Laporan Project Akhir

Sistem Informasi Cerdas

Disusun oleh:

Minhajul Yusri Khairi	H071221006
Nurul Alya	H071221009
Kelvin Leonardo Sianipar	H071221020
Trisman Tegar Wiratama	H071221023
Joy Abrian Rantepasang	H071221030
Izzata Clarissa Salsabila	H071221065
Mifthahul Hoiri Bachrudin Basir	H071221072

Universitas Hasanuddin
Makassar, Indonesia

November 15, 2024

Contents

Chapter 1

Project Charter

1.1 Background

1.2 Objectives

1.3 Scope

Chapter 2

Methods

2.1 Data Collection

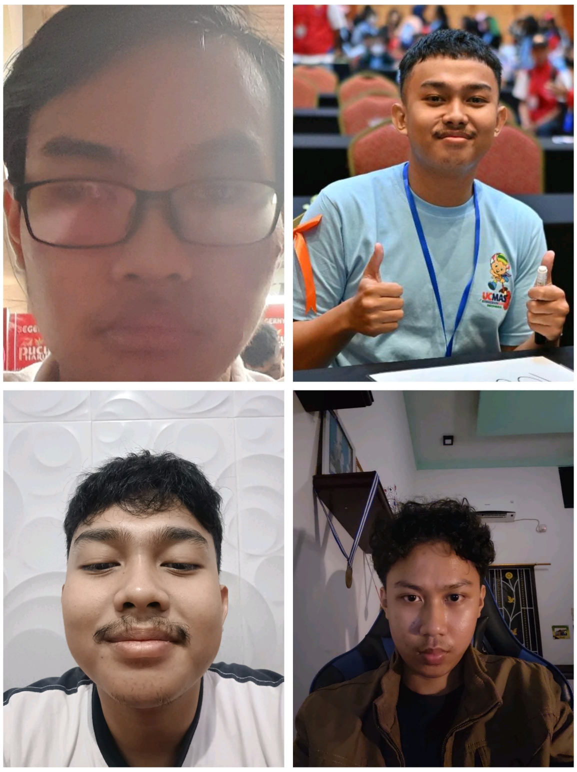


Figure 2.1: Image captured example

The data for the face recognition-based attendance application was collected from multiple sources to ensure the model's ability to recognize student faces in real-world classroom conditions. We used two primary methods for data collection:

2.1.1 Manual Image Capture

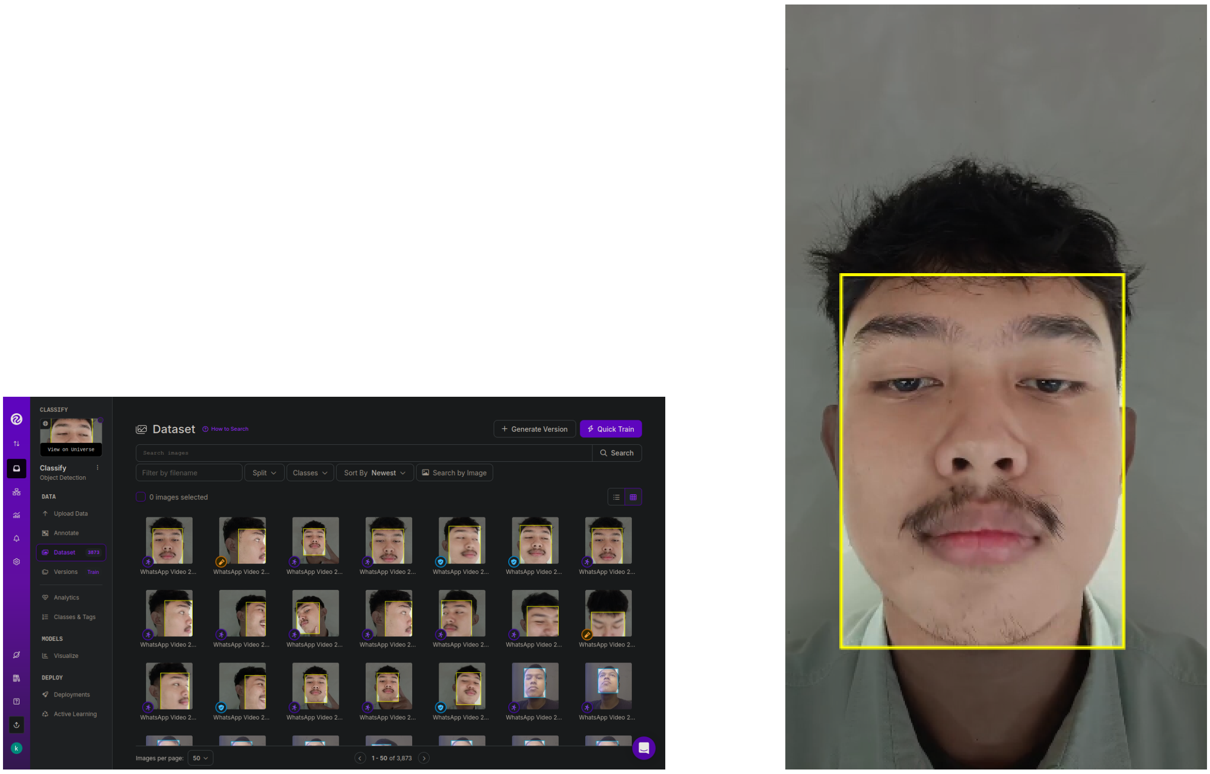
Several images of our team member faces were manually captured using a camera in various lighting conditions and backgrounds. These images were collected to ensure the dataset includes faces in both ideal and suboptimal conditions(see Figure 2.1).

2.1.2 Video Frame Extraction

In addition to manually captured images, we also utilized a video frame extraction technique. Videos were recorded in classroom environments, and frames were extracted from the videos to increase the dataset's size. This method also allows for the inclusion of multiple facial images from different angles, poses, and expressions.

2.1.3 Image Labeling with Roboflow

After gathering the images, we used Roboflow, a popular image annotation tool, to label the dataset. Roboflow enabled us to efficiently label each image with the corresponding student's identity, creating a structured dataset for training the face recognition model. With Roboflow, we were able to organize the images into different classes, each representing our team member (see Figure 2.2).



(a) Roboflow interface for list of annotated and unannotated images

(b) Roboflow interface for image labeling process

Figure 2.2: Roboflow interface: (a) Annotation list view, (b) Labeling process view

2.2 Data Explanation

The dataset used for the attendance system consists of facial images of students, each assigned a label corresponding to a unique student identity. The key features of the dataset are outlined as follows:

2.2.1 Images

Each image is stored in a common format such as JPEG or PNG, with varying resolutions ranging from 640x640 pixels to 720x1920 pixels. The images were captured to ensure the facial features are clear and can be accurately recognized by the model.

2.2.2 Classes and Labels

Each student in the dataset is represented as a separate class, with each image assigned to a corresponding label. There are between 638 and 661 labeled images for each team member, based on the number of images available. The labels, which were generated using Roboflow, allow the model to learn to distinguish between different team member by identifying their unique facial features.

2.2.3 Conditions of Image Capture

The images in the dataset were taken under a variety of conditions to simulate real-world classroom environments. These include low-light conditions, outdoor environments, and images taken at low resolutions. These diverse conditions ensure that the model can recognize faces in less-than-ideal situations, such as varying lighting or poor image quality.

2.2.4 Preprocessing

Once labeled, the images underwent preprocessing before being used for model training. This included resizing the images to a standard resolution and normalizing pixel values to maintain consistency. Additionally, data augmentation techniques such as random rotations and zooms were applied to enhance the dataset and make the model more robust to different face angles and expressions

2.3 Algorithm or Model

For this project, we utilized the YOLOv8 model for face recognition and detection. YOLO (You Only Look Once) is a convolutional neural network known for its accuracy and speed in object detection tasks, making it suitable for real-time applications such as face recognition. The model was trained using a dataset of labeled faces to allow it to distinguish individual team members. We used the medium variant of YOLOv8, which balances accuracy and computational efficiency, making it a good choice for our dataset size and target application.

```
import os
from ultralytics import YOLO

model = YOLO("yolov8m.pt")

train_results = model.train(data="data.yaml", epochs=100, imgsz=640)
```

2.4 Testing

2.4.1 Testing Procedures

The testing process for the Classify App focuses on validating the model's accuracy, functionality, and security compliance. Specific scenarios, such as low-light conditions, angled views, and multiple faces in a frame, are incorporated to assess the system's performance. Procedures include functional verification of features like registration, login, and attendance recognition, ensuring each works as expected across different device and environmental conditions.

2.4.2 Testing Metrics

Key testing metrics, including accuracy, precision, and compliance with security standards, are measured to assess the model's performance. Accuracy metrics gauge the model's reliability across diverse lighting and angle variations, while security tests validate data encryption during capture and storage. These metrics provide insight into both functional performance and data safety.

2.5 Evaluation Metrics

Evaluation metrics encompass recognition accuracy, processing speed, and resilience under varied conditions. Accuracy is measured to ensure an 85

Chapter 3

Problem

Chapter 4

Intelligence System

4.1 System Architecture

Sample paragraph: The system architecture is designed to support real-time face recognition for multiple students. It consists of a camera module, a preprocessing unit, and a face recognition model, all of which integrate seamlessly for efficient attendance marking.

4.2 System Workflow

Sample paragraph: The workflow begins with the camera capturing an image, followed by preprocessing, which includes resizing and normalization. The preprocessed image is then fed into the recognition model, which identifies students and logs their attendance.

Chapter 5

Project Documentation

5.1 Implementation

The implementation of this system involves integrating key components: **image capture**, **face detection**, **face classification**, and **attendance logging**. Each component contributes to the system's capability to perform real-time, accurate face recognition.

- **Image Capture**: The system initiates by capturing images through a connected camera. This module is designed to automatically capture facial images when individuals are within a defined range.
- **Face Detection**: Using the **OpenCV** library, the system detects face locations within the captured images. This step is optimized for quick, precise detection, even under varied lighting conditions.
- **Face Recognition with YOLOv8m**: The **YOLOv8m** model, trained on a specialized dataset, classifies detected faces accurately. The model is implemented using **TensorFlow**, allowing fast inference within the application.
- **Attendance Logging**: Recognized faces are logged in the attendance module, with each entry stored in a database, capturing time and identity details. This module can also generate reports for daily or monthly attendance monitoring.

5.2 Results and Discussion

5.2.1 Model Training and Evaluation

- **Dataset and Annotation**: The dataset used for training contains 3,873 face images across six classes (Ajul, Izzata, Joy, Kelvin, Lia, and Trisman). The **YOLOv8m** model was selected for its proficiency in diverse detection scenarios.
- **Training Process**: The YOLOv8m model was trained using the **AdamW** optimizer with a learning rate of **0.001** and momentum of **0.9** over **100 epochs**. Data augmentations such as blurring and grayscale conversion improved robustness. A validation set of **774 images** ensured generalization.

5.2.2 Precision-Recall Analysis

The **Precision-Recall curve** yielded an average precision of **0.995** across all classes, indicating strong class distinction. Each class (Ajul, Izzata, Joy, Kelvin, Lia, and Trisman) scored a precision of **0.995** at high recall, demonstrating effective classification with minimal errors.

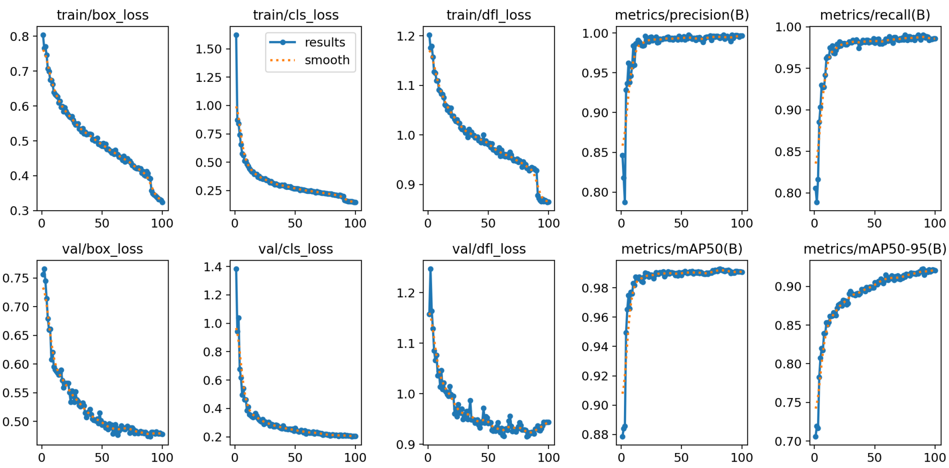


Figure 5.1: Training Results of YOLOv8m Model

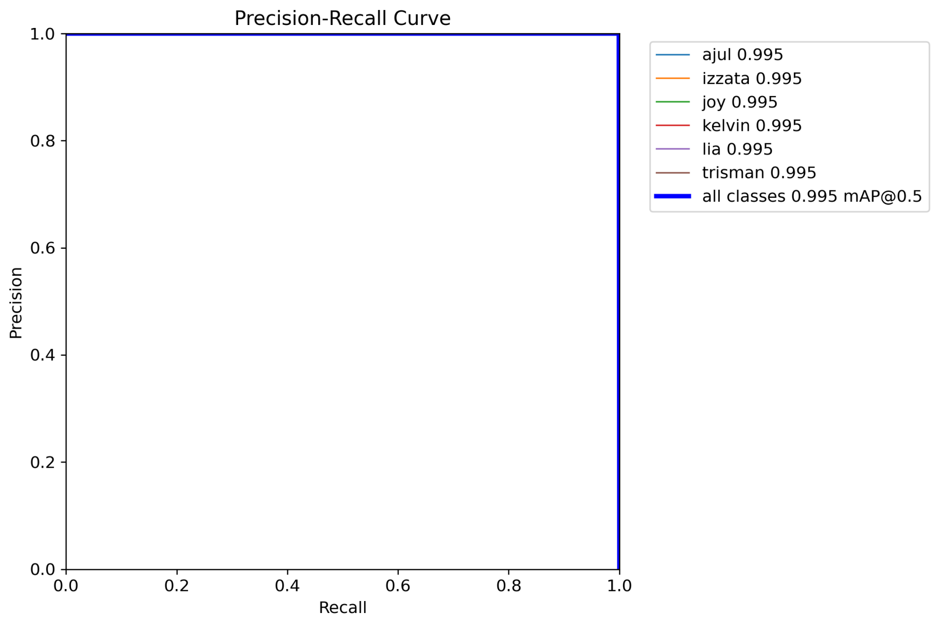


Figure 5.2: Precision-Recall Curve for YOLOv8m Model

5.2.3 F1-Confidence Analysis

The **F1-Confidence curve** shows an F1 score of **1.00** for all classes at a confidence level of **0.8**. This suggests the model's high precision and accuracy. The F1 score remains stable at higher confidence levels, reinforcing its reliability in face classification.

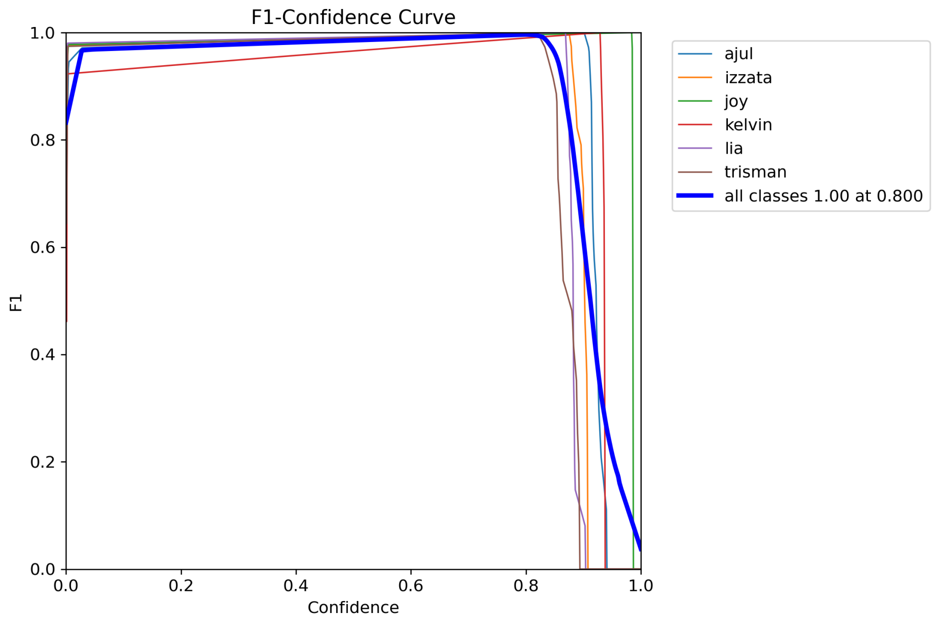


Figure 5.3: F1-Confidence Curve for YOLOv8m Model

5.2.4 Performance Summary

The model achieved a mean Average Precision (mAP@0.5) of **0.995**, demonstrating strong performance in face recognition. This level of accuracy positions the model as suitable for applications demanding high precision, such as attendance systems or security monitoring.

5.3 Conclusion

The YOLOv8m model's high precision, recall, and F1 scores across all classes indicate its effectiveness in recognizing and differentiating between individuals with minimal misclassification. The model's performance confirms its viability for deployment in real-world applications that require accurate and reliable face recognition.

Appendix A

Appendix