Helmet Detection Using YOLOv5

Introduction

Helmet detection is a crucial task in ensuring safety, particularly for motorcyclists. Automatic detection systems can help enforce helmet-wearing rules and prevent accidents. In this project, we utilize YOLOv5, a state-of-the-art object detection model, to develop a helmet detection system. Python 3.11 is used for implementation, with Google Colab and Thonny Python as the development environments.

Objective

The objective of this project is to develop an automated helmet detection system using YOLOv5. The system should accurately identify and localize helmets in images or video frames. The goal is to achieve high detection accuracy and provide a robust solution that can be used in real-world applications, such as traffic monitoring and safety enforcement.

Methodology

YOLOv5 (You Only Look Once version 5) is an advanced object detection model that processes entire images in one go to identify and localize objects. It is known for its speed and accuracy. Python 3.11 is the programming language used for this project. Google Colab provides a cloud-based environment for training the YOLOv5 model, taking advantage of its GPU support. Thonny Python is used for local development and testing.

The project follows these key steps:

1. Data Collection and Preprocessing: Collecting images of motorcyclists with and without helmets. Annotating the images for training.

2. Model Training: Using YOLOv5 in Google Colab to train the model on the annotated dataset.

3. Inference: Implementing the trained model in Thonny Python to detect helmets in real-time or from images.

Implementation

1. Setting Up the Environment

- Google Colab: Import necessary libraries, clone the YOLOv5 repository, and set up the environment.

- Thonny Python: Install dependencies and configure the local environment for running the YOLOv5 model.

2. Training the Model

- Load and preprocess the dataset. Split it into training and validation sets.
- Train the YOLOv5 model on Google Colab using the processed dataset.
- Monitor training metrics such as loss, accuracy, and validation performance.

3. Running Inference

- Use the trained model in Thonny Python to detect helmets in images or video frames.
- Implement real-time detection using a webcam or pre-recorded video.
- Example code snippet:

```
import torch
from PIL import Image
from yolov5 import YOLOv5

# Load model
```

```
model = YOLOv5('yolov5s.pt')


# Load image

img = Image.open('helmet.jpg')


# Inference

results = model.predict(img)


# Display results

results.show()
```

Results

The trained YOLOv5 model successfully detected helmets in various test images and video frames. The model achieved high accuracy, with precise localization of helmets. Key performance metrics such as precision, recall, and F1-score were evaluated to ensure robustness. The model's performance on real-time data was also tested, demonstrating its potential for deployment in safety monitoring systems.

Conclusion

This project successfully developed an automated helmet detection system using YOLOv5. The model demonstrated high accuracy and reliability, making it suitable for real-world applications. Future work could focus on improving detection speed and integrating the system into larger safety enforcement frameworks. Additionally, further training on diverse datasets could enhance the model's generalization to various environments.

References

1. YOLOv5 Documentation: https://github.com/ultralytics/yolov5

2. Python 3.11 Documentation: https://docs.python.org/3.11/

3. Google Colab Documentation: https://colab.research.google.com/

4. Thonny Python IDE: https://thonny.org/