Design and Implementation of a Surveillance and Security System Using OpenCV and YOLO8

Akshitha Reddy Thokala, Ali Abbasi Maleki, Arun Sai Teja Medisetty, Girish Kumar Reddy Boreddy, Nupoor Vijay Kumbhar, Rachana, Sree Sai Preetham Nandamuri, Venkata Achyuth Kumar Sanagapalli, Venkata Kaushik Belusonti, Vishnu Inkollu

***Abstract*—The importance of having effective security measures, whether it's on the streets, at home, or in workplaces, cannot be overstated. Given the paramount importance of security, security experts continuously refine both the mechanisms and associated tools to uphold maximum security measures. Many offices and homes have installed CCTV cameras to identify the intruders and ensure the safety of the employees and residents. CCTV cameras are increasingly being installed in many households, especially those occupied by single individuals and the elderly. Human observation of these security cameras 24x7 is close to impossible and there is a strong necessity to monitor the people captured by CCTV cameras. Creating a proficient surveillance system poses a notable challenge, yet leveraging recent advancements in computer vision, digital image processing, and decision-making tools can help surmount this obstacle. This paper demonstrates the effective security surveillance system developed utilizing digital image processing, computer vision, and deep learning techniques to identify intruders from CCTV footage. A user friendly GUI interface is created that captures the live video and displays the intruder snapshot in the color image. The GUI also displays the filtered images of the latest intruder. To provide a better interaction with the user, a GUI is developed using Flask application. The GUI offers the live streaming, displays the original colored and filtered images of the recent intruder.**

I. Introduction

According to FBI records, residential burglary has shown a notable increase, with an observed annual rise of 5.48% throughout the first decade of the 21st century [1]. With the increase in frequency and unpredictability of the criminal activities, causing significant harm, law enforcement agencies are encouraging the households to employ surveillance systems to detect intruders and avoid hazardous situations effectively. Implementation of a smart security surveillance system at office and home is a vital preventive measure against crimes. A smart surveillance system aids in detecting intruders and promptly alerts of unauthorized entry, allowing for precautionary measures to be taken, thereby mitigating potential losses.

Face recognition, a technique where detected images are matched against trained images is employed. A model is trained using the YOLO8 (You Only Look Once) algorithm, utilizing a dataset comprising images of authorized individuals and random images sourced from the internet for intruders. The graphical user interface (GUI) provides live streaming and displays both the original colored and filtered images of recent intruders. The utilization of the security surveillance system is elaborately explained in the sections named, literature review, data preprocessing, model training, results, and analysis. Additionally, the challenges encountered during the development of the surveillance system and future avenues for improvement are thoroughly discussed.

Github link will be added here

II. Literature Review

Generally, a set of reliable features is extracted from input photos at the outset of object detection processes. Object recognition encompasses various computer vision tasks, such as identifying objects. Multi-object tracking has been the focus of numerous recent studies and has become a popular topic in computer vision. In conventional approaches to multi-object tracking, tracking is often limited to the pixels necessary to generate the motion trajectory coordinates. Frequent deviations in tracking results can significantly hinder the identification of anomalous behaviors, such as wandering and tailing. Deep learning exhibits robust representational capabilities due to its ability to extract multi-layered features latent within data. Wang, Ouyang W, Wang X propose a sequential training approach based on CNN to effectively transfer the deep features of online pre-training applications, leveraging the advantages of deep learning [2]. This approach views the process of training CNNs online as a progressive learning of an ideal ensemble of base learners, ensuring that learned features are not highly correlated with each other. Additionally, a convolution with a mask layer is proposed to further reduce overfitting. Ju, J, Kim D, and Ku B proposed an online multi-object tracking method applicable to real-time applications [3]. Firstly, to swiftly and effectively allocate detections to tracks, an appearance update and a new affinity model are integrated with a frame-by-frame association object tracking technique that does not rely on online learning. Secondly, to address drifting tracks resulting from prolonged incorrect detections and rapid motion changes of objects under occlusion, a two-stage drift management solution with new track confidence is provided. Additionally, another introduced method is a fragmentation handling method based on track-to-track association, aimed at resolving issues where the object trajectory is broken into several tracks.

Chen, Ai H, Shang C, et al. [4] present an innovative online framework for tracking multiple objects. This framework utilizes features from various CNN layers, with the higher-level features trained as a target class classifier and the lower-level features utilized for target matching and associations with higher-level layers containing more detailed information. The depth model is trained offline, and the frames retain historical appearance characteristics for each target to minimize computational costs associated with online fine-tuning. In [5] [6],region-based convolutional neural networks such as R-CNN and Faster R-CNN are highlighted for their improved efficiency and precision. These models excel at accurately identifying and segmenting objects, with their method combining bottom-up region recommendations with high-capacity CNNs, demonstrating enhanced performance when trained on small, labeled datasets. These advancements illustrate the potential for further breakthroughs in object detection and represent a step forward for the field.

In a comparative study conducted by [7],on R-CNN, Fast R-CNN, Faster R-CNN, ResNet, and YOLO, insightful findings were revealed. Faster R-CNN exhibited improved speed in training, detection, and selective search processes while maintaining ease of understanding. However, it is computationally expensive, requires large labeled datasets, and is inefficient. Conversely, YOLO demonstrated accurate results by focusing on the highest probability of object presence. Despite similarities in using probable boundary boxes, YOLO and its variant YOLOv2 differ from Faster R-CNN systems.

It became apparent that object detection and tracking in image sequence is quiet a challenging task and YOLO stands out for it's ability to quickly identify objects and provide immediate results which make it more suitable for real time processing compared to other models. Over multiple iterations YOLO family has evolved and each building upon the previous model to address the limitations and enhance performance as shown in Fig. 1.

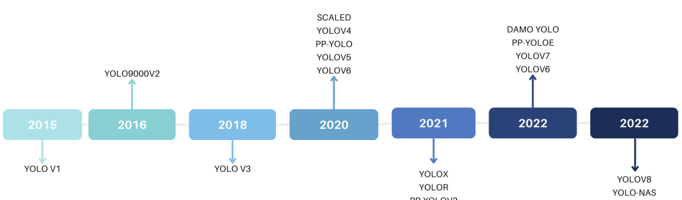


Fig. 1: YOLO Timeline.

YOLO which is also know as You Only Look Once is a real time object detection algorithm which uses Convolution Neural Network to predict the boundary boxes and to calculate the class probabilities of the object in the input image. YOLO was implemented on Darknet frame work at the first.

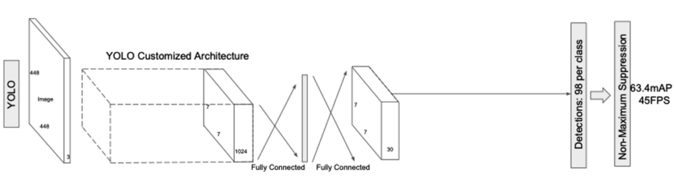


Fig. 2: Architecture of YOLO V1 [8].

The main disadvantages of YOLO V1 model was it suffered from localization errors and the used to suffer more with small objects and struggled with handling occluded objects due to its single-grid-cell approach it can detect maximum of two objects in the same grid. YOLO v2 addressed these limitations by introducing anchor boxes, predefined boxes of different aspect rations and scales for better handling of objects with different sizes and aspect ratios and employing a multi-scale approach with feature pyramid networks, FPN to detect objects at various scales. Additionally, YOLO v2 utilized a sum-squared error, SSE loss function for faster convergence and improved performance. In YOLO V3 they improved object detection accuracy by incorporating a feature pyramid network and implementing a refined loss function that combined classification and localization loss. However, it faced challenges related to computational complexity and longer training times due to its deeper architecture and multi-scale approach. YOLO v4 implemented Spatial Pyramid Processing (SPP) and Cross-stage partial connection, CSP to boost both accuracy and efficiency, while also tackling challenges like effectively detecting partially or completely hidden objects. However, these enhancements resulted in a notable demand for computational resources during both the training and deployment phase. YOLO V5 marked a significant shift by adopting a new architecture which is EfficientDet architecture it, prioritized efficiency without compromising accuracy.

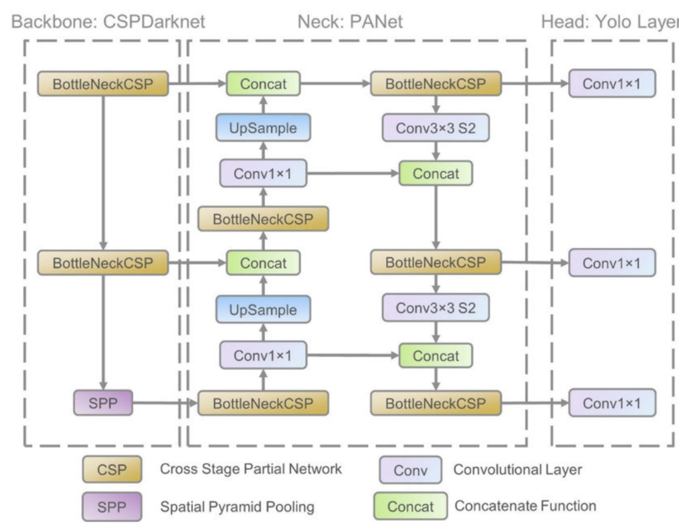


Fig. 3: Network architecture of YOLO V5 [9].

It also introduced anchor-free detection, eliminating the need for anchor boxes and offering more flexibility in object detection. Transfer learning was utilized to enhance generalization and adaptability to new datasets. Despite these advancements, YOLO v5 faced challenges such as potential overfitting and limited interpretability due to its reliance on transfer learning and EfficientDet architecture. In response to the need for lightweight models with faster inference times, YOLO v6 embraced the EfficientNet-Lite family of architectures, offering improved efficiency and reduced computational resources. It also incorporated data augmentation techniques to enhance model robustness and generalization. However, YOLO v6's simplified architecture may sacrifice some detection accuracy, and it may not perform as well on large-scale datasets or complex scenes.

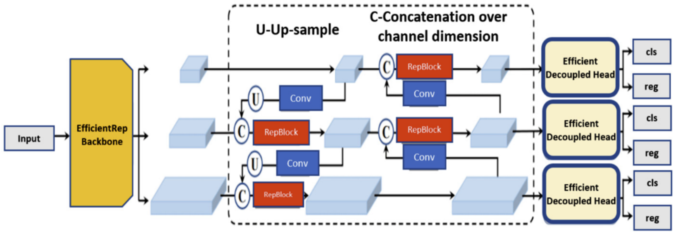


Fig. 4: Base architecture of YOLO V6 model [9].

YOLO v7 continued the trend of refining object detection models for greater efficiency and accuracy. It aimed to overcome the limitations of earlier versions while introducing new techniques to advance object detection capabilities. YOLOv7 undergoes end-to-end training on a sizable dataset of annotated images. Its arsenal includes sophisticated components such as an EfficientNet-based backbone network, an SPP (Spatial Pyramid Pooling) module for comprehensive multi-scale feature capture, and a PAN (Path Aggregation Network) module for seamless integration of features across scales. Empirical findings demonstrate YOLOv7's superiority over previous state-of-the-art object detection models across various benchmarks, including COCO and PASCAL VOC. This dynamic focus led to improved performance, particularly in scenarios with cluttered backgrounds or obscured objects.
Moreover, YOLO v7 further refined its architecture and training methods to achieve quicker inference times without compromising accuracy. Balancing model complexity and computational efficiency allowed YOLO v7 to be suitable for real-time applications and environments with limited resources. Additionally, advanced regularization techniques were employed to prevent overfitting and enhance the model's adaptability across diverse datasets and situations.
However, YOLO v7 encountered challenges in handling very large datasets and complex scenes with intricate object relationships. In response, YOLO v8 was developed with a renewed focus on scalability and scene comprehension. By integrating advanced graph neural network (GNN) architectures, YOLO v8 aimed to capture contextual connections between objects and their surroundings, resulting in more precise and context-aware object detection.
Furthermore, YOLO v8 utilized techniques from self-supervised learning to extract rich representations from unlabelled data, enhancing the model's generalization across various domains. By simultaneously optimizing for detection accuracy and scene comprehension, YOLO v8 delivered significant enhancements in performance and robustness, especially in challenging real-world scenarios.

For decades object detection and tracking have stood as major pillars in surveillance and they are continuously evolving to meet the demands of various applications these technologies have not only facilitated the identification and monitoring the objects within video sequences but also have played a vital role in enhancing security measures across diverse sectors. Originally considered active research areas, object detection and tracking have evolved into essential parts of surveillance systems across the world. Object tracking is the ongoing location of detected objects across time over many frames, whereas object detection is the process of finding and detecting objects within the frames of video sequences. The need for better surveillance capabilities led to the evolution of object detection and tracking, which has seen tremendous breakthroughs from early simple algorithms to complex deep learning systems. Nowadays, these systems offer features like behavior analysis, anomaly detection, and crowd monitoring in addition to real-time item identification and tracking. Nowadays, these systems offer features like behavior analysis, anomaly detection, and crowd monitoring in addition to real-time item identification and tracking. Furthermore, by incorporating state-of-the-art computer vision techniques like semantic segmentation and instance segmentation, contemporary surveillance systems are able to offer more comprehensive analysis and precise evaluations of the situations they view. With advancements in artificial intelligence and machine learning that drive surveillance technology, object detection and tracking are set to become even more crucial in protecting public safety, securing vital infrastructure, and facilitating the development of smarter cities in the future.

TABLE I: Comparison of Yolo models

Yolo Version	***Backbone***	***Description***
V1	DarkNet-24	Can detect only two objects in the same grid
V2	DarkNet-24	Added batch normalization, k-means clustering for anchor boxes, and expanded category detection to over 9000 categories.
V3	DarkNet-53	Utilized multi-scale predictions and spatial pyramid pooling leading to larger receptive field.
V4	CPSDarkNet-53	Spatial Pyramid Processing (SPP) and Cross-stage partial connection (CSP) to boost both accuracy and efficiency.
V5	Modified CSPv7	First variant based in PyTorch, making it available to a wider audience. Incorporated the anchor free detection processes into the YOLO-v5 pipeline.
V6	EfficientRep	Presented new loss determination mechanisms (VFL, DFL, and SIoU/GIoU)
V7	RepConvN	Architectural introductions included E-ELAN for faster convergence along with bag-of-freebies including RepConvN and reparameterization-planning.
V8	YOLO V8	Anchor-free reducing the number of prediction boxes whilst speeding up non-maximum suppression.

III. Approach

The approach for identifying the intruder from the live stream is as follows: The first and foremost is the selection of appropriate model and Yolo8, a mixture of OpenCV and CNN is selected for developing the system. A image dataset is created and all the images in the dataset are labeled. To do this, an open source version of annotation tool, Roboflow is selected. An user-friendly, visually appealing GUI is created to display the live stream and also show the intruder snapshot from the livestream in the separate snapshot. Created a python script to launch the camera and capture the live stream. The captured livestream is converted into frames at an interval of 45 frames per second and the same is fed to the YOLO8 model trained with images dataset. The model extracts the snapshots of the persons in the frame and classifies the snapshots into three authorized classes, named Ali, Arun and Kaushik and the fourth one, the intruder class. The model estimates the confidence of the snapshot into the four mentioned classes and if the confidence level is more than the pre-defined threshold value, the snapshot is assigned to the respective class. The snapshots assigned to the classes are saved to the respective folders. The latest intruder snapshot is displayed in the GUI every 5 seconds. The image processing techniques are applied onto the intruder snapshot and displayed on GUI to ensure the intruder snapshot can be better visualized.

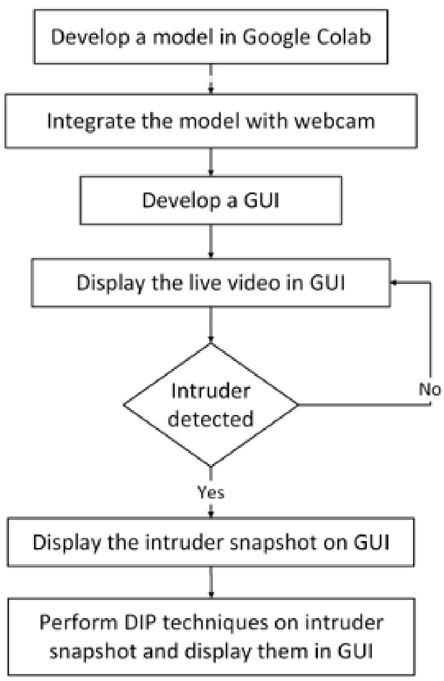


Fig. 5: Process Flow.

IV. Methodology

A. Dataset Preparation

Creating an image dataset for the YOLO v8 model requires multiple steps, which are outlined in detail and depicted in the figure below.

1) Collection: Data collection for the YOLO v8 model involves defining four classes: Ali, Arun, and Kaushik for authorized individuals, and Intruder for unauthorized persons. When the model receives an image of an individual, it categorizes the snapshot into the appropriate class: Ali, Arun,

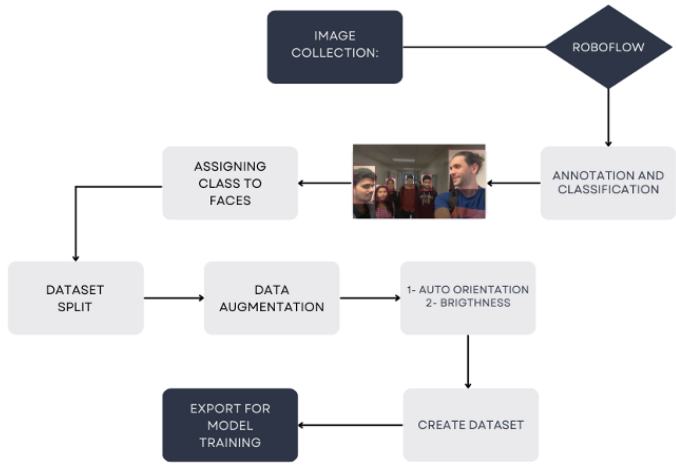


Fig. 6: Process flow for dataset preparation.

Kaushik, or Intruder. A total of 1000 images are collected for each authorized person, while 2000 images are collected for the intruder class. The higher number of images for intruders aims to capture a broader range of features. Conversely, the limited number of images for authorized individuals ensures comprehensive consideration of all relevant features, as multiple images of the same person are gathered. The collected images feature various backgrounds and angles to enhance the model's robustness. The model's effectiveness improves with a greater quantity of images featuring diverse backgrounds, lighting conditions, and angles, albeit at the cost of increased computational resources. However, due to constraints in computational power, the number of collected images is limited to 5000.

2) Annotation: Roboflow, an open-source tool, offers a user-friendly interface for annotating images, facilitating the labeling of collected images into four classes: Ali, Arun, Kaushik, and Intruder. This tool enables users to draw bounding boxes around individuals and assign corresponding class labels. Bounding boxes are drawn around the faces in the image and labeled with their respective class.

3) Split for Model Training: The image dataset is split into training, testing and validation for model development with split ratio of 7:2:1.

4) Augmentation: Seventy percent of the images allocated for training the model undergo additional augmentation in Roboflow, including 90-degree clockwise and anti-clockwise rotations, as well as adjustments in brightness levels. This augmentation process doubles the number of training images from 3500 to 7000, enhancing the diversity and robustness of the training dataset.

In total, 8050 images are utilized for training the model, with 7000 allocated for training, 700 for testing, and 350 for validation.

B. Model

1) Model Configuration: The YOLO v8 model is set up for person detection, achieved by removing the pre-trained model.

The suitable backbone network, Darknet, is chosen for this purpose. The model is configured to recognize four classes, namely Ali, Arun, Kaushik, and Intruder, each assigned a specific label.

2) Model Training: The YOLO v8 model undergoes training using the annotated dataset, with the number of epochs restricted to 50 due to computational constraints. Training the YOLOv8 model with 7000 training images for 50 epochs typically takes around 6 hours in Google Colab. Therefore, it's essential to plan the epochs effectively to maximize their usage. To achieve this, the model is initially trained for 20 epochs, and the best weight combination is selected from this initial phase. The weights obtained from the best epoch within the first 20 epochs serve as the initial weights for the subsequent 30 epochs. This approach ensures the efficient utilization of all 50 epochs, considering the computational limitations.

3) Model Evaluation: The trained model undergoes evaluation using the testing and validation datasets to assess its performance. The evaluation metrics employed for this assessment include box precision, box recall, mean average precision with a threshold of 0.5 (mAP50), and mean average precision with thresholds ranging from 0.5 to 0.9 (mAP50-95).

The performance evaluation metrics for the training dataset are presented in the Table. II.

TABLE II: Model evaluation on the training data

Epoch #	***mAP50***	***mAP50-95***
Epoch 1	0.982	0.673
Epoch 30	0.995	0.815
Epoch 35	0.995	0.831

The model training progress shows a consistent improvement in performance metrics, mAP50, and mAP50-95 as the training progresses from epoch 1 to epoch 35.

The predictions generated by the trained YOLO v8 model are illustrated in Fig. 7.

C. Digital Image Processing Techniques

Seven image processing techniques are employed on the intruder snapshot to ensure that the intruder remains recognizable under various operating conditions, including changes in lighting conditions such as sunlight or darkness, as well as different types of backgrounds.

1) Bicubic Interpolation: The space reserved for displaying the filtered image snapshots is limited, as there is a necessity to accommodate six filtered snapshots. Consequently, the size of the intruder snapshot is reduced to 50% of its original dimensions to fit within the constrained space designated for the filtered intruder snapshots. Bicubic interpolation is used to downsize the intruder snapshot image due to its superior accuracy, smoothness, and robustness compared to other interpolation techniques, such as nearest and linear interpolation methods.

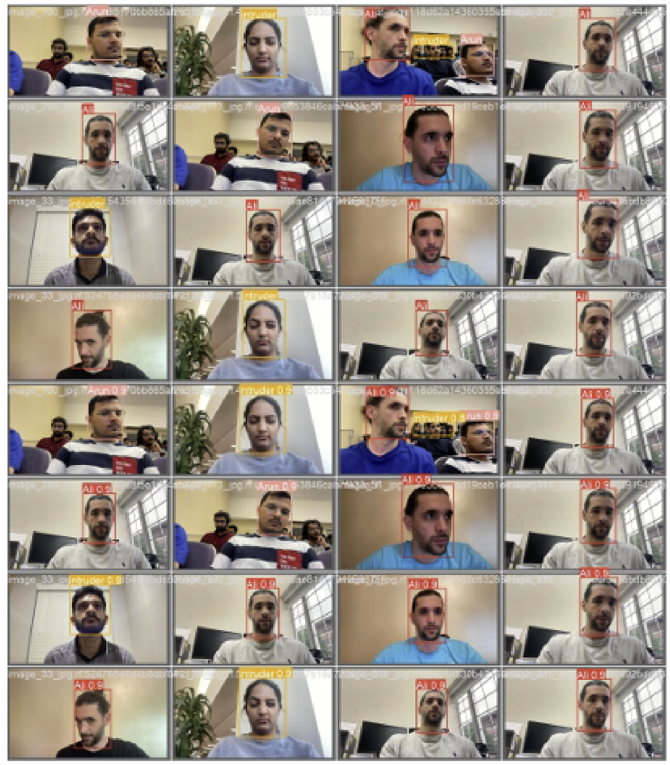


Fig. 7: Prediction of classes with YOLO v8 model.

2) Median Filter: The intruder snapshot undergoes median filter image processing to diminish noise, enhance image quality, and retain edges. This technique is renowned for its resilience to outliers. A 10x10 kernel is employed for implementing the median filter.

3) Ideal High Pass Filter: The ideal high-pass filter, also referred to as a sharpening filter, is implemented on the intruder snapshot to diminish low-frequency components while accentuating high-frequency components. This targeted enhancement of high-frequency elements serves to sharpen the image's edges and intricate details, thereby enhancing its overall clarity. While high-pass filters excel at accentuating image details, they can also amplify noise and irregularities. The cut-off frequency is hardcoded to 50 Hz.

4) Histogram Equalization: Histogram equalization is used to enhance the contrast and brightness of an image by spreading or redistributing the intensities to achieve more uniform pixel distribution. This helps to maximize the utilization of available pixel values and enhancing the details in the image.

5) Edge Detection: As its name implies, edge detection is employed to locate the edges within an image. This technique is designed to detect sudden alterations in pixel intensity, which signify edges. In images, edges typically delineate boundaries between distinct objects, a factor critical in tasks such as image segmentation, object detection, and feature extraction. While edge detection substantially enhances image quality and accelerates automated analysis tasks, it can also be sensitive to noise, potentially yielding false positives in regions with low contrast or intricate textures. The threshold value is automatically determined using the image histogram to ensure the effective detection of edges.

6) Otsu Thresholding: Otsu thresholding is known for separating foreground objects from background by determining the optimal threshold value. This method calculates the optimal threshold by maximizing the inter class variance between foreground and background pixel intensities. Otsu's approach finds the threshold that minimizes the intra-class variation while maximizing the inter-class variance, thereby distinguishing the two classes of pixel intensities (background and foreground), by iteratively evaluating the variance within the two classes. In applications where accurate object boundary delineation is essential, such picture segmentation, this thresholding technique is especially helpful. Otsu thresholding reduces the computational load and does away with the necessity for manual threshold setting by providing an easy-to-use yet efficient method of automatically binarizing images. Otsu thresholding, however, might not work as well in situations including uneven lighting, noisy backgrounds, or pictures with intricate textures; in these situations, more pre-processing stages or adaptive thresholding methods could be required to get reliable segmentation results. Despite these drawbacks, Otsu thresholding is still a useful technique in image processing since it provides a quick and dependable way to automatically threshold images for a range of applications.

7) Adaptive Mean Thresholding: Adaptive mean thresholding is an image processing technique used to segment an image into regions of foreground and background based on local pixel intensity. In this method, the threshold value for each pixel is computed as the mean intensity of its local neighborhood. By considering the local characteristics of the image, adaptive mean thresholding can effectively handle variations in illumination and contrast across different regions of the image. This technique is particularly useful in scenarios where global thresholding methods like otsu thresholding fail to produce satisfactory results due to uneven illumination or varying background intensities. Adaptive mean thresholding helps enhance the accuracy of object detection, text extraction, and other image analysis tasks by providing better delineation between foreground and background regions.

D. Deployment

The deployed security and surveillance system titled, Trinetra is shown in the Figure....

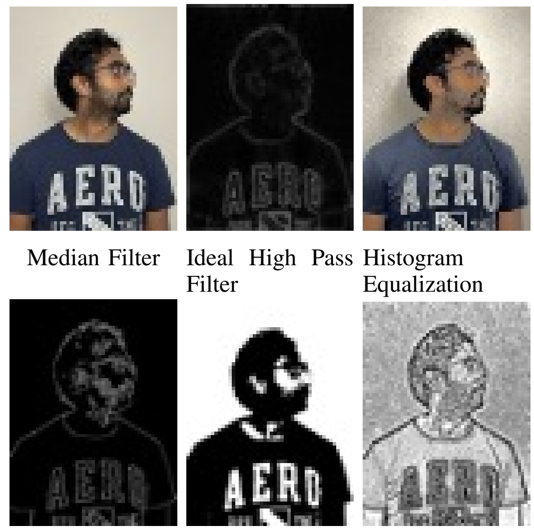
V. Challenges

The following challenges are encountered in developing a security and surveillance system for detecting an intruder using OpenCV and YOLO8 algorithms.

A. Limited Computation Power

1) Challenge: Long Training Time, Reduced model complexity, difficulty in hyper-parameter tuning, limited data augmentation and memory constraints.

2) How to overcome?: Increase the computational power by utilizing high performance clusters, Sabine and Carya of HPE Data Science Institute at University of Houston.



Median Filter | Ideal High Pass Filter | Histogram Equalization

Edge Detection | Otsu Thresholding | Adaptive Mean Thresholding

Fig. 8: Filtered Images

B. Variability in Intruder Appearance

1) Challenge: Intruders can vary significantly in appearance, including factors such as clothing, posture, and lighting conditions. For instance, the presence of an individual wearing a cap posed a detection challenge for the system, illustrating the difficulty in consistently detecting intruders across diverse scenarios.

2) How to overcome?: The dataset utilized for training the model addressed in this paper lacks sufficient size to encompass the broad spectrum of variabilities. To mitigate this issue, enhancing the dataset's variability is essential, incorporating diverse factors such as clothing, headgear, eyeglasses, posture, and background lighting.

C. Ascertaining the optimum thresholds in edge detection filtering

1) Challenge: During the edge detection process, a notable difficulty emerged from manually hard-coding the lower and upper threshold values. Manually adjusting threshold values proved to be a genuine obstacle, considering the variability across different images leading to a decrease in the efficiency and accuracy of the edge detection process.

2) How did we resolve?: Otsu thresholding methods were employed to compute the lower and upper thresholds based on the image intensity. Utilizing the Otsu method alleviated the manual burden of threshold selection, enabling the algorithm to intelligently determine the optimal thresholds for each image.

D. Integration of web stream with YOLO8 trained model

1) Challenge: Unable to integrate the live web stream with trained model. Without a streamlined solution, the real-time detection of intruders was hindered, impacting the effectiveness of the security system.

2) How did we resolve?: The model is trained in Google Colab for efficient computation, and subsequently, the trained model weights are imported as an h5 file extension. This enabled the implementation of the model on a local machine for real-time intruder detection.

E. Grasping the intricacies of YOLO v8

1) Challenge: The YOLOv8 algorithm features a wide array of parameters, and determining the optimal set for implementing this project posed a challenge. Without a thorough comprehension of model parameters, selecting effective ones would be challenging, potentially compromising the performance of intruder detection system.

2) How did we resolve?: Systematically assessed the performance of various parameters, identifying those most suitable for our project's objectives and fine-tuning our YOLO implementation to achieve optimal results.

F. Updating GUI with Intruder Images for Processing

1) Challenge: The challenge was updating the GUI efficiently with the latest intruder image while seamlessly integrating image processing techniques. Without timely updating of the GUI with the most recent intruder image in real-time, the surveillance system was unresponsive, resulting in images not being displayed properly in the GUI, in turn impacted the functionality of the surveillance system GUI. Before initiating an event, the code must check for new folders where intruder images are stored. An event with a watchdog timer is established to monitor the intruder drop folder, refreshing the GUI with the latest frame every 20 seconds.

2) How did we resolve?: To tackle this challenge, we developed an event with a watchdog timer to monitor the intruder drop folder, updating the GUI with the latest frame every 5 seconds. Leveraging the HTML Yield function, we pass the intruder snapshot to the GUI for display. This ensured consistency and efficiency in intruder detection and analysis, as the same image is utilized for implementing image processing techniques.

G. Manual image labeling

1) Challenge: Roboflow, an open-source annotation tool, lacks an automated labeling option. Due to budget constraints, we were unable to acquire a paid version. Consequently, over 15 hours were invested in manually labeling 5000 images in the dataset. However, this manual labeling method may result in misclassification of images into incorrect classes.

2) How to overcome?: Use a paid version of the annotation tool to circumvent manual labeling.

VI. CONCLUSION

The security and surveillance system developed in this project leverages advanced computer vision and deep learning techniques to effectively identify intruders from camera footage. By utilizing the powerful YOLOv8 object detection model and OpenCV, the system can accurately classify individuals into authorized personnel or intruders in real-time.

Our system's comprehensive approach, including dataset preparation, model training and integration of digital image processing techniques, addresses the critical need for robust security measures in various settings, such as homes, offices, and public spaces. The user-friendly GUI interface provides live streaming and displays both the original and filtered images of the most recent intruder, enhancing visibility and aiding in prompt identification.

Despite the challenges encountered during development, such as computational constraints and the need for extensive data collection and annotation, the project successfully demonstrates the potential of leveraging state-of-the-art technologies to enhance security and surveillance capabilities. The integration of techniques like bicubic interpolation, median filtering, histogram equalization, and edge detection ensures that the intruder snapshots remain recognizable under various lighting conditions and backgrounds.

As security concerns continue to escalate, the development of intelligent surveillance systems like the one presented in this project becomes increasingly crucial. Future work could focus on expanding the system's capabilities to include behavior analysis, anomaly detection, and crowd monitoring, further solidifying its role in ensuring public safety and securing vital infrastructure.

VII. FUTURE SCOPE FOR ENHANCEMENT

While the developed security and surveillance system demonstrates promising results, there exist several avenues for further enhancements and expansions.

1) Dataset Expansion and Advanced Hardware: Increasing the dataset size with a larger number of samples can potentially improve model accuracy and robustness. Collecting and annotating more diverse images, including variations in lighting conditions, backgrounds, and angles, can broaden the system's capabilities. Additionally, leveraging advanced GPUs or distributed computing resources can significantly reduce the model training time, enabling more rapid iterations and experimentation.

2) Continuous Model Update: Implementing an automated process to continuously update the model can ensure its adaptability to evolving scenarios. This can be achieved by verifying and annotating new intruder snapshots, which can then be incorporated into the training dataset. This iterative process can help the system learn and adapt to new intruder profiles, improving its overall effectiveness over time.

3) Image Processing Filter Algorithmic Tuning: Exploring algorithmic approaches to automate the tuning of digital image processing filters can optimize their performance for specific use cases. This can involve developing algorithms that analyze the characteristics of the input images and dynamically adjust the filter parameters, such as kernel sizes, thresholds, or cut-off frequencies, to achieve the desired output.

4) Interactive GUI and Visualization: Enhancing the graphical user interface (GUI) to provide greater interactivity and control can improve the user experience and aid in system analysis. Incorporating sliders or similar interface elements can allow users to adjust filter sizes, thresholds, and other parameters in real-time, enabling immediate evaluation of the model's performance under various configurations. Additionally, advanced visualization techniques, such as heat maps or overlays, can provide valuable insights into the system's decision-making process, aiding in debugging and performance monitoring.

5) Integration with Behavior Analysis: Extending the system's capabilities to incorporate behavior analysis and anomaly detection can further elevate its security and surveillance potential. By analyzing the movement patterns, trajectories, and interactions of identified individuals, the system can detect suspicious or anomalous behaviors, enabling proactive response and preventive measures.

6) Scalability and Distributed Deployment: As the system's capabilities expand, ensuring scalability and enabling distributed deployment across multiple locations or large-scale environments will become crucial. This may involve leveraging cloud computing resources, edge computing architectures, or developing efficient data management and synchronization mechanisms to support real-time processing and decision-making across geographically dispersed deployments.

By addressing the above-mentioned future scopes and leveraging emerging technologies, the security and surveillance system can continually evolve, offering enhanced protection, improved situational awareness, and proactive response capabilities to safeguard individuals, organizations, and critical infrastructure.

Acknowledgment

We extend our heartfelt appreciation to the esteemed Professor Dr. Lucy Nwosu, Ph.D., who taught the EDS6397-Digital Image Processing course, as well as Mohit Bushan Varma, the Teaching Assistant for the same course. Their guidance, support, and expertise during this course project were immensely valuable. Their constructive feedback and encouragement significantly contributed to the success of our study.

References

[1] US Burglary Data, https://www.bankrate.com/insurance/homeowners-insurance/house-burglary-statistics/

[2] Wang L, Ouyang W and Wang X, "STCT: Sequentially Training Convolutional Networks for Visual Tracking." Computer Vision and Pattern Recognition IEEE, 2016:1373-1381.

[3] Ju, J., Kim, D., Ku, B., Han, D. K. and Ko, H.. "Online multi-object tracking with efficient track drift and fragmentation handling. " J Opt Soc Am A Opt Image Sci Vis34.2, 2017:280-293.

[4] Chen, Long, Ai, H., Shang, C., Zhuang, Z. and Bai, B.. "Online multi-object tracking with convolutional neural networks." IEEE International Conference on Image Processing IEEE, 2017:645-649.

[5] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[6] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: towards real-time object detection with region proposal networks. Advances in neural information processing systems (pp. 91-99). Neural Information Processing Systems Foundation, Inc.

[7] Roh, M. C., & Lee, J. Y. (2017, May). Refining faster-RCNN for accurate object detection. 2017 fifteenth IAPR international conference on machine vision applications (MVA) (pp. 514-517). IEEE.

[8] 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 (CVPR).

[9] Xu, Renjie, Lin, Haifeng, Lu, Kangjie Cao, Lin & Liu, Yunfei. (2021). A Forest Fire Detection System Based on Ensemble Learning. Forests. 12. 217. 10.3390/f12020217.