Human Detection using YOLOv8 with Squeeze Excitation

Angelita C. Sumera
Dept. of Electrical Engineering
Sam Ratulangi University
Manado, Indonesia
sumeraangelita@gmail,com

Vecky C. Poekoel
Dept. of Electrical Engineering
Sam Ratulangi University
Manado, Indonesia
vecky.poekoel@unsrat.ac.id

Hebron Prasetya
Dept. of Electrical Engineering
Sam Ratulangi University
Manado, Indonesia
hebronprasetya26@gmail.com

***Abstract*— Human detection based on vision systems has become a crucial field in the advancement of information technology. With computer vision systems, we can detect human movements in real-time, which is a significant aspect in security and surveillance applications. One effective architecture for object detection, including humans, is YOLO (You Only Look Once). YOLO has the advantage of fast and accurate detection with a single process, enabling real-time object detection. In this research, we developed the latest YOLOv8 architecture optimized for human detection in various situations and conditions. We also utilized the squeeze-and excitation (SE) attention module to enhance human detection accuracy without significantly increasing parameters. This study aims to create a human detection system capable of achieving high accuracy and can be implemented on Jetson Nano with webcam input. The modified architecture has 4.76 million parameters, mAP 0.548, and GFLOPS 12.**

Keywords*— *CNN; Human Detection; Jetson Nano; Squeeze Excitation; YOLOv8

I. Introduction

One important aspect of technological advancement is the development of human detection systems. In the era of advanced information technology, the use of computer vision to detect objects, particularly humans, has become a significant and compelling area of research. Using vision-based systems, human movement can be detected and monitored in real time [1].

Vision-based human detection refers to the ability to recognize humans under various conditions and scenarios accurately. Detection typically occurs when human activity is present. It is a key process in digital image processing, used to identify specific objects, particularly humans, within a digital image. Human detection has significant potential in various applications, including security surveillance, crowd counting in public areas, and behaviour analysis for security and policy-making purposes. Achieving high accuracy in human detection can provide valuable information across numerous contexts [2].

In response to the increasingly complex challenges associated with human object detection, methods based on artificial intelligence, particularly deep learning techniques, have emerged as the preferred approach. One of the most significant breakthroughs in deep learning is the application of convolutional neural networks (CNNs), which are capable of automatically extracting features from image data. Among the various CNN architectures that have been developed, the YOLO (You Only Look Once) framework has proven to be highly effective for object detection tasks, including human detection. The main advantage of YOLO is its ability to perform detection quickly and accurately in a single-stage process, allowing object detection to be carried out in real time [3]. To optimize the performance of human detection in various complex situations and conditions, recent studies have introduced enhancements to advanced YOLO architectures, such as YOLOv8 [4]. In addition, the squeeze-and-excitation (SE) module is an attention mechanism that has been shown to effectively improve the performance of neural networks [5].

Based on the discussion above, this study aims to develop a YOLOv8 architecture using human detection as the test data. The YOLOv8 architecture is expected to evolve by mimicking the human ability to detect other humans and produce high detection accuracy. This system will be implemented as a surveillance system. The presence of a vision-based security system capable of detecting humans can support various fields, such as safety and monitoring applications [6].

A Convolutional Neural Network (CNN) is a deep learning algorithm capable of training on large datasets containing millions of parameters. It processes 2D dimensional input images by convolving them with filters to extract relevant features and generate the desired output. In this study, CNN models are constructed to evaluate their performance on image recognition and detection datasets [7].

One of the most well-known deep neural network architectures is the Convolutional Neural Network (CNN). Its name originates from the linear mathematical operation known as convolution, which is performed between matrices. A typical CNN consists of several types of layers, including convolutional layers, non-linear activation layers, pooling layers, and fully connected layers. While the convolutional and fully connected layers contain learnable parameters, the pooling and non-linear activation layers do not. CNNs have demonstrated outstanding performance in addressing a wide range of machine learning tasks, particularly in the field of computer vision [8].

The CNN architecture for image classification begins with an input image associated with a specific class. The image is first processed through convolutional layers to extract relevant features. This is followed by the application of the ReLU activation function to introduce non-linearity. The output is then passed through pooling layers, which reduce the spatial dimensions while preserving important features. Finally, the resulting feature maps are fed into fully connected layers, which produce the final classification output.

You Only Look Once (YOLO) is an object detection system based on Convolutional Neural Networks (CNNs). YOLO utilizes a CNN architecture to process the entire image and simultaneously predict both the location and class of

Received: 12 June 2025, Revision: 20 August 2025, Accepted: 24 October 2026

objects in a single step. YOLO adopts a single forward propagation approach, enabling the network to perform object classification and bounding box prediction concurrently [9].

YOLO divides the input image into uniform grids of size $s \times s$. Each grid cell is responsible for performing image classification and predicting bounding boxes for objects whose centers fall within that cell. This process yields class probabilities, object coordinates, and bounding box confidence scores. The object detection workflow in YOLO involves several steps, beginning with feature extraction from the input image. This is followed by a prediction stage that generates bounding boxes and class labels for the detected objects. Each bounding box is defined by four parameters: x, y, w, and h, where x and y represent the coordinates of the center point, and w and h represent the width and height of the bounding box, respectively [10].

In the conventional convolution layer of a CNN, weights are applied uniformly across all channels when generating the output. This means that the same 2D matrix of weights is used for all channels, treating each channel with equal importance. However, the Squeeze-and-Excitation (SE) block introduces an adaptive approach where the importance of each channel is evaluated individually based on its contextual information. Rather than treating all channels uniformly, the SE block dynamically weights each channel according to its specific relevance when generating the output, allowing the network to emphasize more informative features and suppress less useful ones [11].

Therefore, the SE block considers the importance of each channel in contributing to the final result. It provides the flexibility to assign different weights to each channel based on the amount of relevant information it carries. This allows the model to focus more on the most informative channels within a given context, thereby enhancing its ability to extract meaningful features from the input data and improving overall performance [12].

In the initial stage, the SE (Squeeze-and-Excitation) block captures global information from each channel by compressing it into a single scalar value. This is achieved by applying global average pooling (GAP) across the spatial dimensions of each channel, producing a vector of length "*n*", where "*n"* corresponds to the number of channels in the input tensor [13].

After obtaining a vector of size n, it is then passed into a two-layer feed-forward neural network. This network structure is designed to efficiently capture complex dependencies and relationships among the channels. The output of this network is a vector of the same size n, containing n values that represent the learned importance weights for each channel [12].

Object detection is the process of identifying and confirming the presence of specific objects within a digital image or video. This method involves analyzing the features of objects present in the visual data. The primary objective of object detection is to distinguish objects from the background, enabling accurate localization and classification [14].

Object detection is a method that utilizes deep visual analysis to recognize and label objects appearing in images, videos, and other recordings. Object detection models are trained on annotated visual data, enabling them to identify and label objects in new, unseen data. In general, the object detection process can be described as transforming visual inputs into visual outputs with corresponding labels [15].

An essential component of object detection is the bounding box, which precisely delineates the edges of the detected object. Each bounding box is associated with an object label that identifies the type of object, such as a human, car, or animal. Bounding boxes may overlap to accommodate multiple objects within a single image, depending on the model's prior knowledge and detection capabilities [15].

Object detection technology has rapidly advanced through the integration of machine learning algorithms and artificial intelligence. State-of-the-art models, such as YOLO (You Only Look Once), have significantly improved the accuracy and efficiency of object recognition in images and videos. Applications of object detection span a wide range of fields, including security and surveillance, autonomous vehicles, and smartphone applications, demonstrating the considerable potential of this technology in everyday life. [16]

II. Methods

A. Research Methodology

Data collection for the research on Human Detection System Design using YOLOv8 in laboratory rooms was conducted through the following steps:

1. Collecting reference materials and relevant information related to the design of the model to be developed.
2. Preparing supporting devices and equipment necessary for constructing the YOLOv8 architecture.
3. Preparing the COCO-person dataset, focusing specifically on human objects, to be used in training the YOLOv8 model.
4. Configuring the system and programming the training process, including defining hyperparameters and training settings.
5. Evaluating the model's performance during training using the COCO-person dataset, followed by model tuning or adjustments to achieve optimal results.
6. Implementing the trained model and testing its performance using a webcam input.
7. Analyzing the deployed model on the Jetson Nano platform, including measuring the accuracy of its human detection results.
8. Compiling a final project report to document the research process, findings, and outcomes.

B. System Architecture

In the development of a Human Detection System based on a vision system utilizing YOLOv8 enhanced with a Squeeze-and-Excitation (SE) module for video surveillance, a well-structured design concept is essential to ensure that the research outcomes align with the stated objectives. This foundational concept serves as a comprehensive guide for planning the system, detailing the necessary steps and instructions for identifying and integrating the essential supporting components.

During the model training phase, the use of Graphics Processing Units (GPUs) is highly recommended, as they can significantly accelerate the learning process. GPUs are capable of handling parallel computations more efficiently than Central Processing Units (CPUs), thereby reducing training time and enabling researchers to experiment with various model configurations more effectively. After

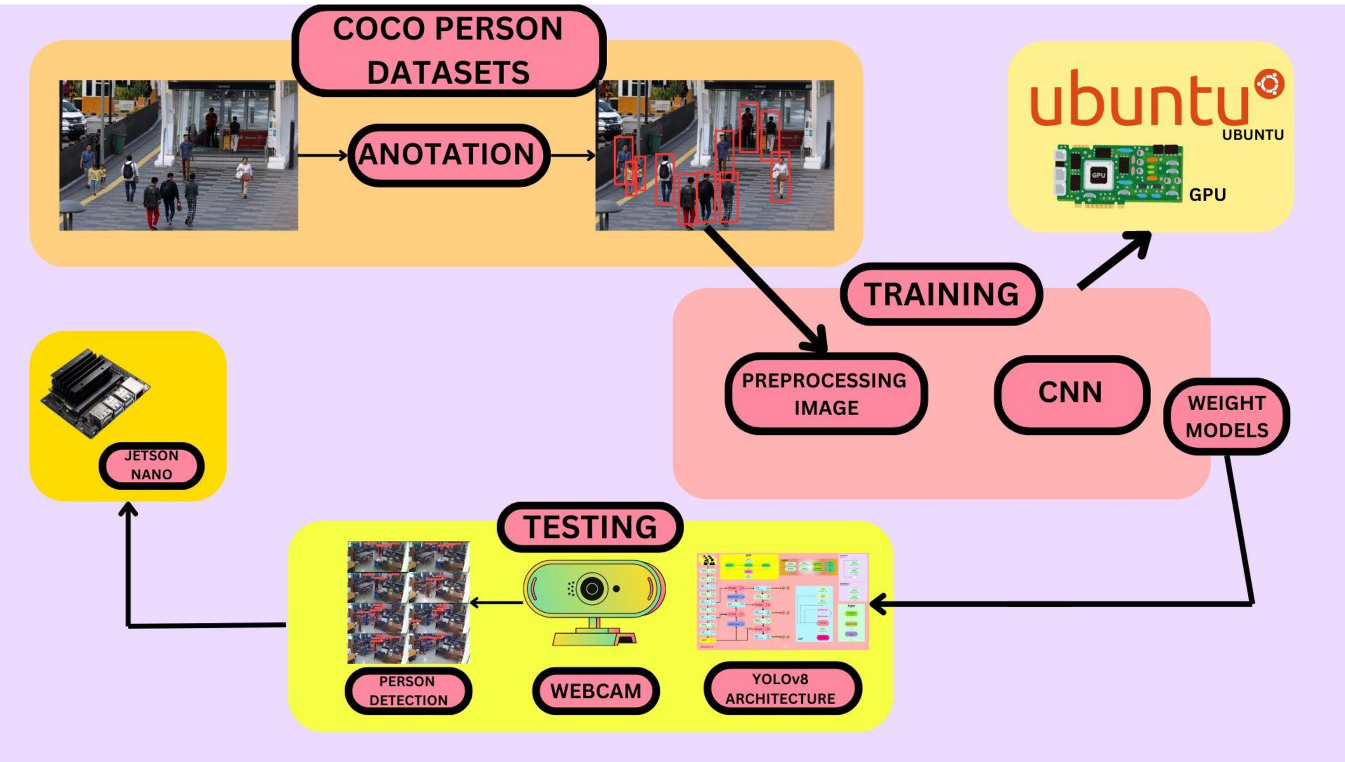


Figure 1. Overview of the training and testing process

completing the training and evaluation stages and obtaining a model with a satisfactory level of accuracy, the next step is hardware deployment. The trained model is implemented on a Jetson Nano device or a standard CPU to perform real-time image processing using a webcam. The Jetson Nano, with its architecture optimized for edge computing and deep learning applications, serves as an efficient platform for real-time inference. Once the model is deployed on hardware such as a CPU or Jetson Nano, it is integrated with a webcam. This integration enables the system to actively and instantly detect and classify humans in real time. The entire process involves several key stages, including data collection and preprocessing, setting up the training environment, conducting model training, performing evaluation and tuning, deploying the model, integrating it with the webcam, and finally conducting testing and validation.

The first step involves collecting relevant data and processing it to ensure it is suitable for training. After the data has been prepared, the training environment must be configured. This includes setting up the hardware, selecting the appropriate GPU, and installing the necessary software and libraries, including the YOLOv8 framework and the Squeeze and Excitation module. During the training phase, the processed data is input into the YOLOv8 model, allowing the model to learn through repeated iterations. In each iteration, the model improves its ability to recognize patterns that indicate the presence of humans in images or videos.

The use of GPUs at this stage is essential for accelerating the training process. Once the initial training is completed, the model is evaluated to assess its accuracy. If the evaluation results do not meet the expected criteria, further tuning is performed by adjusting the training parameters or incorporating additional representative data. After the model achieves the desired level of accuracy, it is then implemented on suitable hardware, such as a Jetson Nano or a standard CPU.

This process includes converting the model to ensure compatibility with the target device and preparing an appropriate runtime environment. Once deployed, the model is integrated with a webcam to enable real-time human detection. This allows the webcam to continuously process and analyze visual input, providing immediate predictions regarding the presence of humans in the surrounding environment. The final stage involves testing and validating the implemented model under various real-world conditions to ensure it functions correctly and delivers accurate results in everyday scenarios. By following this structured approach, the research aims to develop a reliable, effective, and efficient human detection system suitable for video surveillance applications in environments that require continuous and real-time monitoring. Throughout the training and testing phases, several technical challenges were encountered. These challenges extended beyond model and hardware optimization and included the need to manage diverse and complex datasets. The dataset used in this research was designed to encompass a wide range of lighting conditions, viewing angles, and situational contexts. This diversity was essential to ensure the model could maintain high detection accuracy across various scenarios, including challenging situations such as low light or unusual camera angles.

In addition, it is essential to ensure that the model performs accurately not only under test conditions but also generalizes well to real-world data that was not seen during training. To achieve this, the application of data augmentation and cross-validation plays a critical role in the model development process. Data augmentation increases the diversity of the training dataset by introducing variations, thereby reducing the need for manual data collection. Meanwhile, cross-validation helps detect and mitigate overfitting, ensuring the

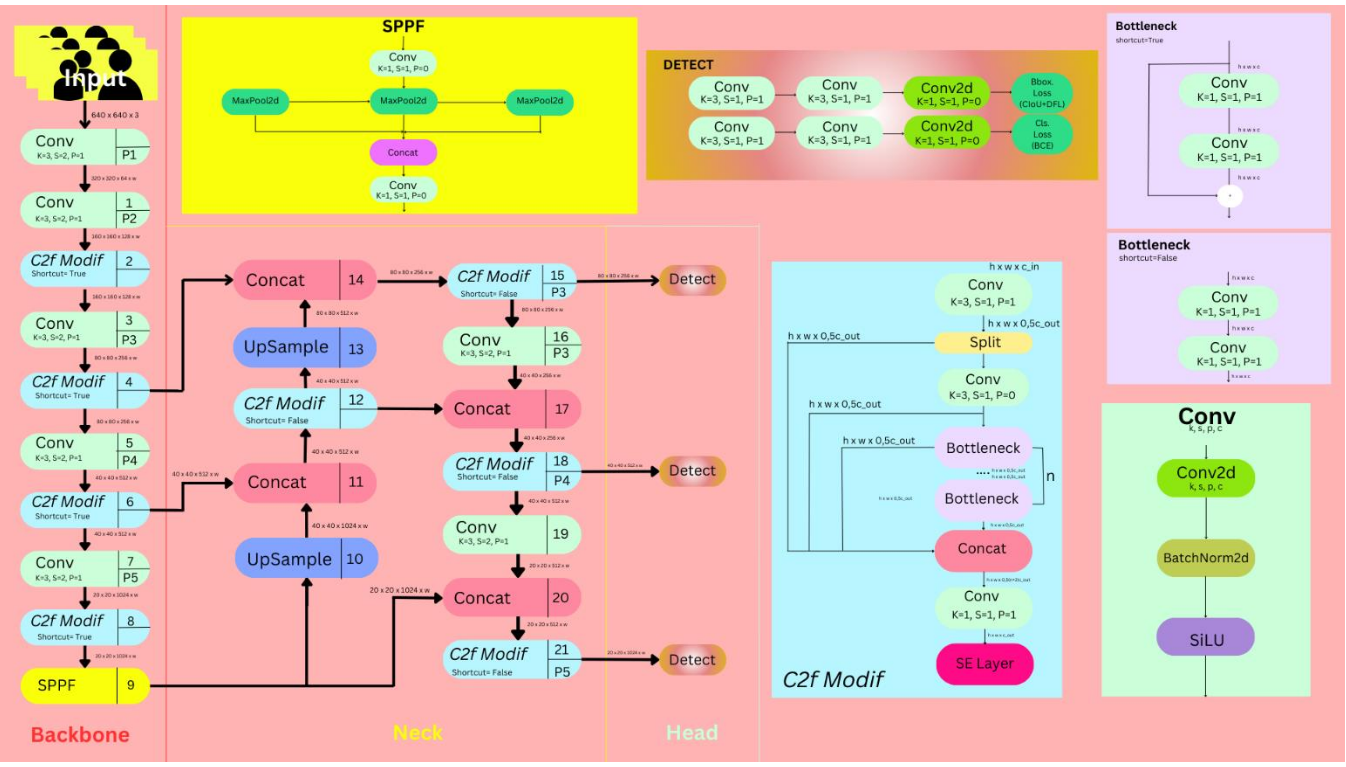


Figure 2. Architecture of the modified YOLOv8 model

model maintains robust performance across different data subsets. Therefore, this research emphasizes not only the development of a sophisticated detection model but also adopts a comprehensive and methodical approach to ensure that the model is reliable and effective for real-world video surveillance applications.

Figure 1 illustrates the overall research workflow, which is divided into two main stages: the training stage and the testing stage. In the training stage, the process begins by inputting the dataset, which is sourced from the MS COCO dataset. Following this, an image preprocessing step is conducted to prepare the data for analysis by the system. This preprocessing involves operations such as noise reduction, resizing, contrast enhancement, and color normalization.

TABLE I. PERFORMANCE COMPARISON OF THE PROPOSED MODEL WITH OTHER ARCHITECTURES

Architecture	Params	Gflops	mAP:50	mAP50:95	EPOCH
YOLOv8	3,011,043	8,2	0,761	0,530	100
YOLOV8-MODIFY	4,768,435	12,0	0, 778	0,548	100
Pd-NET	6,954,838	11,9	0,724	0,524	200

These steps are crucial for improving the quality and compatibility of the image data with the model. Once preprocessing is complete, annotations are applied to label instances of the 'Person' class within the dataset. The annotated data is then used to train a Convolutional Neural Network (CNN) based on the YOLOv8 architecture. This stage enables the model to learn and optimize its accuracy in detecting human objects. The next stage is the testing phase, which begins with the use of a webcam to capture real-time images. These images are input into the trained CNN model, which processes them based on the knowledge acquired during training. The model then predicts the presence of human objects with the learned accuracy level. This stage represents the final phase of the research, aimed at evaluating detection accuracy by analyzing the performance metrics generated by the developed CNN model. In this context, the Squeeze-and-Excitation Network (SENet) offers an effective mechanism for enhancing relevant information within Convolutional Neural Networks (CNNs) by assigning greater weights to the most significant features. The incorporation of the Squeeze-and-Excitation attention module significantly improves the network's capacity to extract fine-grained features, thereby enhancing the performance of various image processing tasks, particularly object detection. The integration of the Squeeze-and-Excitation (SE) module in vision-based human detection is motivated by its ability to enhance object detection accuracy and provide adaptive feature selection. This attention mechanism enables the model to dynamically focus on the most informative features within an image. By incorporating the SE module, the object detection model can optimize the feature extraction process more effectively. The SE module operates through two primary stages: the "squeeze" stage, which condenses spatial information into a channel-wise descriptor, and the "excitation" stage, which adaptively recalibrates the importance of each feature channel based on its relevance.

This mechanism allows the model to be more responsive to variations in image data, such as lighting changes, and more adaptive to diverse image contexts and tasks, resulting in improved performance across various scenarios. Moreover, SENet enhances the YOLO model's ability to capture contextual information and understand inter-object relationships. By emphasizing critical features while suppressing less relevant or noisy information, the SE module helps reduce overfitting and improves the model's generalization capability. An additional advantage is the modular nature of SENet, which allows it to be integrated into a wide range of CNN architectures, including YOLO, without requiring significant changes to the network's original structure.

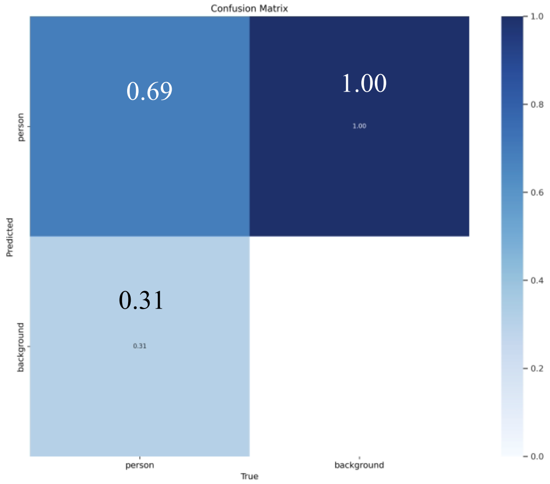


Figure 3. Confusion Matrix

TABLE II. PERFORMANCE COMPARISON OF THE PROPOSED MODEL WITH OTHER ARCHITECTURES

Architecture	Params	Gflops	mAP:50	mAP50:95	EPOCH
YOLOv8	3,011,043	8,2	0,761	0,530	100
YOLOV8-MODIFY	4,768,435	12,0	0, 778	0,548	100
Pd-NET	6,954,838	11,9	0,724	0,524	200

In this experiment, the operating system used was Ubuntu 18.04.6, supported by 17 GB of RAM. The system is powered by an Intel Core i5-2320 CPU and equipped with an NVIDIA GeForce GTX 1080 graphics card with 17 GB of memory.

The proposed architecture was implemented using the PyTorch framework, and CUDA 11.4 was utilized to enable GPU acceleration.

The training process was conducted for more than 100 epochs to ensure that the model underwent sufficient iterations to effectively learn to detect human objects. The input image resolution for both training and evaluation was set to 640×640 pixels. A learning rate of 0.001 was used, and the batch size was set to 64.

A loss function was employed to compute the difference between the predicted bounding boxes and object classes and the corresponding ground truth values. The model's performance was evaluated on an NVIDIA Jetson Nano device equipped with 4 GB of 64-bit LPDDR4 RAM during the testing and implementation stages.

In Table I, the researchers compare training results across several architectures. The YOLOv8n architecture achieved a mean Average Precision (mAP) of 0.530. In contrast, the proposed architecture, developed using the PyTorch library, demonstrates improved performance. As shown in Table I, the YOLOv8n-SE architecture achieved an mAP of 0.548, outperforming both the original YOLOv8n and the Pd-Net models.

Figure 2 illustrates the modified YOLOv8 architecture, in which the C2f module has been enhanced, while the backbone, neck, and head components remain consistent with the original architecture. In the modified C2f module, an additional convolutional layer with a 3 × 3 kernel is introduced. Furthermore, a Squeeze-and-Excitation (SE) attention module is integrated at the end of the C2f module. These enhancements aim to improve the model's capacity to learn detailed features from the training dataset. The added convolutional layer contributes to capturing more refined spatial information, whereas the SE module emphasizes salient features while suppressing irrelevant ones. Together, these modifications are designed to increase the accuracy and robustness of the object detection performance.

III. Results and Discussions

Based on Figure 3, which presents the Confusion Matrix, the model's performance in recognizing specific classes, such as the "person" class, can be analyzed effectively. The Confusion Matrix provides insight into the accuracy of classification by showing how well the model distinguishes between correct and incorrect predictions. In this study, the model achieved a detection value of 0.69 for the "person" class, indicating a moderate level of accuracy in correctly identifying human objects.

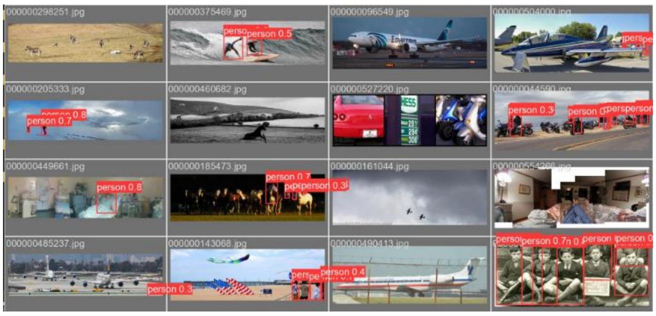
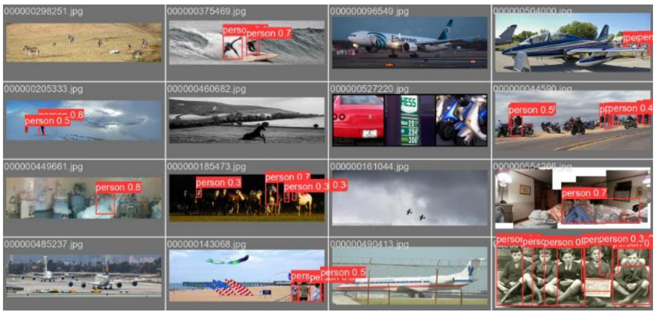


Figure 5. Visualization of object detection results during validation using the modified YOLOv8n architecture



By examining this matrix, it is possible to assess the distribution of predictions, including true positives, false positives, and false negatives. This analysis is essential for understanding the strengths and limitations of the model in object detection tasks. The score of 0.69 suggests that while the model performs reasonably well, there is still potential for further improvement to enhance its accuracy and reliability in real-world applications.

Table II presents a comparison of accuracy between the original YOLOv8 architecture and the modified YOLOv8, demonstrating a notable improvement in performance. The

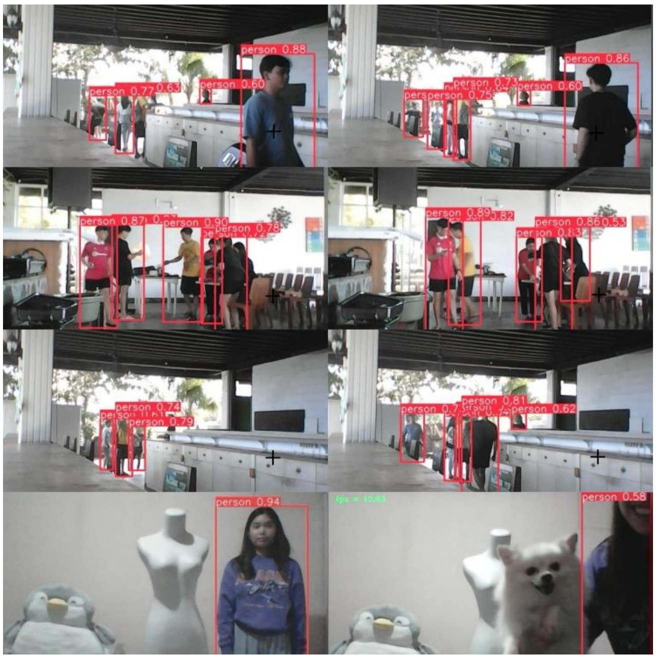


Figure 6. Output of real-time object detection using the model on a laptop device

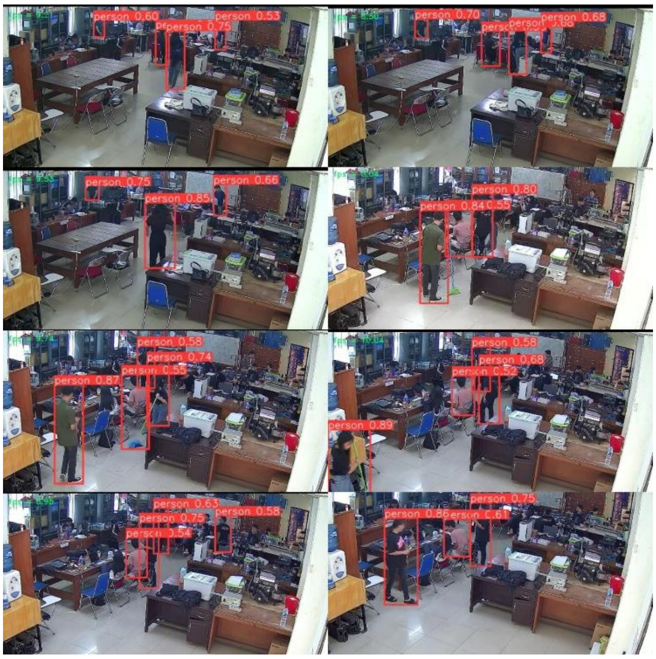


Figure 7. Output of real-time object detection using the model on the Jetson Nano device

original model achieved an accuracy of 0.530, while after modification, the accuracy increased to 0.548. This improvement indicates that the applied modifications successfully enhanced the model's detection capability, allowing it to recognize objects more accurately than the original version. These results confirm the effectiveness of the modifications in improving YOLOv8's overall performance.

Figure 4 shows the results of the unmodified YOLOv8n model. At the top, some persons are not detected properly, as indicated by the areas circled in white. This indicates that the model's accuracy is still not optimal. Although the model can detect humans, some shortcomings must be addressed to ensure all persons in the image are correctly identified. This demonstrates the need for further refinement of the model to improve its accuracy.

Figure 5 shows the results of the modified YOLOv8n architecture. It can be seen that after the modification, the previously undetectable person is now successfully detected. For example, in the circled area, the person could not be detected properly before the modification. However, after the improvements, as shown in Figure 5, the person is accurately identified. This demonstrates that the enhancements and adjustments made to the model effectively improved its detection capability and accuracy. In Figure 6 presents the accuracy test results in a real-world scenario using a laptop. The test shows that the model performs well with a laptop camera, accurately distinguishing between objects classified as 'Person' and 'Not Person'. The testing involved various scenarios to ensure that detection is both accurate and reliable. One such test included detecting a statue.

The test results show that statues are not detected as 'Person', which indicates the model performs well in distinguishing between humans and non-human objects. This is important because it demonstrates the model's ability to avoid false positive detections on non-human objects.

Another test was conducted using a dummy, which was also not detected as a 'Person'. This confirms that the model does not rely solely on the general shape of the human body but can recognize additional features that differentiate humans from objects with human-like shapes.

In addition, tests were carried out on dogs to evaluate how the model distinguishes animals from humans. The results showed that the dog was not detected as a 'Person', confirming the model's strong ability to differentiate between humans and animals. These findings demonstrate that the model maintains high accuracy in detecting humans and is not influenced by non-human objects such as statues, dolls, or animals.

In Figure 7 presents the results of testing conducted with a webcam and a Jetson Nano, which also show very accurate predictions of 'Person' and 'Not Person'. This test was performed in the Control Lab of Sam Ratulangi University, Manado, during the day, providing favorable lighting conditions.

During the test, the system successfully detected several individuals. This demonstrates that the model performs well in real-world conditions, not only in controlled environments. The high detection accuracy confirms that the model is reliable for surveillance tasks in practical settings.

The webcam used in the test was positioned at the top corner of the laboratory entrance, a strategic location for capturing the movement of people entering and exiting the room. This placement enables the system to operate effectively as a surveillance camera, providing wide coverage and ensuring that all individuals passing through the entrance are detected.

In addition, testing conducted during daylight with natural lighting demonstrates that the system performs optimally under favorable lighting conditions. However, for broader surveillance applications, further testing is needed under various lighting scenarios, including nighttime and artificial lighting, to ensure detection accuracy remains high in all environments. The positive results obtained from the test at the Control Lab of Sam Ratulangi University, Manado, indicate the promising potential of this system for real-world surveillance applications.

IV. Conclusions and Futher Work

A. Conclusions

Based on the research and discussion on vision-based human detection using YOLOv8 with the Squeeze Excitation module for video surveillance, several conclusions can be drawn. First, the vision-based human detection system using the YOLOv8n-Modify architecture has been successfully implemented. Second, the YOLOv8n model integrated with the Squeeze Excitation module contains 4,768,435 parameters and has a GFLOPS value of 12.0; therefore, modifications were necessary to optimize the model for deployment on the Jetson Nano, achieving a mean Average Precision (mAP) of 0.548 with the same GFLOPS. Third, the modified YOLOv8n model with the Squeeze Excitation module achieved an accuracy of 0.548, surpassing the original architecture's accuracy of 0.530. Fourth, implementation on the Jetson Nano demonstrated reliable performance with a detection speed of approximately 10 frames per second. Finally, this architecture exhibited competitive accuracy compared to other models tested on different datasets.

B. Future Work

This architecture has the potential for further enhancement by integrating advanced attention modules to significantly improve detection accuracy. By leveraging attention mechanisms, the model can better focus on salient features and enhance its understanding of contextual information and object relationships within images or video frames. Additionally, further optimization can be performed to tailor the model for deployment on Jetson Nano devices, improving accuracy while maintaining computational efficiency. Such improvements will increase the model's reliability and effectiveness for vision-based human detection, particularly in real-time surveillance and monitoring applications across diverse and dynamic environments.

Acknowledgment

The authors express their sincere gratitude to the AIVISION research team for their invaluable support and expertise in computer vision and deep learning. Their provision of computational resources and constructive feedback greatly contributed to the experimental work, methodological refinement, and the overall quality of this manuscript.

References

[1] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, "A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition," *Drones*, vol. 7, no. 5, Art. no. 5, May 2023, doi: 10.3390/drones7050304.

[2] K. Liu, "STBi-YOLO: A Real-Time Object Detection Method for Lung Nodule Recognition," *IEEE Access*, vol. 10, pp. 75385–75394, 2022, doi: 10.1109/ACCESS.2022.3192034.

[3] M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," Machines, vol. 11, no. 7, Art. no. 7, Jul. 2023, doi: 10.3390/machines11070677.

[4] H. Lou *et al.*, "DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor," *Electronics*, vol. 12, no. 10, Art. no. 10, Jan. 2023, doi: 10.3390/electronics12102323.

[5] J.-S. Kim *et al.*, "E-HRNet: Enhanced Semantic Segmentation Using Squeeze and Excitation," *Electronics*, vol. 12, no. 17, Art. no. 17, Jan. 2023, doi: 10.3390/electronics12173619.

[6] B. Yang, M. Tang, S. Chen, G. Wang, Y. Tan, and B. Li, "A vehicle tracking algorithm combining detector and tracker," *EURASIP J. Image Video Process.*, vol. 2020, no. 1, p. 17, Apr. 2020, doi: 10.1186/s13640-020-00505-7.

[7] V. C. Poekoel et al., "North Sulawesi Single Local Fruit Detection Using Efficient Attention Module Based on Deep Learning Architecture," Jurnal Nasional Pendidikan Teknik Informatika: JANAPATI, vol. 12, no. 2, pp. 213–222, Jul. 2023, doi: 10.23887/janapati.v12i2.54754.

[8] "Convolutional neural network," *Wikipedia*. Apr. 25, 2024. Accessed: Apr. 25, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network&oldid=1220667944.

[9] M. Sohan, T. Sai Ram, and Ch. V. Rami Reddy, "A Review on YOLOv8 and Its Advancements," in *Data Intelligence and Cognitive Informatics*, I. J. Jacob, S. Piramuthu, and P. Falkowski-Gilski, Eds., Singapore: Springer Nature, 2024, pp. 529–545. doi: 10.1007/978-981-99-7962-2_39.

[10] M. Hussain, "YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection," *Machines*, vol. 11, no. 7, Art. no. 7, Jul. 2023, doi: 10.3390/machines11070677.

[11] Q. Hu, H. Liu, Z.-H. Chen, H.-X. Hu, Y. Zhang, and Z.-Y. Hu, "Pedestrian target detection on high-speed pavement using improved YOLOv5," in *Proceedings of the 15th International Conference on Digital Image Processing*, in ICDIP '23. New York, NY, USA: Association for Computing Machinery, Oct. 2023, pp. 1–10. doi: 10.1145/3604078.3604128.

[12] T. Samavati, "Squeeze-and-Excitation explained," Medium. Accessed: Apr. 29, 2024. [Online]. Available: https://medium.com/@tahasamavati/squeeze-and-excitation-explained-387b5981f249.

[13] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Comput. Appl.*, vol. 35, no. 28, pp. 20939–20954, Oct. 2023, doi: 10.1007/s00521-023-08809-1.

[14] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768. Accessed: Apr. 25, 2024. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Liu_Path_Aggregation_Network_CVPR_2018_paper.html.

[15] "Object detection," *Wikipedia*. Nov. 26, 2023. Accessed: Apr. 29, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Object_detection&oldid=1186971351.

[16] F. Chen, M. Deng, H. Gao, X. Yang, and D. Zhang, "NHD-YOLO: Improved YOLOv8 using optimized neck and head for product surface defect detection with data augmentation," *IET Image Process.*, vol. n/a, no. n/a, doi: 10.1049/ipr2.13073.