach, proposed an ensemble model for precise and effective event recognition in traffic video surveillance. The model had been trained on the publicly accessible Car crash dataset, and its evaluation parameters numbers were improved over previous approaches. The model was used in real-time traffic surveillance system applications were demonstrated through experiments conducted with actual traffic video data.

Geethapriya. *et al*. [9] introduced YOLO for real-time object detection, emphasizing its speed and accuracy advantages over other algorithms. It discussed anchor boxes, IoU, and Non-Max Suppression techniques. Practical implications include applications in surveillance and robotics. However, detailed performance analysis, limitations, and real-world challenges were not extensively addressed.

The study introduced by Tanvir Ahmad *et al.* [10] a modified YOLOv1 neural network, enhancing object detection accuracy and speed. It adjusted the loss function, added a pooling layer, and incorporated. Experimental results on Pascal VOC datasets confirmed performance improvements. However, the study did not explicitly discuss limitations.

Saluky *et al.* [11] combined a dual background model with YOLO-NAS. It outperformed existing methods in accuracy and speed, promising enhanced security in public spaces. Despite its advancements, limitations such as computational requirements and ethical considerations were not addressed.

3. Proposed Methodology

3.1 YOLO-NAS

DeciAi developed the YOLO-NAS model. It is the successor of the previous 8 versions of Yolo. It is successful due its balance between speed and accuracy. It comes with additional features of attention mechanisms, quantization aware blocks, and inference time reparameterization. It worked by using neural architecture search techniques which automatically design the network architecture of model. The technique which is used in this version is called AutoNac[12]. It starts with a numerous variety of candidate architecture and evaluate each on validation dataset to calculate the loss function based on that it selects the best candidate architecture and uses it to generate a new population of candidate architectures and it is repeated until the best network architecture is found. YOLO-NAS is a Convolutional Neural Network (CNN) type of network which has same type of stack of layers of convolutional and pooling layer and fully connected layer[13]. It is also called as backbone , neck and head layer. The convolution or backbone layer extract feature from the input layer based on the filter weight and produce the feature map[14].

output = convolution(input, filter_weights)

The feature map is then passed to a pooling or neck which reduces the dimensions of feature map.

output = pooling(input)

The output of pooling layer is passed to full connected or head layer which takes the reduced feature map and convert it into vector of probabilities that represent the probability of image belonging to each class.

output = fully_connected(input)

The YOLO-NAS network architecture uses a hybrid quantization method to reduce the size of the network by converting the weights of a neural network from floating-point numbers to integer numbers. This can reduce the size of the network and make it faster to run on edge device. Yolo Nas also contains the QSP and QCI blocks which helps in post-training quantization [15]. It consists of NAS technology which helps in determining size and structure of stages in the architecture.

3.2 Block diagram of the proposed system

Following fig.1 illustrate the block diagram of the proposed system which showing various steps involved in performing experiment.

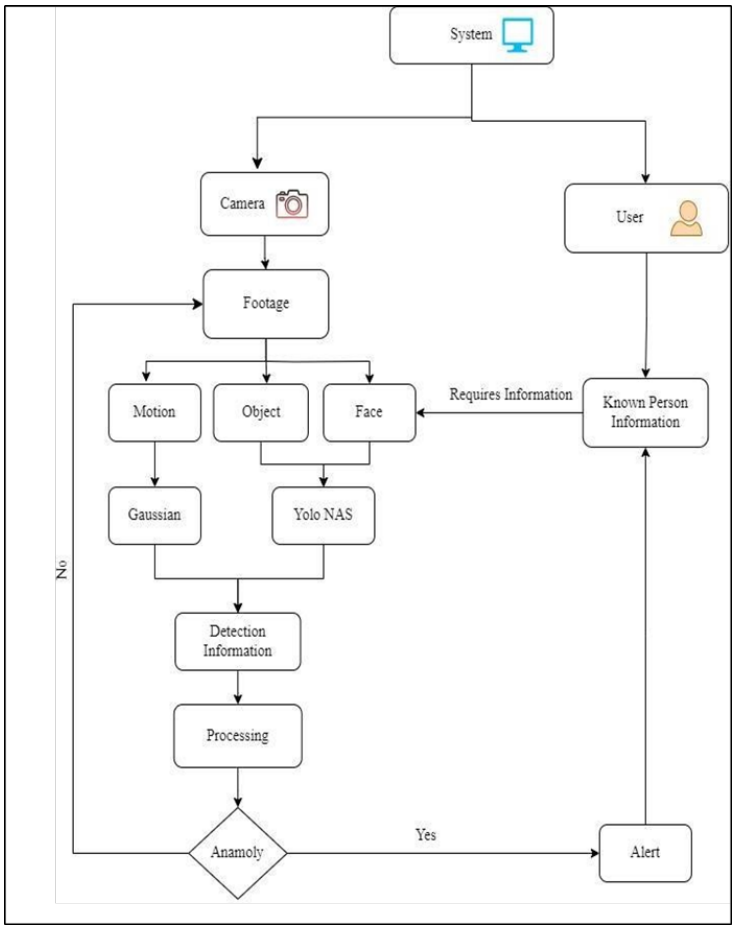


Fig1. Block diagram of the proposed system

The Motion detection model uses the Gaussian method to identify any unusual or suspicious movements within a user- defined region of interest in the video feed. This helps us detect potential threats and any unusual activity based on motion patterns. Simultaneously, the Object detection model, which is trained using the YOLO NAS algorithm, analyzes the footage to

recognize and classify various objects present in the frame. This allows us to identify any anomalous or unauthorized objects in the surveillance area.

The Face detection model, also trained using YOLO NAS, focuses specifically on detecting and recognizing human faces in the video feed. This information is then cross-referenced with a database of known individuals to determine if an unknown or unauthorized person is present in the area under surveillance.

The detection information from all three models is then processed, and if any anomaly is identified, such as an unknown object, suspicious motion, or an unrecognized individual, an alert is triggered. This alert can be further processed and communicated to the relevant authorities or personnel for appropriate action. Additionally, the system can access information about known individuals, which can be used for purposes like access control or identity verification within the surveilled premises.

The user also has the flexibility to enable or disable any of the detection models based on their specific requirements or the nature of the surveillance operation. It's important to note that we are implementing these models on edge devices to minimize latency and ensure real-time processing of the video feeds, while leveraging the cloud for further processing and analysis of the detection information when required.

4. Implementation

4.1 Dataset

The experimental dataset includes 9868 images, has been gathered for a number of uses, involving research, computer vision techniques, and deep learning model training. The images here offer a wide variety of topics and scenarios for in-depth investigation and testing. Three subsets of the dataset are used for testing, validation, and training. To train deep learning models, 6907 photos (or 70% of the dataset) make up the training set. 1982 photos (20%) from the validation set are used to improve and fine-tune the model's performance. 979 images (10%) make up the testing set, which is used to determine how well the trained models perform on untested data and how well they generalize. This division guarantees robust model construction and evaluation.

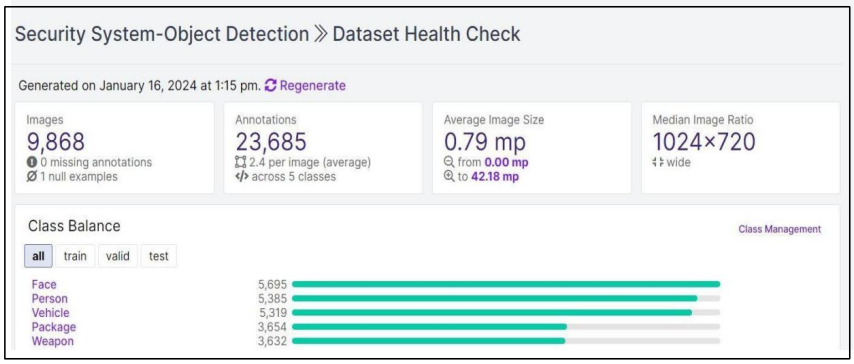


Fig 2. Collected Dataset

4.2 Classes

Classes in the dataset include 5 objects: 'Person', 'Face', 'Vehicle', 'Package', and 'Weapon', which are used to identify any activity happening in the footage. The 'Person' object is used to identify any humans passing by in the video, aiding in knowing the presence of humans around

the premises. The 'Face' object is used to identify the identity of a person, which is then matched by known faces stored in the database. The 'Vehicle' object is used to identify the presence of any vehicle on the premises, enhancing the security of the owner's vehicle against theft. The 'Package' object is used to identify the presence of any delivery boxes lying around or being carried on the premises. The 'Weapon' object helps in identifying any presence of ammunition around the premises.

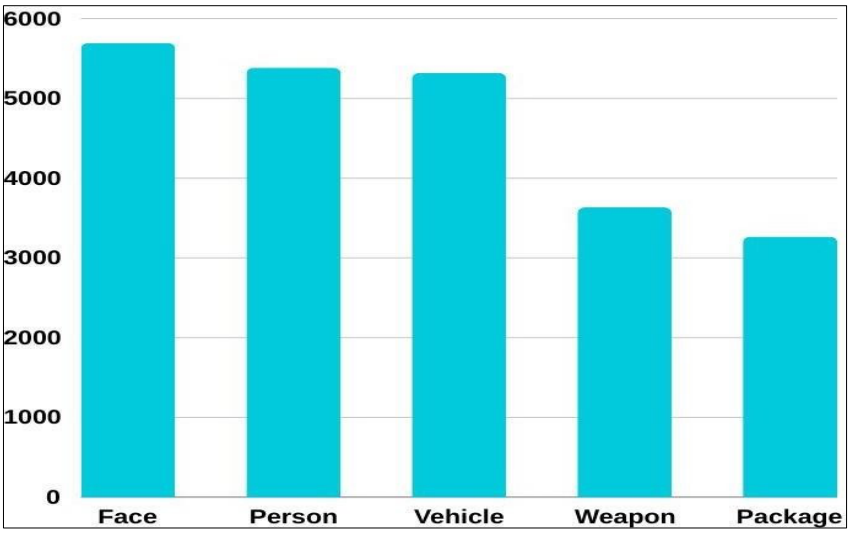


Fig 3.Classes of the dataset

4.3 Preprocessing

In the preprocessing pipeline, the auto-oriented step involves automatically detecting the orientation of each image and correcting it as necessary. This ensures that all images are consistently oriented, regardless of how they were originally captured. By standardizing the orientation, subsequent analysis tasks can be performed more accurately, as orientation discrepancies can introduce errors in feature extraction or object detection algorithms. Auto-orientation algorithms typically leverage image features or metadata to determine the correct orientation, allowing images to be uniformly aligned before further processing.

Following auto-orientation, the next step in the preprocessing pipeline is resizing the images to fit within a predefined dimension of 640 x 640 pixels. Resizing ensures uniformity in image sizes, which is crucial for model compatibility and computational efficiency in downstream tasks such as object recognition or classification. By resizing images to a consistent size, computational resources can be utilized more effectively, and models can be trained or deployed more efficiently. Additionally, fitting images within the specified dimension ensures that valuable information is retained while optimizing computational performance, striking a balance between accuracy and efficiency in image processing workflows.

4.4 Augmentation

In augmentation of image data, several transformations are applied to enhance the diversity and robustness of the dataset. Firstly, horizontal flipping is employed, which mirrors images along the vertical axis. This augmentation technique helps mitigate biases related to orientation and increases the variability of the dataset. Additionally, cropping is implemented to extract specific regions of interest from the images, enabling focus on relevant features while discarding irrelevant background information. Brightness adjustment is then utilized to modify

the overall luminance of the images, enhancing their visual appearance and adaptability to different lighting conditions. Furthermore, a blur effect with a maximum radius of 2.6 pixels is applied to introduce smoothness and reduce high-frequency noise, which can improve the generalization capability of models trained on the augmented data. Finally, noise is introduced into the images, adding random variations to pixel values, thereby simulating real-world imperfections and enhancing the dataset's resilience to noise during inference. By incorporating these augmentation techniques, the dataset becomes more diverse, robust, and representative, ultimately enhancing the performance and generalization of machine learning models trained on it.

4.5 Training Process

The training process for the Intelligent Security System began with the selection of the model architecture. The team opted for YOLO NAS due to its capability for faster latency, essential for the project's requirements. Following this, they embarked on fine-tuning the YOLO NAS model for their specific task, leveraging transfer learning with a pretrained model and training it on their dataset.

For dataset collection, images of persons, vehicles, weapons, faces, and packages were gathered to enable prediction on video inputs from cameras. Roboflow facilitated effective dataset collection and management, resulting in 10,000 images initially. Subsequently, preprocessing and augmentation techniques were applied, including auto orientation, resizing, horizontal flipping, and various enhancements like cropping, brightness adjustments, blur, and noise addition, resulting in a dataset augmentation to 20,000 images.

The training phase involved setting hyperparameters for efficient model training, including a batch size of 8, maximum epochs of 100, input image size of 640 x 640, optimizer as Adam, loss function as PPYoloELoss, and weight decay of 0.0001. the model was trained until 100 epochs achieving mAP of 50.75% for the dataset. Following training, the model underwent evaluation on a test dataset, with various evaluation metrics such as precision, F1 score, recall, and mAP being computed. The testing phase was successful, with the model detecting most classes in the images.

After successful evaluation, the trained YOLO NAS model was saved in a .pth format for inferencing on video inputs. Subsequently, a system pipeline was created to integrate the model into the project. Functionalities such as sending text messages and generating detection logs were added to enhance system efficiency and functionality. The system aimed to utilize the model's predictions to address security concerns, enabling traditional cameras to autonomously detect and alert people of suspicious activities.

4.6 Detection Outputs

Fig. 4 displays the output of object detection, specifically detecting the face class among the five classes. The system continuously scans for the presence of relevant objects and, upon detection, highlights them by drawing bounding boxes. In this case, the face class is identified and highlighted with a red bounding box. The five detectable classes include Face, Vehicle, Weapon, Package, and Human, all utilized to detect anomalies within the premises through automated pipeline logic. When a known face is detected, the system automatically identifies the person and displays their name, while also capturing a screenshot of the moment. In this instance, 'Akash' is recognized and identified from the known person database, as depicted in

the above screenshot.

Fig 4. Face detection output

The following fig. 5, shows the output of motion detection, where it detects motion in the region of interest(ROI) inputted by the user. If an object exhibits motion beyond the set threshold, it is considered in motion, and a red bounding box is drawn around it. The room status label above will update to 'Room Occupied' whenever motion is detected in the ROI. A screenshot is taken at set intervals to identify the moving object through object detection. In the image, you can observe a hand in motion, indicated by the drawn bounding box. Motion detection occurs exclusively within the user-defined ROI.

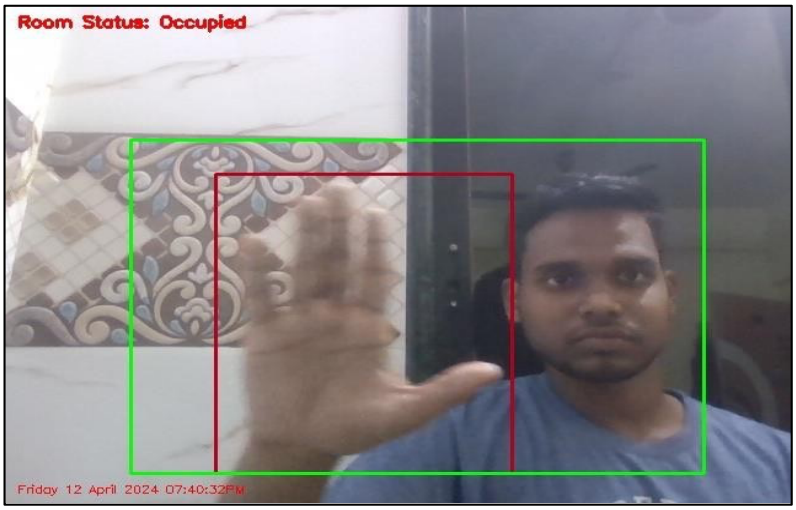


Fig.5. Motion detection output

5. Result and Discussion

To measure the performance of proposed system following performance parameters are considered.

1. mAP (Mean Average Precision)

It is a widely used performance metric in object detection tasks. It combines precision and recall to evaluate how well an object detection model performs across different confidence thresholds.It considers both the accuracy of detected objects (precision) and the coverage of actual objects (recall).

mAP achieved is 50.75% through various fine-tuning techniques of hyperparameters. It was gradually increasing with every increase in epochs and finally we were able to get the better mAP of 50.75%.

$$mAP = \frac{1}{k}\sum_{i}^{k} AP_i \quad \ldots\ldots\ldots\ldots\ldots (1)$$

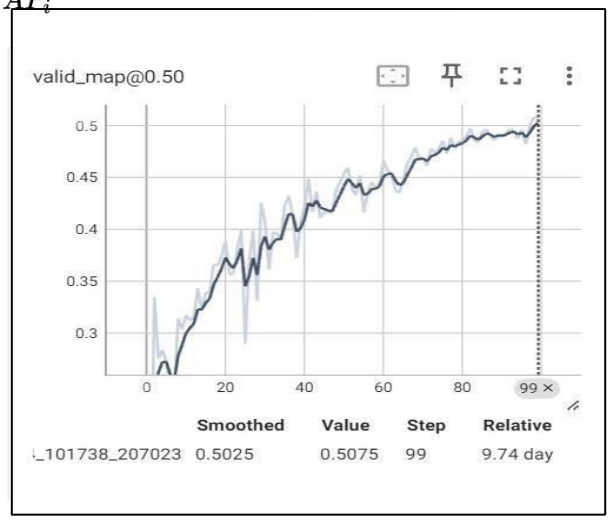


Fig 6. Graph of mAP

2. Precision

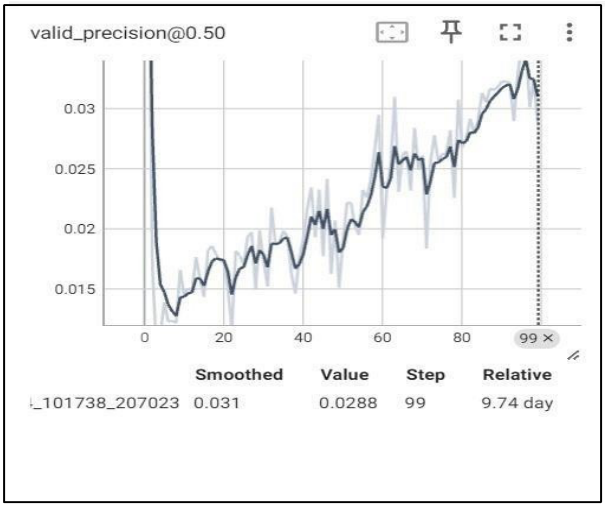
It is a measure of how accurately the model predicts favorable outcomes. It determines the frequency with which the model accurately predicts good occurrences. Precision, in mathematical terms, is quantified as the ratio of true positive predictions to the overall number of positive predictions, which includes both true positives and false positives.

Here, Precision achieves the score of 2.88% at 99th epoch.

$$\text{Precision} = \frac{TP}{TP + FP}$$

…… (2)

Fig 7. Graph of Precision

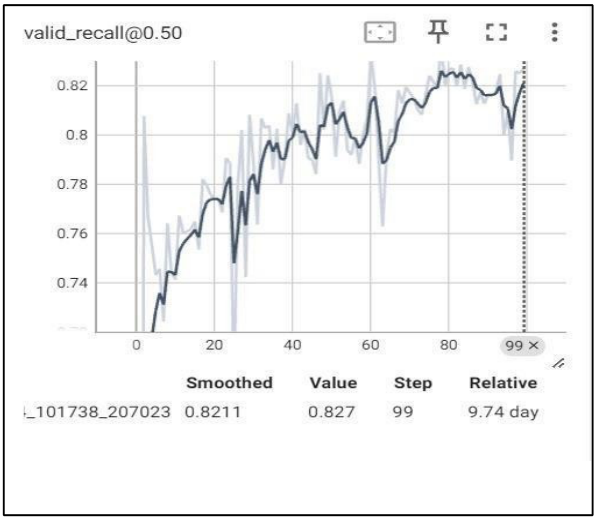


3. Recall

It measures the ability of a model to accurately detect true positive cases. The true positive rate, also known as sensitivity or recall, is calculated by dividing the number of correctly predicted positive instances by the total number of actual positive instances(true positives + falsenegatives).

Recall measures the model's capacity to correctly identify all instances that are positive. Recall was having a great ups and downs but then achieved the great score of 82.7%. It was gradually increasing with each increasing epoch.

$$\text{Recall}=\frac{TP}{TP+FN} \qquad \ldots\ldots\ldots\ldots\ldots (3)$$



4. F1 Score

It is computed by taking a harmonic average of precision and recall. It connects these two measurements into one single value, giving a reasonable evaluation of a model's effectiveness.

The achieved F1 score is 5.51% through various *fine-tuning* techniques of hyperparameters. It first gradually decreased during the warmup period of training then it increased gradually and attained this score.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad \ldots\ldots\ldots\ldots\ldots (4)$$

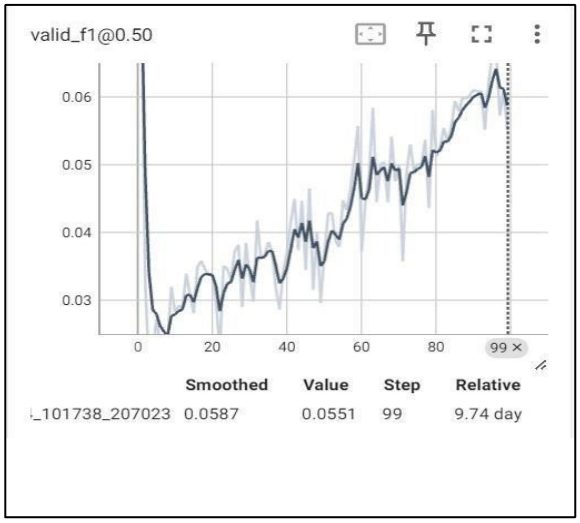


Fig 9. Graph of F1 Score

6. CONCLUSION AND FUTURE SCOPE

In conclusion, incorporating cutting-edge technology into traditional surveillance systems signifies a significant advancement in improving security protocols. By implementing intelligent modules like Face Detection, Motion Detection, Object Detection, and Face Recognition, our system not only improves the effectiveness of surveillance cameras but also provides them proactive ability to react immediately to possible threats. Our approach converts regular security cameras into intelligent devices that can extract valuable data from video streams instantly by utilizing cutting-edge hardware and sophisticated algorithms. Security system's modular design guarantees flexibility and modification, enabling customers to customize information retrieval and alarm making to meet their needs and preferences. This flexibility enables seamless integration into diverse environments, offering comprehensive security solutions tailored to the unique needs of each setting. In a rapidly evolving security landscape, where threats are dynamic and multifaceted, the need for intelligent and efficient surveillance systems is paramount. This project bridges the gap by introducing a holistic approach to security, combining advanced technologies with customizable features to provide a robust defense mechanism. Ultimately, the intelligent security and surveillance system promises to usher in a new era of safety and vigilance, ensuring the protection of individuals, assets, and communities in an ever-changing world

In the future, advanced algorithms will boost accuracy in object detection and facial recognition for enhanced security. Automation will streamline threat detection and response, with a focus on improving identity verification and detecting abnormal activities. Optimization for edge devices will ensure speed, efficiency, scalability, and adaptability.

7. References

1. Saluky, Saluky, Gusti Baskara Nugraha, and Suhono Harso Supangkat. "Enhancing Abandoned Object Detection with Dual Background Models and Yolo-NAS." *International Journal of Intelligent Systems and Applications in Engineering* 12, no. 2 (2024): 547-554.
2. Wang, Xiangheng, Hengyi Li, Xuebin Yue, and Lin Meng. "A comprehensive survey on object detection YOLO." *Proceedings http://ceur-ws. org ISSN* 1613 (2023): 0073.
3. Zou, Xinrui. "A review of object detection techniques." In *2019 International conference on smart grid and electrical automation (ICSGEA)*, pp. 251-254. IEEE, 2019.
4. Charoenjai, Kittikan, Worapan Kusakunniran, Tipajin Thaipisutikul, Nutcha Yodrabum, and Irin Chaikangwan. "Automatic detection of nostril and key markers in images." *Intelligent Systems with Applications* 21 (2024): 200327.
5. Luong-Huu, Dang-Khoa, Tan-An Ngo, Huy-Tan Thai, and Kim-Hung Le. "A Real-time Border Surveillance System using Deep Learning and Edge Computing." In *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pp. 1-6. IEEE, 2022.
6. Thakur, Narina, Preeti Nagrath, Rachna Jain, Dharmender Saini, Nitika Sharma, and Jude Hemanth. "Object detection in deep surveillance." (2021).
7. Amin, Praahas, B. S. Anushree, Bhavana B. Shetty, K. Kavya, and Likitha Shetty. "Object detection using machine learning technique." *International Research Journal of Engineering and Technology (IRJET)* 6, no. 05 (2019): 2395-0056.
8. Gautam, Subash, Prabin Sharma, Kisan Thapa, Mala Deep Upadhaya, Dikshya Thapa, Salik Ram Khanal, and Vítor Manuel de Jesus Filipe. "Screening Autism Spectrum Disorder in childrens using Deep Learning Approach: Evaluating the classification model of YOLOv8 by comparing with other models." *arXiv preprint arXiv:2306.14300* (2023).
9. Mane, D. T. ., Sangve, S. ., Kandhare, S. ., Mohole, S. ., Sonar, S. ., & Tupare, S. . (2023). Real-Time Vehicle Accident Recognition from Traffic Video Surveillance using YOLOV8 and OpenCV. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(5s), 250–258.
10. Geethapriya. S, N., et al. *Real-Time Object Detection with Yolo*. International Journal of Engineering and Advanced Technology (IJEAT), Feb. 2019.
11. Ahmad, Tanvir,Muhammad Yahya, Belal Ahmad, Shah Nazir ,and Amin ul Haq "Object Detection through Modified YOLO Neural Network." *Scientific Programming*, vol. 2020, 6 June 2020, pp. 1–10.
12. Sharma, M. N. (2023). Image and video segmentation using YOLO-NAS and Segment Anything Model (SAM): Machine learning. International Research Journal of Modernization in Engineering Technology and Science, 05(10), 1915.
13. Zhao, X., Wang, L., Zhang, Y. *et al.* A review of convolutional neural networks in computer vision. *Artif Intell Rev* **57**, 99 (2024).
14. Saluky, I., Gusti Baskara Nugraha, I., & Harso Supangkat, S. (2024). Enhancing abandoned object detection with dual background models and Yolo-NAS. International Journal of Intelligent Systems and Applications in Engineering, 12(2), 547–554.
15. Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* 2023, *5*, 1680-1716. https://doi.org/10.3390/make5040083