JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 8 Number 2 | December 2023: 261-270
DOI: 10.15575/join.v8i2.1196

Optimizing YOLOv8 for Real-Time CCTV Surveillance: A Trade-off Between Speed and Accuracy

Muhammad Rizqi Sholahuddin[1], Maisevli Harika[2], Iwan Awaludin[3], Yunita Citra Dewi[4], Fachri Dhia Fauzan[5], Bima Putra Sudimulya[6], Vandha Pradiyasma Widarta[7]
[1,2,3,5,6]Department of Computer and Informatics, Politeknik Negeri Bandung, Bandung 40559, Indonesia
[4]Department of Mechanical Enginering, Politeknik Negeri Bandung, Bandung 40559, Indonesia
[7]Department of Information System, Pukyong National University, Busan, South Korea

Article Info

Article history:

Received September 25, 2023
Revised November 26, 2023
Accepted December 04, 2023
Published December 28, 2023

Keywords:

YOLOv8
CCTV Surveillance
Model Pruning
Face Detection
Streamlit

ABSTRACT

Real-time video surveillance, especially CCTV systems, requires fast and accurate face detection. Object detection models with slow inference times are ineffective in real-time. This study aims to improve the inference speed of the YOLOv8 model, a preeminent object detection framework known for its accuracy and expeditiousness. The method used in this research focuses on pruning the model's architecture, particularly the P5 head section, which detects larger objects. According to Bochkovskiy's 2020 research, this modification enhances the model's performance specifically for medium and small objects depicted in CCTV footage. The standard YOLOv8 model and its modified version were compared for inference time, mean Average Precision (mAP), and model weight. This research found that the pruned YOLOv8 model demonstrated a 15.56% reduction in inference time, from 4.5 ms to 3.8 ms, accompanied by decrease in model weight. The advantages mentioned above are offset by a 1.6% decrease in mean average precision. This research contributes to the advancement of object detection technology by showcasing the architectural modifications' efficacy. These changes make the model faster and lighter, making it suitable for real-time surveillance. The accuracy trade-off is minimal. The implications of these findings are crucial for implementing efficient object detection systems in CCTV surveillance. These findings also lay the groundwork for future research to improve such systems' speed-accuracy trade-off.

Corresponding Author:

Maisevli Harika,
Department of Computer and Informatics, Politeknik Negeri Bandung, Indonesia,
Jl. Gegerkalong Hilir, Ciwaruga, Kec. Parongpong, Kabupaten Bandung Barat, Jawa Barat 40559
Email: maisevli@polban.ac.id

1. INTRODUCTION

In recent years, computer vision has experienced a notable increase in the widespread use of face detection and recognition systems. The recognition of faces is of utmost importance in systems specifically designed for human identification. This technology is widely utilized in various domains, encompassing academia for attendance monitoring, as well as diverse sectors such as security, biometrics, law enforcement, entertainment, and personal safety. The primary focus of these applications is to provide functionalities of real-time surveillance and the tracking of individuals.

Real-time surveillance refers to the continuous monitoring and analysis of activities and relevant data through a system of multiple Closed-Circuit Television (CCTV) cameras. This configuration, combined with a set of algorithms, is effective in identifying, tracking, and recognizing individuals in real-time video streams [1].

Significant advancements have been observed in object detection, primarily due to the rapid progress in artificial neural networks, particularly within the domain of neural networks. As a result, there have been notable advancements in detection algorithms that utilize deep learning techniques in recent years [2]. The recent progress in this field has garnered significant interest among researchers, particularly concerning YOLO networks, which have demonstrated notable advancements in object localization and recognition.

The development of the YOLO framework took place significantly. In July 2022, WongKinYiu and AlexeyAB released YOLOv7 with state-of-the-art claims, beating the previous version of YOLO [3]. Then, in January 2023, Ultralytics released YOLOv8, which is even better than YOLOv7 [4]. This makes the proposed research use YOLOv8 by Ultralytics.

Based on Purwita's research, two new neural network architectures were implemented, namely Feature Pyramid Network (FPN) and Path Aggregation Network (PAN). In addition, there is a labeling tool that can simplify the data annotation process. The features of the YOLOv8 labeling tool are automatic labeling, shortcut labeling, and customizable hotkeys [5].

When using FPN, the input image's spatial resolution gradually decreased while the number of feature channels increased. As a result, a feature map is created that can detect objects at various scales and resolutions. The PAN architecture, through skip-connections, integrates features from multiple network layers. As a result, the network has improved in its ability to capture features across a range of input sizes and resolutions, which is crucial for object detection [6].

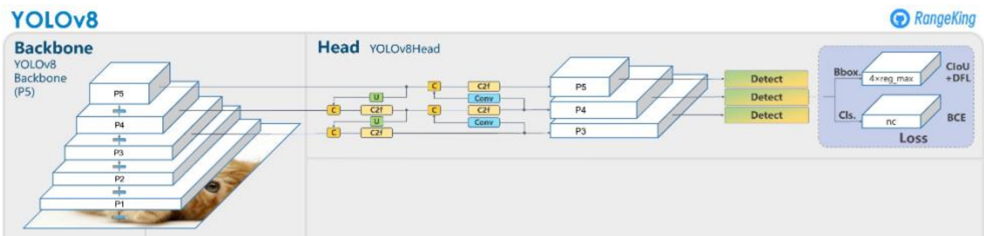

Figure 1. YOLOv8 Architecture [6]

The architecture of YOLOv8 (Figure 1) dramatically affects the outcome metrics of the trained model. The architecture of YOLOv8 can be changed by adding convolution layers or reducing existing layers. In the research of Li et al. (2023), providing an additional layer of Bi-PAN-FPN at the neck of the YOLOv8 architecture can reduce misdetection and missed detection [7]. In this study, the architecture will be pruned at the right part to reduce the inference time of the YOLOv8 model.

The trained YOLOv8 model needs to be deployed using a GUI-web-based app, in this context, streamlit. Streamlit is a web-based library for flexible and interactive AI models [8]. "Streamlit has numerous applications in creating machine learning as a service. For example, it can facilitate user authentication without a backend [9]. In this research, web-based app users with Streamlit used to load data and models to infer images, videos, or live streams.

Pruning the YOLO architecture can impact many metrics, such as inference speed, mAP accuracy, and weight size. In Bochkovskiy's research, simplifying the parameter structure of YOLOv4 to create YOLOv4-tiny is done to speed up the training process and inference time. The YOLOv4-tiny model has a decrease in accuracy (mAP) up to 26.8% of YOLOv4's mAP [10].

In this study, the YOLOv8 architecture will be simplified by reducing the structure of the YOLOv8 architecture to get a faster inference time by paying attention to architectural changes in order to minimize the impact on the decrease in mAP.

2. METHOD

Machine learning algorithms and frameworks for object detection are numerous and continue to evolve. However, the YOLO (You Only Look Once) framework provides the highest accuracy among other object detection, which makes it a state-of-the-art that beats other algorithms [1]. YOLO has various versions created by various publishers. Currently, the best YOLO version is YOLOv8 which provides the highest mAP accuracy value among other YOLO versions [4].

The proposed method consists of labeling the dataset as the ground truth, modifying the architecture of YOLOv8, and the comparison stage between the pre-trained YOLOv8 model and the modified model. Dataset labeling is done in a Python environment using labelimg. This stage is labels the image data in the parts that are faces and persons. The architecture modification stage is a pruning stage carried out to improve the speed performance of YOLOv8 in performing face and person detection so that face recognition becomes more effective. The comparison stage between the pre-trained model and the modified model is carried out to conclude the success of the experiments carried out.

2.1. Preprocessing Data

Data obtained is in the form of CCTV video recording data, so to make it easier to do the data labeling process, the video recordings are converted into images using a Video LAN Client (VLC). VLC is a free and open-source media player and processing framework [11]. VLC was developed using FFmpeg and other components commonly used in multimedia, stream media players, and streaming processors [12]. In the process of converting video into an image, the frame rate is adjusted to the frame rate of the video, where in this case, the video frame rate is 15.17, as shown in Figure 2. Then a framerate of 15 is also used to extract the video data.

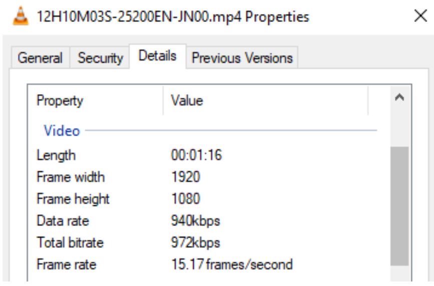

Figure 2. Properties Video CCTV

The resulting image data transformation process is carried out data filtering because there is data that does not have an object image in it, or the image is too blurry. From the 3-day data CCTV, the hours between 7 am to 5 pm were selected because these hours are active hours. Around 10.000 image data is produced, then 1000 data that has objects in it and clear images are selected for labeling.

2.2. Dataset Labelling

Dataset labeling is carried out on a dataset of frames extracted from video recordings from the CCTV Department of Computer Engineering at Bandung State Polytechnic. The images used as datasets are 1000 images. The formation of the dataset as ground truth for the model is done by labeling using a bounding box (illustrated by Figure 3, which is done using a Python framework, namely labeling [13]. The ground truth created from the labeling stage is the label of the face and person of each existing image data.


Figure 3. Labeled Image (Yellow: Face, Purple: Body)

 p-ISSN: 2528-1682
e-ISSN: 2527-9165

The labeled dataset is then divided into two parts, namely training data and validation data, with a ratio of 80:20. i.e., 80% training and 20% validation data [13]. After the data preprocessing and labeling stages, the resulting dataset includes 985 images tagged with 'person' and 538 tagged with 'face'. Person tag data in training is 424 images and validation is 114, while person tag data in training is 791 images and validation is 194 images. The distribution of those training and validation data available in Figure 4.

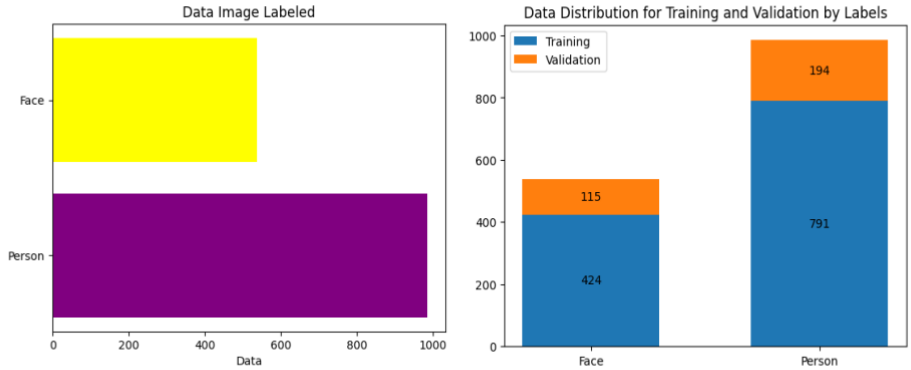

Figure 4. Classes Distribution (Left: Total Data Labeled, right: Data Distribution Training and Validation)

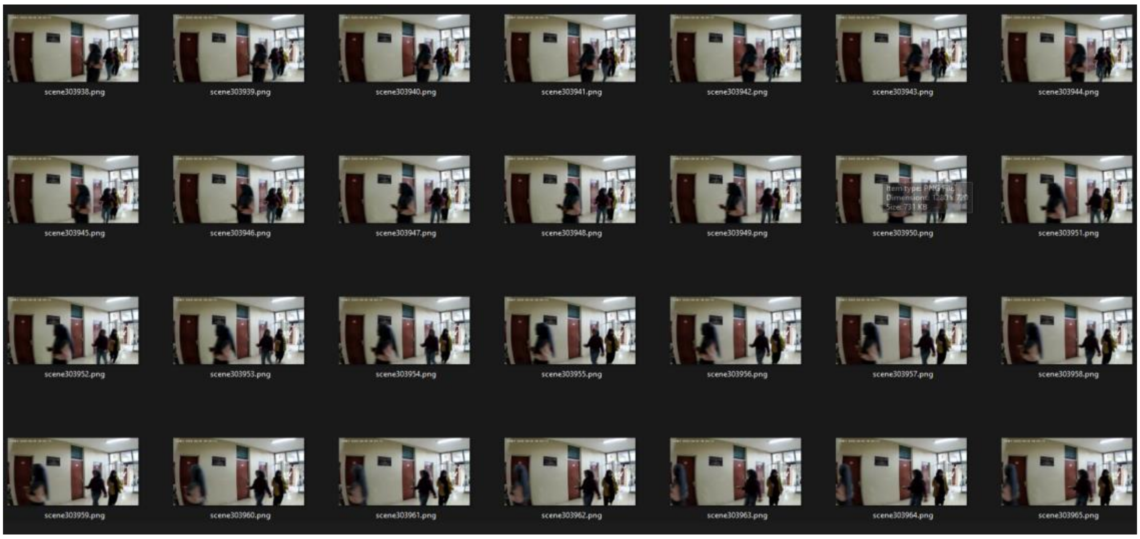

Figure 5. CCTV Recording Dataset Sample

2.3. *Proposed YOLOv8 Architecture*

YOLOv8 has an architecture with three detection heads [14]. Each head performs detection for each file size. The first head, P3, is responsible for detecting small objects, the second head, P4, is responsible for detecting medium objects, and the third head, P5, is responsible for detecting large objects. This experiment detects objects on CCTV footage (see Figure 5), where the face object will look relatively small, and the human body object will look medium. This makes it possible to prune the architecture of the head in charge of large object detection (P5) so that the model focuses on P4 and P3. One way to improve the performance of YOLO is to prune the architecture [15]. In this experiment, the P5 head architecture of YOLOv8.

Pruning P5 head aims to obtain higher speed performance for small and medium object detection because it does not need to pass the processing stage for large objects. In addition, the specific pruning in P5 aims to minimize the effect of the significant decrease in mAP accuracy since it does not interfere with heads P4 and P3. Basis of YOLOv8 is available in Figure 6, while the proposed architecture available in Figure 7.

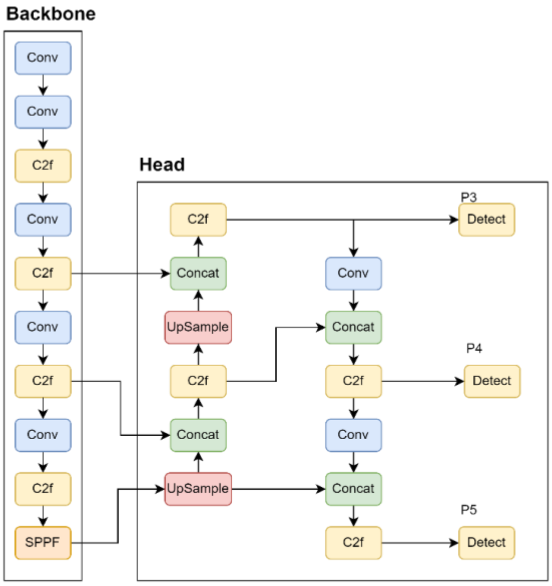

Figure 6. Standard YOLOv8 Architecture Layer

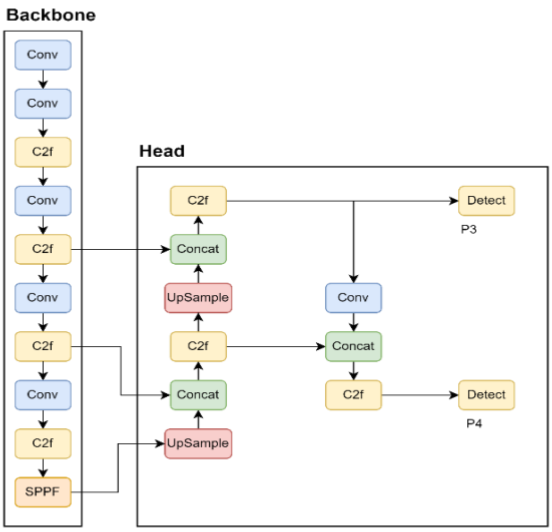

Figure 7. Proposed YOLOv8 Architecture Layer

2.4. Recall

Recall or sensitivity value is a value that calculates the percentage of true positives (TP) from all available precision values (ground truth). Recall calculates the model's ability to detect the correct value, with the formula as in (1) [16]:

$$Recall = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Negatives\ (FN)} \tag{1}$$

2.5. Precision

Precision or confidence value is the calculated value of the true positives (TP) percentage of all predictions made. Precision does not determine the percentage error of predictions not made. Only predictions made are considered by this metric, with the formula as in (2) [16]:

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)} \tag{2}$$

p-ISSN: 2528-1682
e-ISSN: 2527-9165

2.6. Mean Average Precision

Mean Average Precision (mAP) is the average value of the average precision (AP) for the available classes. AP is the area under the curve of precision and recall values. AP evaluates the precision and recall curves based on a single scalar value. The AP value is high if the precision and recall values are high, and vice versa. AP values range from 0 to 1 [17], with the formula as in (3) [18]:

$$Average\ Precision\ (AP) = \int_0^1 Precision(t)d[Recall(t)] \quad (3)$$

The AP value will determine the mAP, the higher the AP value, the higher the mAP value. The mAP value calculates the average AP value of all available classes. The mAP value has the following formula as in (4) [19]:

$$Mean\ Average\ Precision\ (mAP) = \frac{1}{n}\sum_{i=1}^{n} AP_i \quad (4)$$

2.7. Model Training

There are two models that need to be trained: standard YOLOv8 and modified YOLOv8. The standard YOLOv8 architecture configuration is taken from YOLOv8 ultralitycs GitHub, while the modified YOLOv8 architecture configuration is created by removing unnecessary layers from the standard configuration. Both models were trained with 200 epochs and other standard hyperparameters of YOLOv8 (Table 1) .

Table 1. Training Hyperparameters

Hyperparameter	Value
Epoch	200
Initial Learning Rate	0.01
Final Learning Rate	0.01
Optimizer	‘auto’
Validate	True

2.8. Training Result Comparison

This section meticulously compares the performance metrics of the trained YOLOv8 models to assess the efficacy of the proposed modifications. The evaluation focuses on three critical metrics: accuracy, speed, and model size, as these are pivotal in determining the practicality of the models in real-time CCTV face detection applications.

For a comprehensive analysis, each of these metrics is compared across different versions of the YOLOv8 model, including the original and pruned versions. Charts, tables, and graphs visually represent the data, allowing for more straightforward interpretation and comparison. Additionally, a statistical analysis is conducted to determine the significance of the observed differences in performance metrics.

The outcome of this comparison will conclusively reveal whether the pruning approach applied to YOLOv8 effectively balances speed and accuracy, thereby validating its suitability for real-time face detection in CCTV surveillance systems. This evaluation will not only inform the effectiveness of the proposed method but also contribute to future developments in the field of object detection.

2.9. Model Integration with Streamlit

The model deployment stage is done after training the model and is ready for use. The YOLOv8 model is integrated with the web-based GUI frontend, namely streamlit. Streamlit is an open-source Python coding framework for building web applications or “web apps”. It is now being used by researchers to share large data sets from published studies and other resources [20]. The streamlit dashboard is used to perform the data inference process. There are two options, namely image files, and video files. One of the advantages of streamlit is that it is a paradigm that fully uses Python, so it can be an enabler for the YOLO project, which also uses Python [21].

3. RESULT AND DISCUSSION

This study involves the training and validating of the YOLOv8 model using a carefully curated dataset derived from CCTV video frames. The initial dataset consists of 1,000 images that have been meticulously annotated to identify the presence of individuals and facial features. Labeling is of utmost importance as it allows the model to differentiate between broad human figures, referred to as "person tags," and more specific facial characteristics, referred to as "face tags."

Following the completion of data preprocessing and labeling procedures, the dataset has undergone refinement, resulting in the inclusion of 985 images annotated with 'person tags' and 538 images annotated with 'face tags.' In order to enhance the efficacy of training and validation processes, the dataset is partitioned into two distinct components: The training dataset consists of 80% of the total images, specifically 787 images for person tags and 424 images for face tags. The remaining 20% of the images, which amounts to 198 images for person tags and 114 for face tags, are designated for validation purposes. The segmentation process guarantees a thorough learning experience and rigorous evaluation of the model.

Every individual image within the dataset is carefully chosen and subjected to a series of processing steps with the primary aim of accurately depicting the wide range of scenarios that are commonly encountered in closed-circuit television (CCTV) footage. These factors encompass fluctuations in illumination, perspective, and image sharpness, all essential for developing a model that can effectively operate in a wide range of surveillance environments.

3.1. Experiment Result Comparison

This study presents a comprehensive comparison of the performance metrics between the standard YOLOv8 and the modified YOLOv8 models after undergoing 200 epochs of training. This comparison primarily examines crucial metrics such as Recall, Precision, mean Average Precision (mAP), Inference Time, and Weight Size. These metrics are essential for evaluating the suitability of these models for real-time object detection in CCTV surveillance.

3.1.1 Overall Model Performance

Table 2. Result in Comparison Table between Models

Framework	Recall	Precision	mAP	Inference Time	Weight Size
Standard YOLOv8	86.9%	91.1%	92.1%	4.5 ms	21MB
Modified YOLOv8	82%	91.4%	90.5%	3.8 ms	4MB
Difference	-4.9%	+0.3%	-1.6%	0.7 ms	-17MB

Based on each class of standard and modified models, there are also some differences in parameter values as presented in Table 2. The standard YOLOv8 model exhibited a higher mAP and Recall compared to the modified model but was outperformed in terms of inference speed and exhibited a slightly lower Precision. The specific findings were as follows:

- **Recall**: The standard model's Recall was 4.9% higher than the modified model, indicating superior detection capabilities of relevant objects.
- **Precision**: The modified model showed a slight increase in Precision by 0.3%, suggesting a marginal improvement in accurately predicting identifications.
- **mAP**: There was a decrease of 1.6% in the mAP for the modified model, reflecting a slight drop in overall detection accuracy.
- **Inference Time**: A significant improvement was observed in the modified model, reducing inference time from 4.5 ms to 3.8 ms.
- **Weight Size**: The modified model demonstrated a substantial reduction in weight size, decreasing from 21MB to 4MB, indicating enhanced efficiency.

3.1.2 Class-Specific Performance Analysis

A deeper analysis of the performance of each class ('Face' and 'Person') revealed nuanced differences between the standard and modified models. The result is provided in the Table 3.

Table 3. Result in Comparison Table between Classes

Framework	Class	Recall	Precision	mAP
Standard YOLOv8	Face	79.6%	89.8%	87.7%
	Person	94.1%	92.5%	96.6%
Modified YOLOv8	Face	71.7%	89.8%	85.0%
	Person	92.4%	92.9%	96.0%
Difference	Face	-7.9%	0.0%	-2.7%
(Modified – Standard)	Person	-1.7%	+0.04%	+0.6%

- **Face Class**: In the modified model, the Recall for the face class decreased by 7.9% and the mAP by 2.7%, with no change in Precision. This indicates a reduced ability to detect all relevant faces, although the accuracy in correctly labeling detected faces remained consistent.
- **Person Class**: The modified model displayed a minor decrease in Recall of 1.7% for the person class but an improvement in mAP by 0.6%, with Precision remaining nearly constant. This suggests a balanced performance in detecting persons, maintaining a good level of accuracy.

3.1.3 Implications of the Findings

The results indicate that the architectural pruning in the modified YOLOv8 model leads to significant improvements in inference speed and a reduction in model size, which are critical for real-time applications like CCTV surveillance. However, these benefits come with a trade-off in accuracy, particularly in detecting faces.

The decrease in Recall for the 'Face' class is an important consideration, especially in contexts where missing a face detection could have profound implications. On the other hand, the improved efficiency metrics (inference time and model size) highlight the model's potential in environments where processing speed and storage are limited.

The findings underscore the importance of balancing efficiency and accuracy in optimizing models for real-time surveillance applications. Future work could explore methods to minimize the loss in accuracy while maintaining or further improving the efficiency gains.

3.2. Streamlit Integration

The trained model needs to be integrated with the Streamlit framework, so it is necessary to create the functions needed to perform inference on images and videos. In addition, Streamlit requires functions to add the required models. The first function of Streamlit is to store the trained model to be used in the application (Figure 8). If the model has been saved to streamlit, the user can perform inference on the existing images and videos by selecting the desired image and model (Figure 9). In addition to images, the model can be used to perform inference on videos. The YOLOv8 model will perform object detection for each frame in the video (Figure 10).

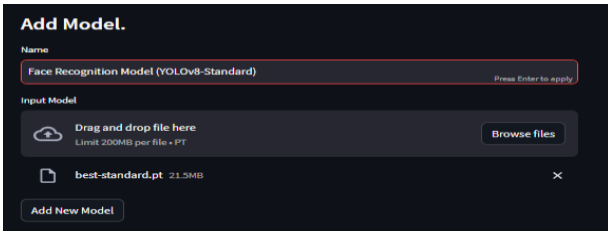

Figure 8. Adding Trained Model

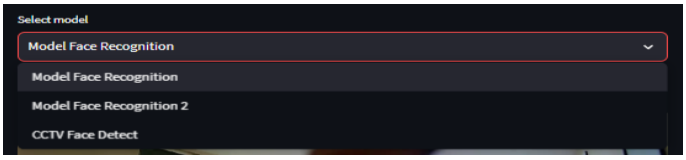

Figure 9. Selecting Model in Streamlit web app

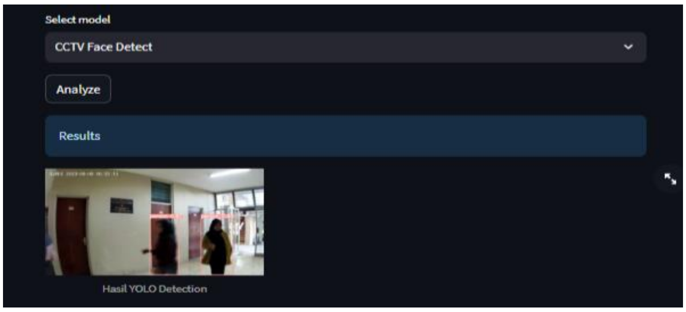

Figure 10. Performing Object Detection on a Picture

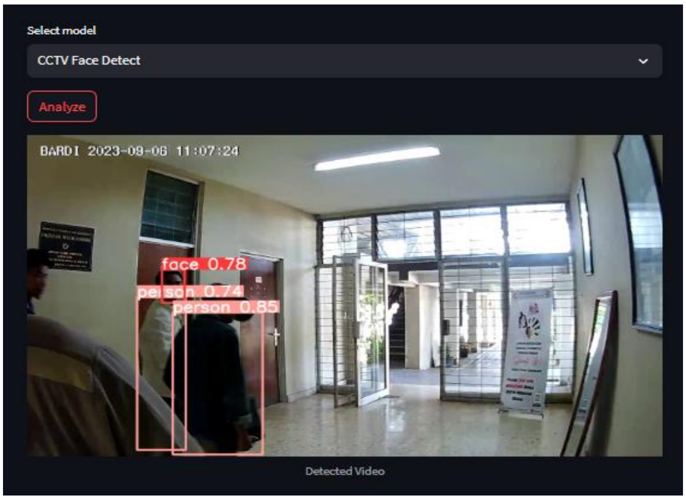

Figure 11. Performing Object Detection on a Video

4. CONCLUSION

The primary objective of this study was to address a significant obstacle encountered in real-time closed-circuit television (CCTV) surveillance, namely the requirement for a more expeditious and effective model for object detection. The study was centered on the optimization of the YOLOv8 model, with a specific emphasis on the pruning of its architecture. The aim was to improve the speed of inference while minimizing any notable impact on accuracy. The outcomes of this optimization can be observed through the comparative examination of the standard and modified YOLOv8 models, as depicted in Tables 2 and 3.

The modified YOLOv8 model demonstrated a significant improvement in inference time, decreasing from 4.5 milliseconds to 3.8 milliseconds. This enhancement is of utmost importance in real-time applications where rapidity is paramount. In addition to this enhancement, the model underwent a considerable decrease in its weight size, decreasing from 21MB to 4MB. This reduction has resulted in a significant decrease in the computational resources needed for its operation.

Nevertheless, the efficiency improvements were concomitant with a decline in accuracy metrics. In particular, the recall rate decreased 7.9% for the face category and 1.7% for the person category. Meanwhile, the mean Average Precision (mAP) experienced a marginal decrease of 2.7% for faces and a slight increase of 0.6% for persons. The precision of the face class remained consistent, while there was a slight improvement in precision for the person class.

The primary contribution of this study lies in its ability to showcase the effectiveness of strategically pruning the YOLOv8 architecture, resulting in a model that is both lighter and faster. This optimized model proves to be well-suited for real-time surveillance applications. This holds particular significance in light of the growing need for effective and resilient real-time object detection systems within security and surveillance domains. The considerable decrease in the size of the model, is a notable accomplishment, which improves the feasibility of implementing these models in systems that have limited computational capabilities.

The study effectively accomplished its objective of developing a more efficient and less cumbersome model, while also shedding light on the difficulties associated with balancing between velocity and precision. Hence, it is recommended that future investigations prioritize the refinement of the pruning methodology in order to mitigate the adverse effects on accuracy. This can be achieved by potentially exploring alternative pruning techniques or adaptive training methods, which have the potential to improve the precision of the model while preserving its efficiency.

In summary, the enhanced YOLOv8 model signifies a notable progression in advancing effective object detection models tailored for real-time applications. The enhanced speed of inference and decreased model size, along with sustained accuracy, render it a significant advancement in the domain of computer vision and a viable remedy for real-time closed-circuit television (CCTV) surveillance systems.

ACKNOWLEDGEMENTS

The researcher would like to express their gratitude to Politeknik Negeri Bandung (POLBAN) for their support in making this research possible through the Research Grant for Laboratory Capacity Enhancement with reference No. B/94.3/PL1.R7/PG.00.03/2023.

REFERENCES

[1] Majeed, F., Khan, F.Z., Iqbal, M.J., Nazir, M.: Real-Time Surveillance System based on Facial Recognition using YOLOv5. In: Proceedings of the 2021 Mohammad Ali Jinnah University International Conference on Computing, MAJICC 2021. Institute of Electrical and Electronics Engineers Inc. (2021)
[2] Kwaghe, O.P., Gital, A.Y., Madaki, A.G., Abdulrahman, M.L., Yakubu, I.Z., Shima, I.S.: A Deep Learning Approach for Detecting Face Mask Using an Improved Yolo-V2 With Squeezenet. In: 2022 IEEE 6th Conference on Information and Communication Technology, CICT 2022. Institute of Electrical and Electronics Engineers Inc. (2022)
[3] Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. (2022)
[4] Ultralytics: Ultralytics - YOLOv8, https://github.com/ultralytics/ultralytics
[5] Purwita Sary, I., Ucok Armin, E., Andromeda, S., Engineering, E., Singaperbangsa Karawang, U.: Performance Comparison of YOLOv5 and YOLOv8 Architectures in Human Detection Using Aerial Images. Ultima Computing : Jurnal Sistem Komputer. 15, (2023)
[6] Reis, D., Kupec, J., Hong, J., Daoudi, A.: Real-Time Flying Object Detection with YOLOv8. (2023)
[7] Li, Y., Fan, Q., Huang, H., Han, Z., Gu, Q.: A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition. Drones. 7, (2023). https://doi.org/10.3390/drones7050304
[8] Kuzlu, M., Catak, F.O., Sarp, S., Cali, U., Gueler, O.: A Streamlit-based Artificial Intelligence Trust Platform for Next-Generation Wireless Networks. In: Proceedings - 2022 IEEE Future Networks World Forum, FNWF 2022. pp. 94–97. Institute of Electrical and Electronics Engineers Inc. (2022)
[9] Khorasani, M., Abdou, M., Hernández Fernández, J.: Streamlit Use Cases. In: Web Application Development with Streamlit. pp. 309–361. Apress (2022)
[10] Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. (2020)
[11] Lin, J.: MultiStream: A Simple and Fast Multiple Cameras Visual Monitor and Directly Streaming. (2022)
[12] Li, M., Yeh, C.L., Lu, S.Y.: Real-time QoE monitoring system for video streaming services with adaptive media playout. International Journal of Digital Multimedia Broadcasting. 2018, (2018). https://doi.org/10.1155/2018/2619438
[13] Castellano, G., Castiello, C., Mencar, C., IEEE Systems, M., IEEE Computational Intelligence Society, Institute of Electrical and Electronics Engineers: 2020 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS) : conference proceedings : Bari, Italy, 27-29 May 2020 (online). (2020)
[14] Terven, J., Cordova-Esparza, D.: A Comprehensive Review of YOLO: From YOLOv1 and Beyond. (2023)
[15] Van Rijthoven, M., Swiderska-Chadaj, Z., Seeliger, K., Van Der Laak, J., Ciompi, F.: You Only Look on Lymphocytes Once.
[16] Powers, D.M.W., Ailab: EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION. (2011)
[17] Park, L., Park, L.A.F.: Bootstrap confidence intervals for Mean Average Precision Bootstrap confidence intervals for Mean Average Precision Bootstrap confidence intervals for Mean Average Precision Bootstrap confidence intervals for Mean Average Precision Bootstrap confidence intervals for Mean Average Precision. (2011)
[18] Su, W., Yuan, Y., Zhu, M.: A relationship between the average precision and the area under the ROC curve. In: ICTIR 2015 - Proceedings of the 2015 ACM SIGIR International Conference on the Theory of Information Retrieval. pp. 349–352. Association for Computing Machinery, Inc. (2015)
[19] Tan, L., Huangfu, T., Wu, L., Chen, W.: Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification. BMC Med Inform Decis Mak. 21, (2021). https://doi.org/10.1186/s12911-021-01691-8
[20] Nápoles-Duarte, J.M., Biswas, A., Parker, M.I., Palomares-Baez, J.P., Chávez-Rojo, M.A., Rodríguez-Valdez, L.M.: Stmol: A component for building interactive molecular visualizations within streamlit web-applications. Front Mol Biosci. 9, (2022). https://doi.org/10.3389/fmolb.2022.990846
[21] Khorasani, M., Abdou, M., Fernández, J.H.: Web Application Development with Streamlit: Develop and Deploy Secure and Scalable Web Applications to the Cloud Using a Pure Python Framework. Apress Media LLC (2022)