



Advanced Face Detection with YOLOv8: Implementation and Integration into AI Modules

Handiso Misgana Yisihak*, Li Li

Tianjin University of Technology and Education, Tianjin, China

Email: msganayisihak@gmail.com

How to cite this paper: Yisihak, H.M. and Li, L. (2024) Advanced Face Detection with YOLOv8: Implementation and Integration into AI Modules. *Open Access Library Journal*, 11: e12474.

<https://doi.org/10.4236/oalib.1112474>

Received: October 14, 2024

Accepted: November 24, 2024

Published: November 27, 2024

Copyright © 2024 by author(s) and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper presents a comprehensive approach to face detection utilizing the YOLOv8 model, specifically trained on a diverse dataset consisting of images from four individuals. The trained model is seamlessly integrated into an AI module from Huada, a leading AI company, equipped with a camera and LED indicators, enabling real-time face recognition and classification of known and unknown individuals. The model's performance is evaluated across various metrics, demonstrating its high accuracy, robustness, and efficiency in real-world scenarios. Additionally, the deployment process is detailed, showcasing the practical challenges and solutions encountered during the integration into a security application. Our results indicate that YOLOv8 is not only effective in identifying individuals with high precision but also scalable and adaptable to different environments. This work contributes to the development and deployment of advanced face detection systems, with significant implications for security and surveillance applications.

Subject Areas

Applications of Communication Systems

Keywords

YOLOv8, AI Module, Evaluation Matrices

1. Introduction

Face detection is a critical component in a wide range of applications, from security systems to personal devices, where the need for accurate and efficient

*First author.

detection methods is paramount. Traditional face detection methods, such as Haar Cascades and Histogram of Oriented Gradients (HOG), have been widely used; however, these models often struggle with variations in lighting, occlusions, and processing speed, making them less suitable for real-time applications but recent advancements have seen the emergence of the YOLO (You Only Look Once) series, known for their exceptional speed and accuracy. Earlier YOLO versions, like YOLOv3 and YOLOv4, improved object detection performance but were not specifically optimized for face detection tasks in dynamic real-world scenarios. These models also encountered issues with detecting smaller faces or handling occlusions effectively. For example, recent research has shown improved object detection in other domains like colorectal polyp detection, using YOLO combined with super-resolution techniques to enhance detection accuracy in challenging conditions. To address these shortcomings, this research focuses on YOLOv8, the latest iteration in the YOLO family, which offers state-of-the-art object detection capabilities [1]. YOLOv8 introduces several architectural improvements, such as a more efficient backbone and a refined head for better localization and classification accuracy. In this paper, we focus on the latest iteration, YOLOv8, which represents a state-of-the-art object detection algorithm designed to enhance performance, particularly in real-time application.

The primary objective of this research is to train a YOLOv8 model on a custom dataset consisting of images of four individuals and to integrate the trained model into an AI module for real-time face recognition. The integration of YOLOv8 into AI modules offers significant improvements in face detection accuracy, even in challenging conditions such as varying lighting or occlusions [2]. The AI module in this study is equipped with a camera and LED indicators that provide visual feedback based on detection results, making it suitable for applications like gate security systems, where rapid and reliable face recognition is essential [3]. This paper contributes to the field by demonstrating the successful training of YOLOv8 for face detection, the seamless integration of the model into a practical AI system, and its effectiveness in real-world scenarios. By leveraging the advancements of YOLOv8, this project pushes the boundaries of AI-driven face detection, offering a robust solution for various applications.

2. Related Work

The field of facial recognition has seen rapid advances with the development of deep learning models, especially with the YOLO (You Only Look Once) family of algorithms.

First introduced by Redmon et al., YOLO revolutionized object detection by presenting it as a regression problem, allowing real-time object detection with impressive speed and accuracy [4]. YOLO's efficiency and performance have made it a popular choice in a variety of fields, including facial recognition, where real-time performance is essential.

Recent versions of YOLO, such as YOLOv8, continue to build on the

foundations of previous versions by incorporating more complex neural network architecture and optimization techniques. These advances have further improved YOLO's ability to detect smaller objects, improving accuracy while maintaining the speed needed for real-time applications [3].

Integrating such models into AI modules for tasks such as face recognition is a natural progression, given the growing need for intelligent systems capable of performing complex tasks in dynamic environments.

In the context of face recognition systems, several studies have explored the use of YOLO for applications such as security systems and employee attendance tracking. For example, Al Farizi et al. demonstrated the effectiveness of YOLO combined with geometric analysis for automatic feature detection, which is critical for robust face recognition under various environmental conditions [5]. Their work highlights the potential of YOLO-based systems in solving real-world challenges, making them suitable for integration into AI modules.

Moreover, research by Daffa Arifadilah *et al.* on script detection using YOLO underscores the algorithm's versatility in detecting specific patterns and structures within images [6]. Although focused on script detection, their findings contribute to the broader understanding of how YOLO can be adapted for specialized detection tasks, including facial feature detection, by leveraging its inherent strengths in pattern recognition.

These studies provide a solid foundation for the present work, which focuses on the implementation and integration of YOLOv8 into AI modules for advanced face detection. By leveraging the improvements in YOLOv8, this project aims to achieve high accuracy and real-time performance in face detection, with practical applications in security and automated systems.

3. YOLOv8 Model Architecture

YOLOv8 incorporates several advancements over its predecessors. The backbone and the head are the two main parts of the convolutional neural network used by YOLOv8, marking an improvement over earlier YOLO algorithms. The backbone of YOLOv8 is built upon the CSPDarknet53 architecture, which consists of 53 convolutional layers with cross-stage partial connections to enhance information flow between layers. This allows for better feature extraction and information propagation through the network, which significantly improves the model's accuracy and efficiency. Several convolutional layers followed by fully connected layers comprise the head of YOLOv8 [7].

YOLOv8 also introduces several new features that distinguish it from previous versions:

Refined Backbone Engineering: YOLOv8 presents CSPDarknet53, a new backbone architecture, which works on the progression of data among layers and keeps a harmony among accuracy and computational proficiency.

Improved Feature Pyramid Network (FPN): The FPN in YOLOv8 is more refined, upgrading the model's capacity to detect objects at various scales, particularly

small and medium-sized objects.

Anchor-Free Detection Head: YOLOv8 incorporates an anchor-free detection head, eliminating the need for predefined anchor boxes. This allows the model to adapt more dynamically to various object shapes and sizes, leading to improved accuracy and a simpler training process.

New Loss Function: A new loss function in YOLOv8 is optimized for better convergence during training, further developing execution across detection, segmentation, and classification tasks.

Versatile Hardware Support: YOLOv8 is designed to run efficiently across various hardware platforms, from CPUs to GPUs, making it more versatile and accessible for different applications, including edge computing and real-time processing.

Enhanced Export and Deployment Flexibility: YOLOv8 supports multiple export formats, which facilitates easier deployment across various platforms and systems, crucial for real-time applications like security systems.

Overall, YOLOv8 is designed to balance speed and accuracy, making it ideal for real-time applications like face detection in security systems.

As illustrated in **Figure 1**, the YOLOv8 Model Architecture Diagram provides a comprehensive overview of the model's structure and components.

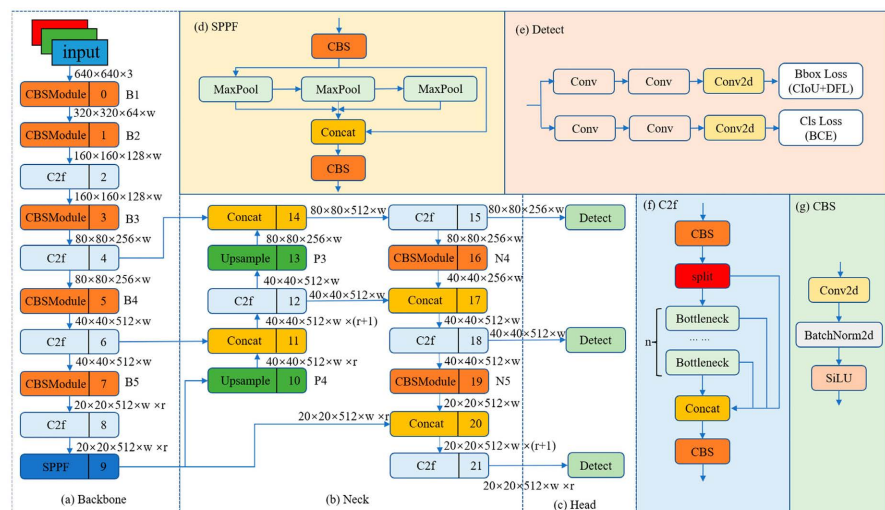


Figure 1. YOLOv8 model architecture diagram.

Design Process

The design process for training YOLOv8 in this project involved several steps.

Dataset Preparation: A custom dataset of four individuals was collected, with images labeled using bounding boxes around the faces. The dataset was split into training and validation sets, ensuring sufficient variety in lighting, pose, and occlusion to make the model robust for real-world scenarios.

Model Training: The YOLOv8 model was trained using transfer learning, initializing the weights from a pre-trained model on the COCO dataset to accelerate

the learning process. The model was fine-tuned on the custom dataset using an appropriate learning rate, batch size, and number of epochs. Data augmentation techniques, such as random flipping and scaling, were applied to enhance the generalization capability of the model.

Performance Optimization: Hyperparameter tuning was performed to achieve optimal performance in terms of both detection accuracy and inference speed. The batch size, learning rate, and anchor box sizes were adjusted to improve the model's ability to detect faces under challenging conditions, such as varying illumination or partial occlusions. The model was trained over multiple iterations to minimize the loss function, focusing on reducing classification and localization errors.

Integration into AI Module: After the training phase, the YOLOv8 model was integrated into an AI module equipped with a camera and LED indicators for real-time face recognition. The module was programmed to provide visual feedback based on detection results, such as lighting up specific LEDs when a known face is detected. The model's real-time performance was a key consideration, and optimizations were made to ensure the system could operate with low latency in real-world conditions.

4. Methodology

4.1. Data Collection

The dataset utilized in this study comprises images from four people. The pictures were captured in different lighting conditions and from different angles to ensure robustness. The dataset was annotated manually—using labeling—with bounding boxes marking the faces.

As shown in **Figure 2**, the collected data samples demonstrate the variety and characteristics of the dataset used for training.

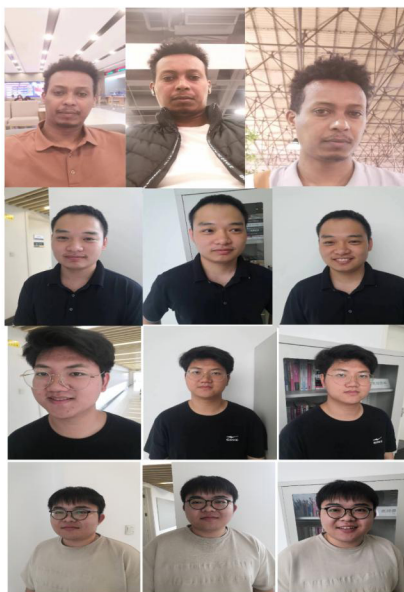


Figure 2. Collected data samples.

Figure 3 illustrates the process of collecting and annotating images for YOLOv8, including the flowchart of the data collection process, directory structure, and examples of annotated images.

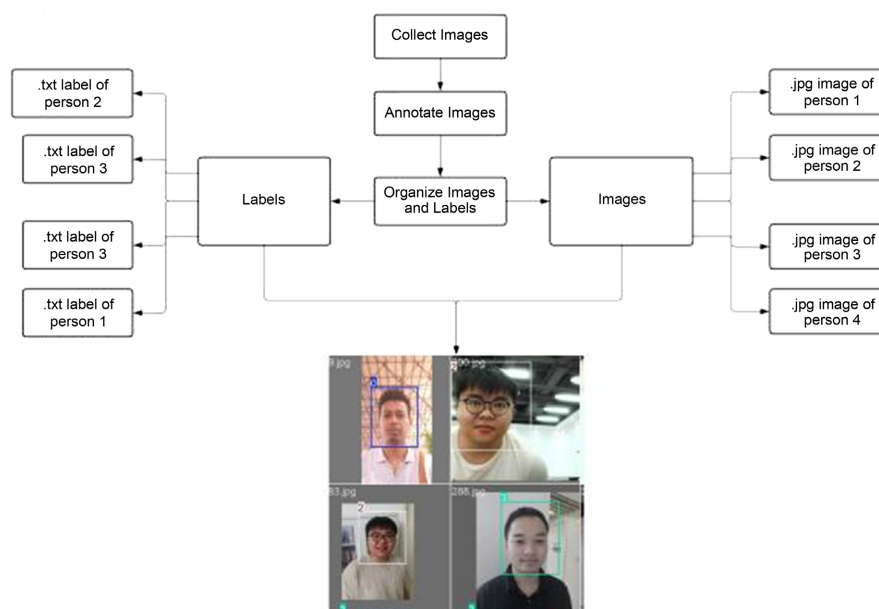


Figure 3. YOLOv8 data collection and annotation process.

4.2. Model Training

The training of the YOLOv8 model was carried out over 100 epochs, meaning the entire dataset was passed through the model 100 times. This extensive training was necessary to enable the model to thoroughly learn and generalize the patterns associated with faces across diverse images. Dataset Preparation Annotated face images that had been pre-processed and labeled using the YOLO format made up the training dataset. In addition to applying data augmentation techniques like random flipping, scaling, and color modification, the images were resized to match the model's input dimensions. The model's capacity to generalize across many situations and facial features was enhanced by these additions.

Loss Function

The model's predictions and the ground truth annotations were compared during the training phase. A combination of loss functions was used to calculate the difference between the actual labels and the predicted bounding boxes. Typically, these consist of:

- Bounding Box Regression Loss: Measures the error in predicting the bounding box's position and dimensions.
- Objectness Loss: Measures how well the model identifies if face exists in a given grid cell.
- Classification Loss: Measures how accurately the model classifies the detected faces into the correct category.

The total loss was minimized using an optimizer, which adjusted the model's

parameters based on the calculated gradients. Optimization and Learning Rate The Adam optimizer, which works well with complicated models and large data sets, was used to optimize the model's parameters. In order to enable the model to learn quickly in the early phases of training, the learning rate—a crucial hyperparameter—was first set to a larger value. Later on, it was gradually lowered to improve the model's performance.

Monitoring and Evaluation

To evaluate the model's performance, important parameters like precision, recall, and mean Average Precision (mAP) were tracked during the training phase. The model's ability to recognize and locate faces was shown by these measures. Furthermore, to look for indications of overfitting or underfitting, the training loss and validation loss were monitored. If there was a considerable decline in the validation loss, training was thought to end early. At the end of 100 epochs, the model was able to generalize well on the training data and showed promising results in terms of accurately detecting faces in unseen images (See [Figure 4](#)).

```

Epoch   CPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
100/100  6G       0.7441   0.3201   1.134     15         640: 100%|██████████| 10/10 [09:20<00:00, 31.13s/it]
          Class  Images  Instances  Box(P)   R         mAP50  mAP50-95: 100%|██████████| 9/9 [02:06<00:00,
          all   287    284      0.983   0.997    0.99   0.807

100 epochs completed in 17.500 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 6.2MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 6.2MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.2.69 Python-3.12.4 torch-2.4.0 CPU (ARMv8 Processor rev 0 (v8l))
Model summary (fused): 168 layers, 3,006,428 parameters, 0 gradients, 8.1 GFLOPs
Class      Images  Instances  Box(P)   R         mAP50  mAP50-95: 100%|██████████| 9/9 [01:53<00:00,
  all       287    284      0.983   0.997    0.99   0.807
  person1   74     74      0.982    1     0.995  0.895
  person2   73     73      0.959   0.986   0.973  0.765
  person3   75     75      0.995    1     0.995  0.639
  person4   62     62      0.995    1     0.995  0.727

Speed: 1.1ms preprocess, 336.2ms inference, 0.0ms loss, 1.5ms postprocess per image
Results saved to runs/detect/train2

```

Figure 4. Final training and validation metrics after 100 epochs.

The results shown in [Figure 5](#) were obtained by the process of training a YOLOv8 model for face detection of four individuals. This training involved a custom dataset consisting of images of these individuals, where the primary task was to accurately detect their faces under various conditions. The training framework automatically logged and visualized these metrics to provide insight into the model's performance over the 100 epochs. These graphs help us understand how well the model is learning and if it's improving in terms of accuracy, loss reduction, and detection quality.

Each image in the batch is processed simultaneously, with the model identifying and bounding faces, as shown by the detection boxes (See [Figure 6](#)). This visual demonstrates the model's capability to handle multiple images and detect faces in various conditions.

4.3. AI Module Integration

The trained YOLOv8 model was integrated into an AI module equipped with a camera and an LED indicator, as illustrated in [Figure 7](#), which shows the schematic of

the AI module. The camera captures real-time video feeds, which are processed by the YOLOv8 model. The LED indicator provides visual feedback: a green light for recognized (known) faces and a red light for unrecognized (unknown) faces.

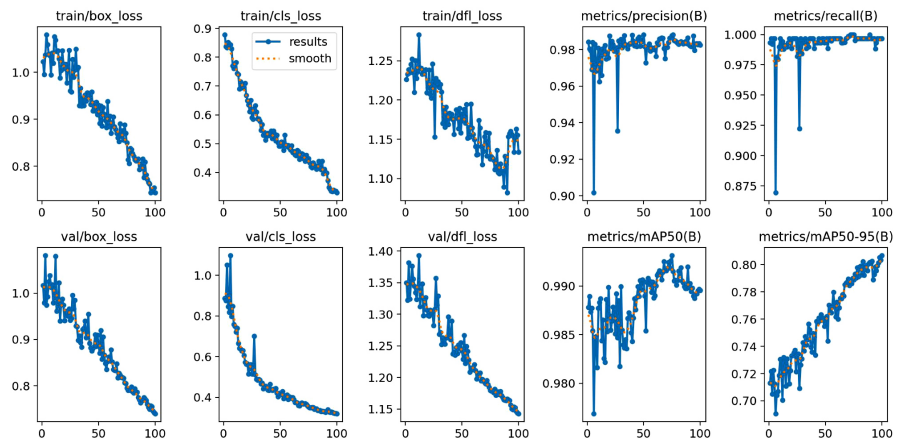


Figure 5. Training loss and accuracy curves over 100 epochs.

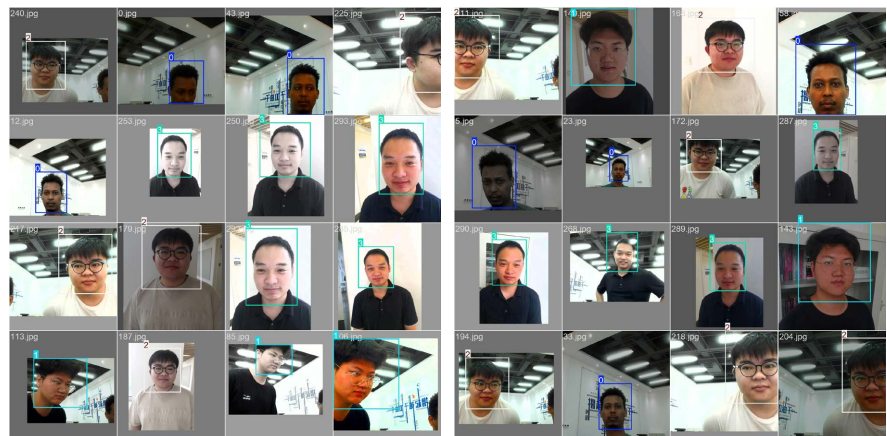


Figure 6. Example of a batch of training images.



Figure 7. Schematic of the AI module with camera and LED indicator.

4.4. Evaluation Metrics

The performance of the model was evaluated using precision, recall, and F1-score.

Additionally, a confusion matrix was generated to provide insight into the model's classification accuracy.

The specific formulas for these metrics [8] are:

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

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}'}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}'}$$

$$\text{mAP} = \frac{AP}{N} = \frac{\int_0^1 p(r) dr}{N'}$$

where TP (true positive) represents the number of correct predictions, FP (false positive) denotes the number of incorrect positive predictions, FN (false negative) represents the number of positive instances that the model failed to predict correctly. Precision represents the proportion of correct positive predictions, and recall represents the proportion of all correct predictions. The F1-score ranges from 0% to 100%. The F1-score represents the weighted average of precision and recall. The F1-score of 100% represents the best possible classification performance. The higher the F1-score, the better the model performance [9].

5. Results

5.1. Model Performance

The trained YOLOv8 model achieved high accuracy in detecting the faces of the four individuals in the dataset. Precision, recall, and F1-score metrics are reported in **Table 1**.

Table 1. Precision, recall, and F1-score of the YOLOv8 model.

Class	Image	Instances	Precision	Recall	mAP50	mAP50-95
All	287	284	0.98	0.997	0.99	0.807
Person 1	74	74	0.98	1	0.995	0.895
Person 2	73	73	0.95	0.986	0.973	0.765
Person 3	75	75	0.99	1	0.995	0.839
Person 4	65	62	0.99	1	0.995	0.727

5.1.1. Recall-Confidence Curve

The Recall-Confidence Curve in **Figure 8** illustrates the model's performance across different confidence thresholds for each individual (Person 1, Person 2, Person 3, Person 4):

-Higher Recall: Person 1, Person 2, and Person 4 maintain higher recall at increasing confidence levels, meaning the model detects them well even when requiring more certainty.

-Overall Performance: The blue line (all classes) shows that the model has high recall at lower confidence levels but experiences a sharp decline as confidence increases.

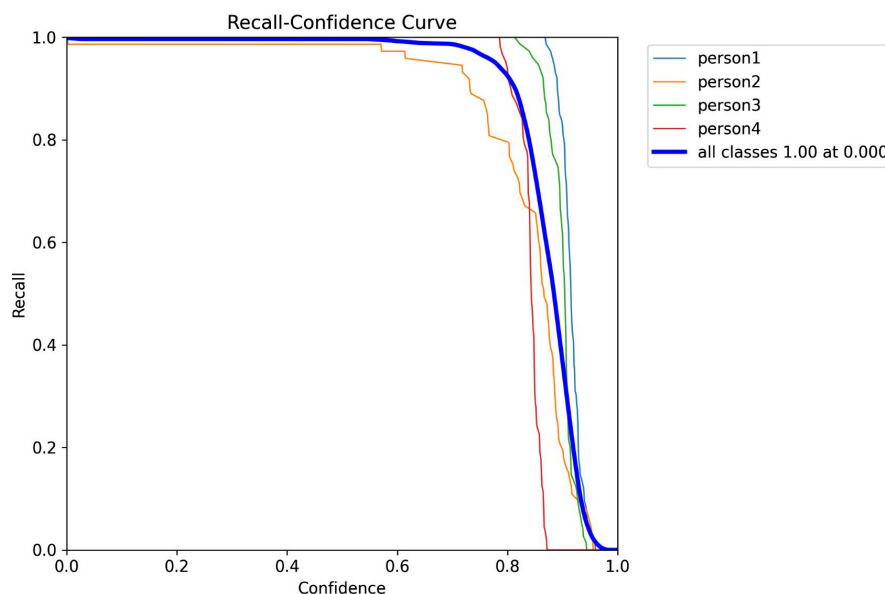


Figure 8. Recall-confidence curve.

5.1.2. Precision-Recall Curve

The Precision-Recall Curve in **Figure 9** shows the model's performance in terms of precision and recall for each individual (Person 1, Person 2, Person 3, Person 4) as well as the overall performance across all classes.

-Precision: All individuals (Person 1, Person 2, Person 3, Person 4) have very high precision close to 1.0, meaning the model is highly accurate in its predictions.

-Recall: The recall values are also very high, staying close to 1.0 for all individuals, indicating that the model is capable of detecting most true positives (faces).

-mAP@0.5: The mean Average Precision (mAP) at IoU 0.5 for all classes is 0.990, reflecting excellent overall performance.

The model shows strong performance with nearly perfect precision and recall across all individuals, suggesting that it can reliably detect faces with minimal false positives and missed detection.

5.1.3. Precision-Confidence Curve

The Precision-Confidence Curve in **Figure 10** shows how precision varies with confidence for each individual (Person 1, Person 2, Person 3, Person 4) and across all classes.

-High Precision: All individuals maintain very high precision, close to 1.0, across almost all confidence levels. This means that the model is highly accurate in making predictions with very few false positives, even as the confidence threshold increases.

-All Classes: The blue line representing all classes shows that the model achieves

perfect precision (1.0) for most confidence thresholds.

-Stability: The curves are stable across confidence levels, indicating that the model maintains a high level of precision consistently.

The model demonstrates excellent precision across all confidence levels for each individual, making it highly reliable in correctly identifying faces with minimal errors.

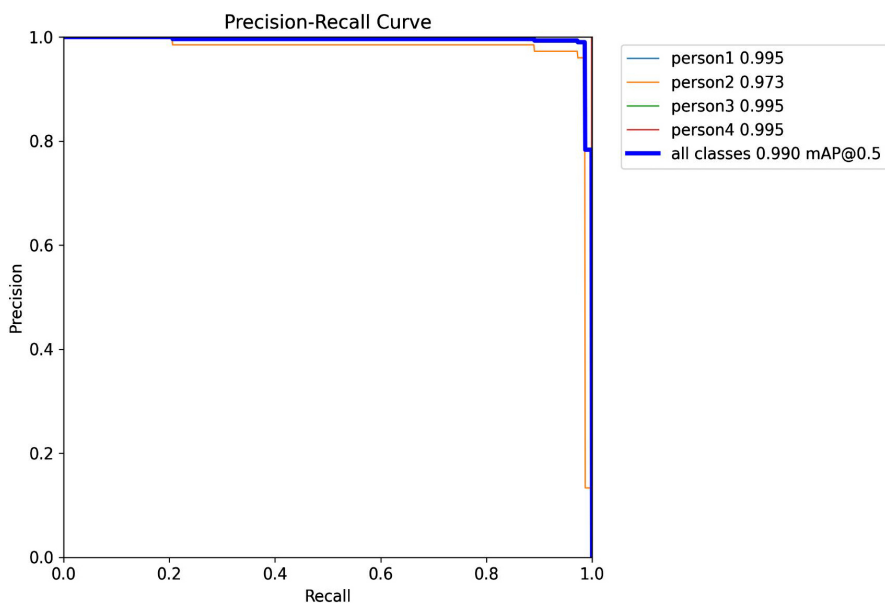


Figure 9. Precision-recall curve.

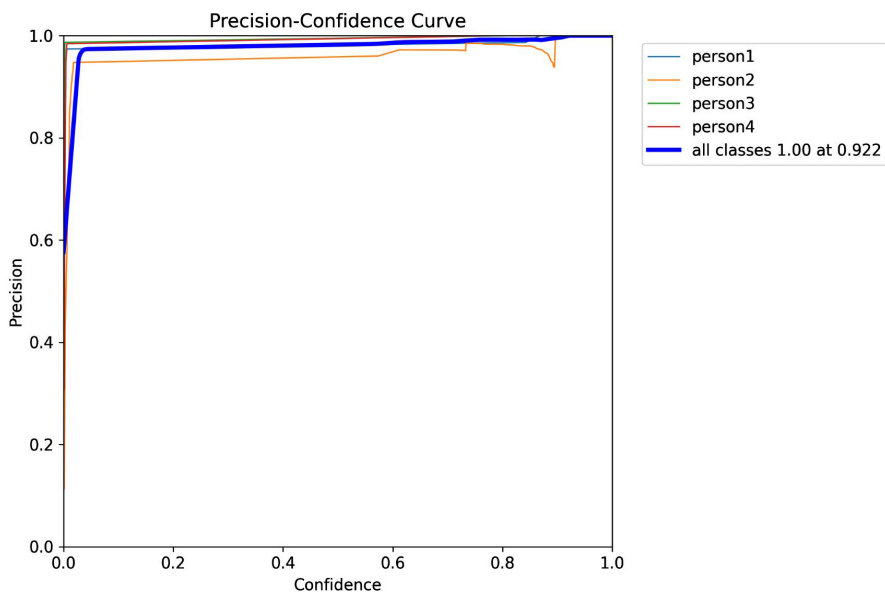


Figure 10. Precision-confidence curve.

5.1.4. Confidence Curve

The F1-Confidence Curve illustrates how the F1 score, a balance between precision and recall, changes with increasing confidence thresholds for each person

(Person 1, Person 2, Person 3, Person 4) and across all classes as it is shown in **Figure 11**.

-Person 1, Person 2, Person 4: These individuals maintain high F1 scores close to 1.0 across a broad range of confidence levels, indicating that the model achieves a good balance of precision and recall for them.

-All Classes (blue line): The overall F1 score is 0.99 at a confidence threshold of 0.546, meaning the model performs well for all classes up to this point. The score drops sharply as confidence increases further.

The model achieves an excellent balance of precision and recall (high F1 scores) for most confidence thresholds, making it robust in detecting faces while minimizing false positives and missed detections.

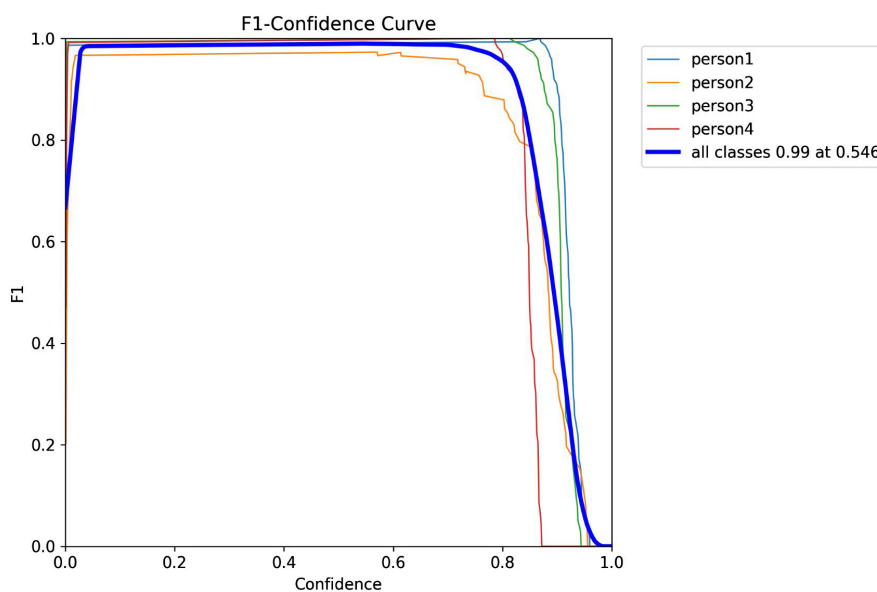


Figure 11. F1-confidence curve.

5.1.5. Labels

Figure 12 presents the labels used in the dataset, providing insight into the classification criteria applied during the training process.

- Top-left (Bar Graph): This graph compares the number of instances detected for each person (Person 1, Person 2, Person 3, Person 4). The height of the bars indicates the frequency of detections across different classes, with similar counts for all persons, suggesting a balanced data distribution.

- Top-right (Bounding Box Visualization): This overlay of bounding boxes shows the localization of detected faces. The overlap suggests that the model predicted consistent bounding box coordinates, though there might be some variance in object sizes and positioning.

- Bottom-left (Scatter Plot—X-axis vs. Y-axis): This scatter plot seems to show the distribution of detected objects in the image space (x vs. y coordinates). The spread indicates that the model detected faces across various parts of the image.

- Bottom-right (Scatter Plot—Width vs. Height): This scatter plot shows the

width and height of the bounding boxes. The variance suggests that faces with different sizes are detected, possibly due to variations in distance or image scaling.

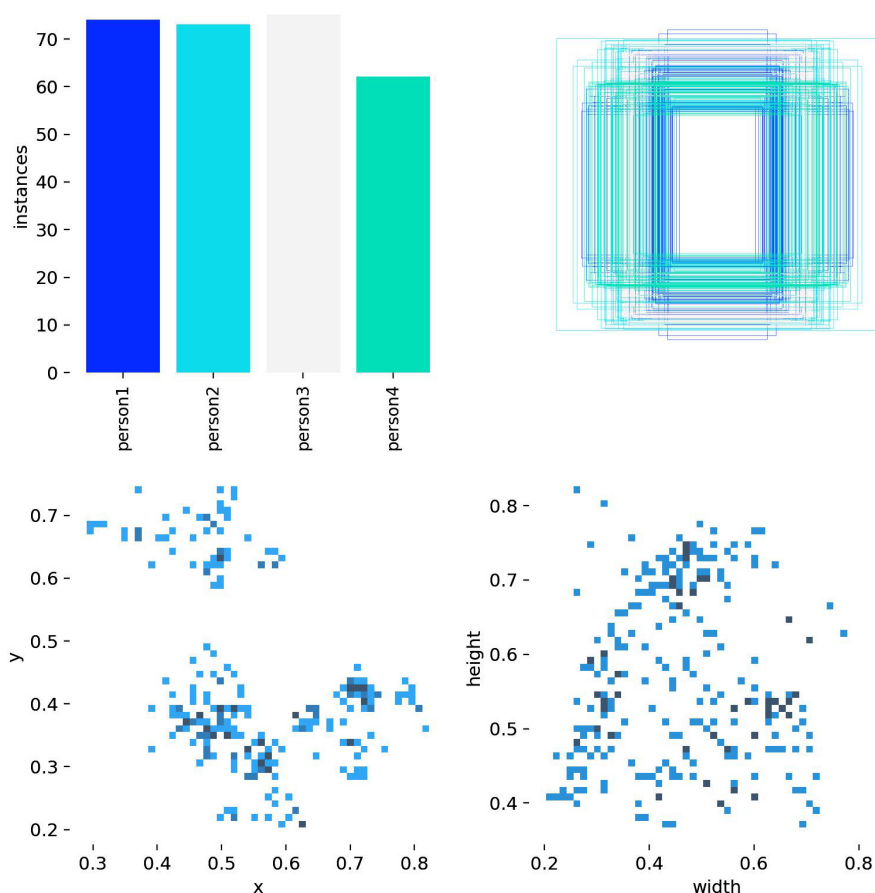


Figure 12. Labels.

5.1.6. Labels-Correlogram

Figure 13 illustrates the labels-correlogram, highlighting the relationships and correlations among the different labels in the dataset.

Diagonal Histograms:

The histograms along the diagonal represent the distribution of each individual variable.

x and y coordinates show that the objects are spread evenly across the image space.

The width and height distributions indicate that the model has detected objects of various sizes, with some skewness towards certain object dimensions.

Scatter Plots (Off-diagonal):

x vs. y: Shows how objects are distributed across the image.

width vs. height: Displays the relationship between object size parameters. There's a diverse range of sizes, but the objects mostly cluster around specific width-to-height ratios.

Other pairs (e.g., x vs. width, y vs. height): Illustrate correlations between the object location and their size. This can help understand if objects tend to be larger in specific areas of the image.

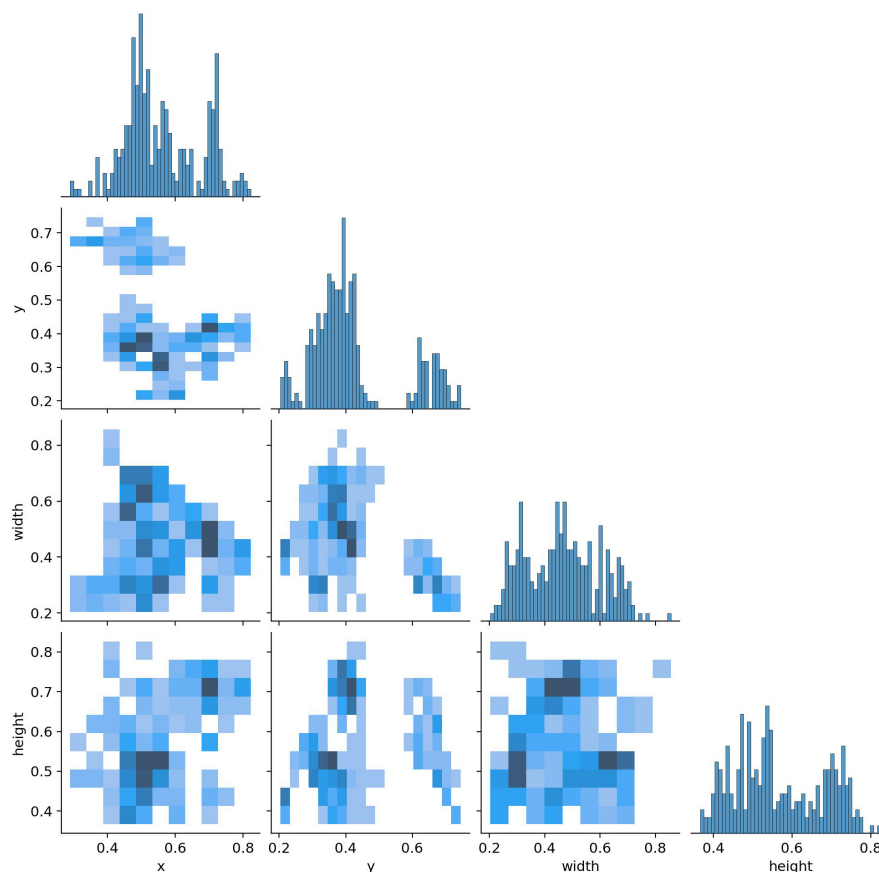


Figure 13. Labels-correlogram.

5.2. Confusion Matrix

Figure 14 displays the confusion matrix of the YOLOv8 model's predictions, providing a detailed view of the model's classification performance across different categories.

The confusion matrix provides a detailed view of the model's performance across different classes. The results indicate that the model correctly identifies known faces with a high degree of accuracy.

The model performs well, with most predictions correctly classified.

The model's confusion matrix demonstrates effective classification of individuals, with very few errors across all classes.

5.3. Real-World Testing

The AI module was tested in a real-world environment, where it successfully identified known individuals and provided the correct visual feedback via the LED indicator (See **Figure 15**).

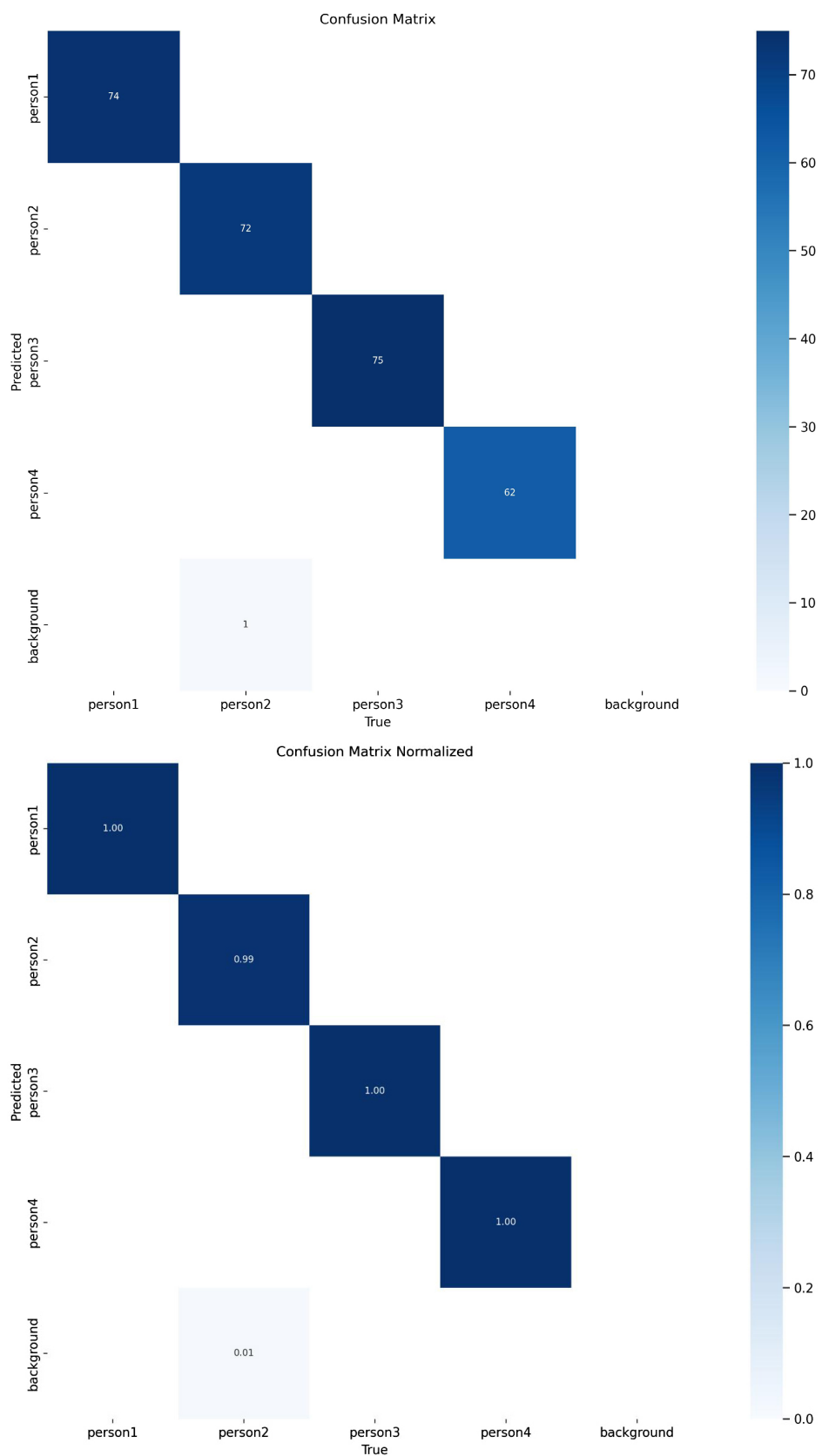


Figure 14. Confusion matrix of the YOLOv8 model's predictions.



Figure 15. Real-world deployment of the AI module, showing the green LED for a recognized face.

6. Discussion

The results clearly showcase the strong performance of the YOLOv8 model in the task of face detection, especially when trained on a dataset tailored specifically for this purpose. The model achieved high precision and recall scores, underscoring its suitability for applications in security, where accurate and reliable face recognition is paramount. These metrics indicate that the model can effectively distinguish between faces with minimal false positives and negatives, making it a dependable choice for scenarios that require heightened accuracy.

However, it is important to note that the model's effectiveness might be somewhat limited by certain factors not fully addressed in the training phase. Specifically, variations in lighting conditions and facial poses—elements that were underrepresented in the training dataset—could potentially impact the model's detection accuracy in real-world scenarios. This limitation suggests a need for further research and development.

To enhance the model's robustness and adaptability, future work could focus on incorporating a more diverse and comprehensive dataset that includes a wider range of lighting conditions, facial expressions, and angles. Additionally, employing advanced data augmentation techniques could simulate these variations during training, thereby helping the model to generalize better and maintain high performance across different environments.

7. Conclusions

In this paper, we have successfully trained and implemented a YOLOv8 model for face detection, integrating it into an AI module capable of real-time face recognition. Our results underscore the efficacy of YOLOv8 as a robust and accurate tool for face detection in controlled environments. The seamless integration into a practical system highlights the model's feasibility and effectiveness in real-world applications, particularly in security and surveillance contexts. We would like to acknowledge **Huada, a leading AI company, Tianjin** for providing the AI

module, which played a critical role in enabling the real-time face recognition capabilities of our system.

Looking ahead, future research could focus on extending the capabilities of YOLOv8 by applying it to more diverse and complex environments, where varying lighting conditions, occlusions, and other challenges are present. Additionally, the integration of supplementary sensors or data sources, such as thermal cameras or biometric inputs, could further enhance the system's accuracy, robustness, and overall performance.

Acknowledgements

I would like to extend my heartfelt thanks to my supervisor, **professor Li Li**, for her unwavering support and guidance throughout my research. I also wish to express my sincere gratitude to **Mr. Chai**, the manager at Huada Company, for his invaluable support and for providing the AI module used in this study. Additionally, I am deeply grateful to my big brother, **Dr. Selamu Yisihak**, for his encouragement and inspiration, which have been instrumental in motivating me to pursue research and continually improve myself. I would also like to express my deepest appreciation to my beloved wife, **Bilisuma Dereje**, for her love, patience, and unwavering support throughout this journey. Their collective contributions were crucial to the successful completion of this work.

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] Lakshmi Devi, S. and Sridhar, D. (2024) An Analogy Analysis of the Object Detection Algorithms Using YOLOv5, YOLOv7, and YOLOv8. Sri Krishna Adithya College of Arts and Science.
- [2] Wang, S.F., Xie, J., Cui, Y.R. and Chen, Z.J. (2022) Colorectal Polyp Detection Model by Using Super-Resolution Reconstruction and YOLO. School of Computer Science, Yangtze University.
- [3] Lin, B.Y. and Hou, M. (2024) Face Mask Detection Based on Improved YOLOv8. *Journal of Electrical Systems*, **20**, 365-375.
- [4] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 27-30 June 2016, 779-788. <https://doi.org/10.1109/cvpr.2016.91>
- [5] Al Farizi, M., et al. (2023) Using YOLO with Geometric Analysis for Robust Face Recognition. *Journal of Physics: Conference Series*, **2199**, Article No. 012010. <https://doi.org/10.1088/1742-6596/2199/1/012010>
- [6] Arifadilah, D., Asriyanik and Pambudi, A. (2024) Sunda Script Detection Using You Only Look Once Algorithm. *Journal of Artificial Intelligence and Engineering Applications (JAIEA)*, **3**, 606-613. <https://doi.org/10.59934/jaiea.v3i2.443>
- [7] Sridhar, D. and Karad, V. (2024) An Analogy Analysis of the Object Detection Algorithms Using YOLOv5, YOLOv7, and YOLOv8.

- [8] Bishop, C.M. (2006) Pattern Recognition and Machine Learning. Springer.
- [9] Fiveable Library. (n.d.). F1 Score in Machine Learning. Fiveable AI Resources.
<https://library.fiveable.me/key-terms/natural-language-processing/f1-score>