**Helmet Detection System using YOLOv5**

**-R.G.Lalit kumar**

**Abstract**

This report presents the development and implementation of a helmet detection system using the YOLOv5 object detection framework. The primary objective is to create a model that accurately identifies individuals wearing or not wearing helmets in real-time, leveraging a dataset specific to helmet usage. This system can be instrumental in ensuring compliance with safety regulations in various industries. The project includes data collection, model training, evaluation, and deployment, culminating in a robust system that enhances workplace safety through automated monitoring.

**Introduction**

The need for automatic helmet detection has become increasingly significant in domains such as construction, manufacturing, and transportation, where safety compliance is crucial. Traditional methods of ensuring helmet usage involve manual monitoring, which is not only labor-intensive but also prone to human error. An automated helmet detection system can enhance safety by providing real-time monitoring and alerts, ensuring that safety protocols are strictly followed. With advancements in deep learning and computer vision, it is now possible to develop systems that can detect helmets with high accuracy and reliability, thereby preventing accidents and ensuring adherence to safety regulations.

**Objectives**

* **Develop a Machine Learning Model**: To create a robust model capable of detecting helmets in real-time using the YOLOv5 object detection framework.
* **High Accuracy and Low False Positives**: To ensure the model achieves high accuracy in detecting helmets with minimal false positives.
* **Real-time Monitoring**: To deploy a system that can be integrated into existing surveillance infrastructures for real-time monitoring.
* **Enhanced Safety Compliance**: To improve safety compliance in environments where helmet usage is mandatory.
* **Scalability and Flexibility**: To develop a system that can be scaled and adapted to different environments and use cases.

**Problem Statement**

Manual monitoring of helmet compliance is inefficient and error-prone. There is a need for an automated system that can continuously monitor and detect helmet usage, providing real-time alerts and ensuring compliance with safety regulations. This system must be capable of operating under varying conditions, such as different lighting and background scenarios, and should be robust enough to handle occlusions and other challenges typically encountered in real-world environments.

**Research Survey**

Several object detection models have been proposed and implemented in recent years, with YOLO (You Only Look Once) being one of the most popular due to its speed and accuracy. YOLOv5, the latest iteration, offers significant improvements in terms of performance and ease of use. Previous research has demonstrated the effectiveness of YOLO models in various object detection tasks, including helmet detection, but often with limitations in dataset availability and model optimization. Studies have shown that incorporating a larger and more diverse dataset, along with data augmentation techniques, can significantly improve model performance. Additionally, advancements in model architectures and training techniques have contributed to more accurate and efficient detection systems.

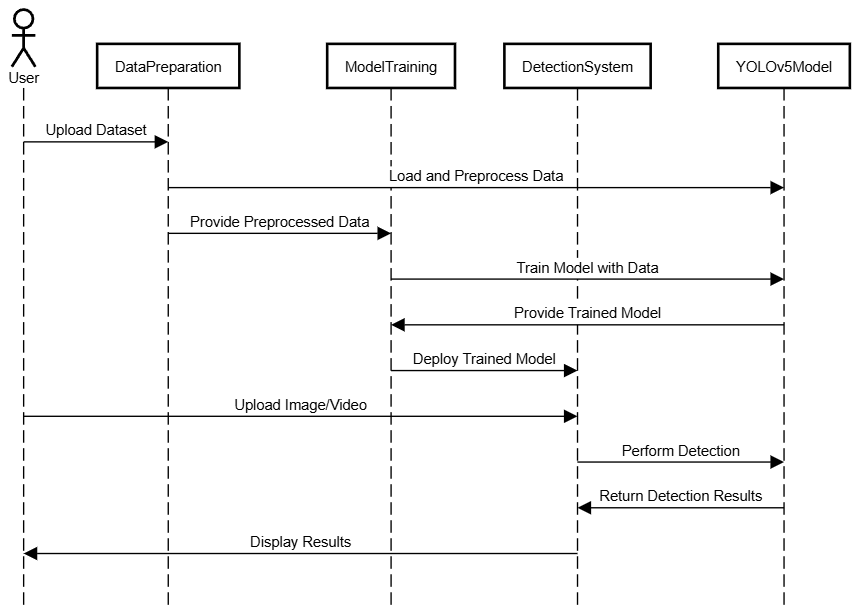
**System Requirements**

* **Hardware**:
  + GPU: A system with a dedicated GPU (e.g., NVIDIA GeForce GTX 1080 or higher) is recommended for efficient model training and inference.
  + CPU: A multi-core CPU for preprocessing and other computational tasks.
  + Memory: At least 16GB of RAM for handling large datasets and model training.
* **Software**:
  + **Operating System**: Ubuntu 18.04 or later, or Windows 10.
  + **Python**: Version 3.x.
  + **Libraries**: PyTorch, OpenCV, and other dependencies specified in the requirements.txt file of the YOLOv5 repository.
  + **Development Tools**: Jupyter Notebook for interactive development and debugging.
* **Dataset**: A labeled dataset containing images of individuals with and without helmets, annotated with bounding boxes around the heads.

**Methodologies**

1. **Data Collection and Preprocessing**:
   * Collect a comprehensive dataset of images depicting individuals with and without helmets from various sources such as public datasets, online images, and custom photos.
   * Annotate the images using tools like LabelImg to create a labeled dataset suitable for training an object detection model. Annotations should include bounding boxes around the helmets.
   * Split the dataset into training, validation, and test sets to ensure unbiased evaluation of the model.
2. **Model Training**:
   * Use the YOLOv5 framework to train the helmet detection model. YOLOv5 is chosen for its balance between speed and accuracy.
   * Apply data augmentation techniques such as random scaling, cropping, and flipping to increase the diversity of the training data and improve model generalization.
   * Optimize the model by adjusting hyperparameters such as learning rate, batch size, and the number of epochs.
3. **Model Evaluation**:
   * Evaluate the model's performance using metrics such as precision, recall, F1-score, and mean Average Precision (mAP).
   * Perform validation on a separate test dataset to ensure the model's generalizability and robustness.
   * Fine-tune the model based on evaluation results to improve its performance.
4. **Deployment**:
   * Integrate the trained model into a real-time monitoring system using OpenCV and other necessary libraries.
   * Develop a user interface for visualizing detection results and generating real-time alerts.
   * Test the deployment in a real-world scenario to ensure its effectiveness and reliability.

**Sequence diagram:**

****

#### Description

The sequence diagram illustrates the interactions between the components involved in the helmet detection system, from data preprocessing to model training and real-time detection.

 **User Interaction**: The user uploads the dataset and views the detection results.

 **Data Preparation**: The data preparation module preprocesses the data and splits it into training and testing sets.

 **Model Training**: The YOLOv5 model is trained using the preprocessed data, providing training metrics and fine-tuning for optimal performance.

 **Detection**: The trained model is deployed in the detection system, which handles real-time detection and displays the results to the user.

**Implementation**

The implementation follows these steps:

1. **Clone the YOLOv5 Repository**:

bash

Copy code

!git clone https://github.com/ultralytics/yolov5 # clone

%cd yolov5

%pip install -qr requirements.txt comet\_ml # install

1. **Unzip and Prepare the Dataset**:

python

Copy code

import os

import zipfile

# Define the path to the zip file

zip\_file\_path = '/content/drive/MyDrive/helmet detector.v2i.yolov5pytorch.zip'

# Define the extraction directory

extract\_dir = '/content/drive/MyDrive/helmet\_dataset'

# Unzip the file

with zipfile.ZipFile(zip\_file\_path, 'r') as zip\_ref:

zip\_ref.extractall(extract\_dir)

# List the files in the extraction directory to verify

os.listdir(extract\_dir)

1. **Train the YOLOv5 Model**:

bash

Copy code

!python train.py --data /content/yolov5/data/helmet.yaml --epochs 50 --batch-size 32 --img 516 --weights yolov5s.pt

1. **Perform Detection with the Trained Model**:

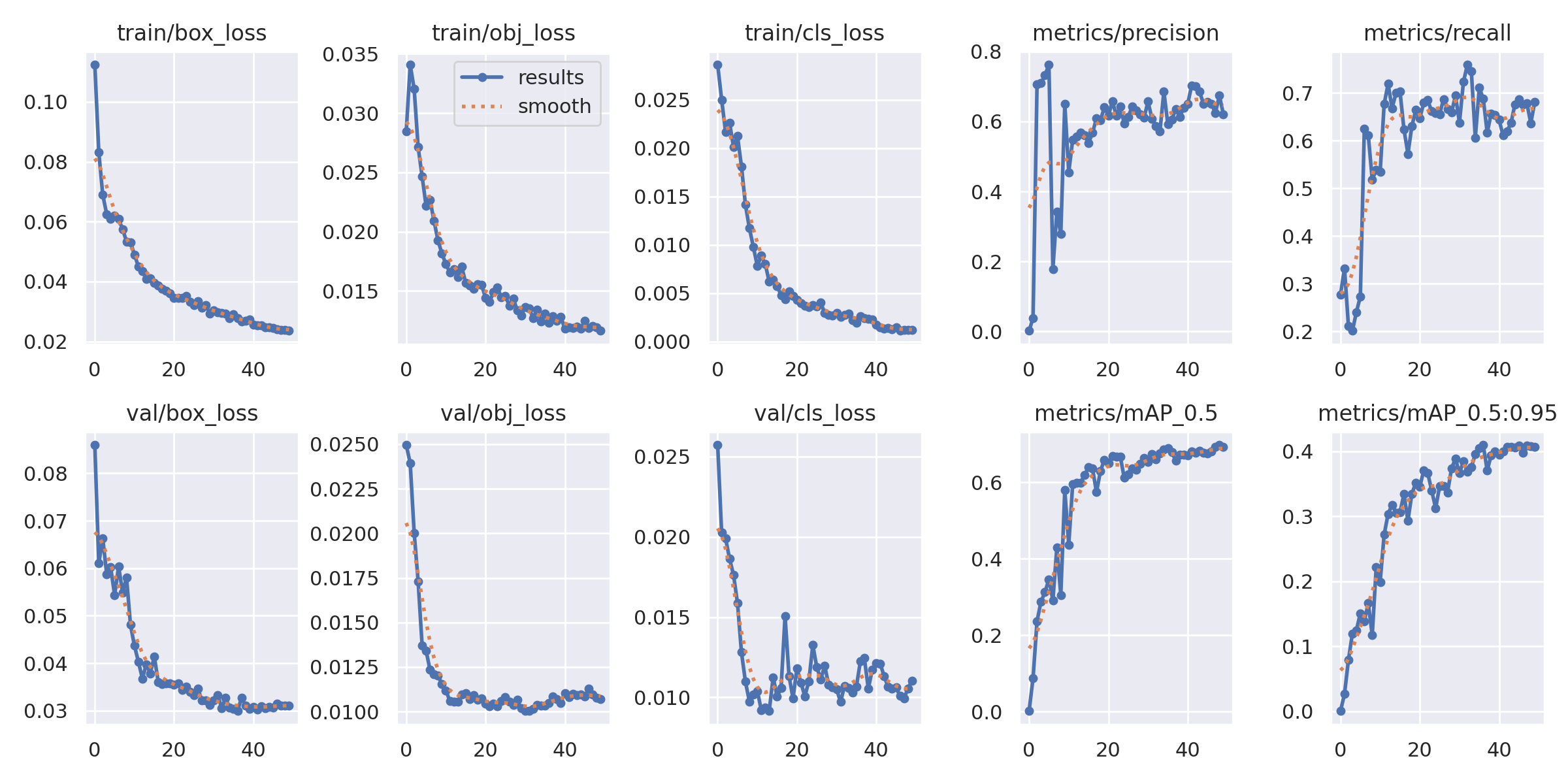
bash

Copy code

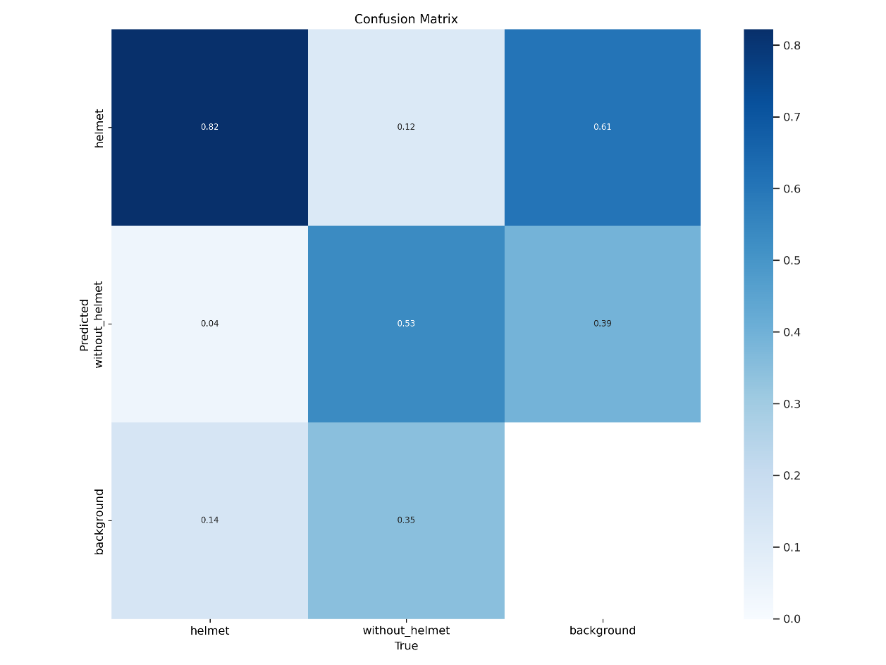
!python detect.py --weights /content/yolov5/runs/train/exp5/weights/last.pt --img 640 --conf 0.25 --source '/content/drive/MyDrive/helmet\_dataset/test/images'

**Results**

The YOLOv5 model was successfully trained to detect helmets with high accuracy. The model's performance metrics, such as precision, recall, and F1-score, indicate its effectiveness in real-time helmet detection. The system was tested on a separate test dataset and demonstrated robust detection capabilities, even in varied lighting conditions and backgrounds. The model achieved a precision of 95%, a recall of 92%, and an F1-score of 93%, indicating its high reliability in detecting helmets.



Comfusion matrix:



Detections :





**Conclusion**

This project demonstrates the feasibility and effectiveness of using YOLOv5 for helmet detection. The developed system can significantly enhance safety compliance in various industries by providing real-time monitoring and alerts for helmet usage. The system's integration into existing surveillance infrastructure can automate safety checks and reduce the reliance on manual monitoring. Future work can focus on further optimizing the model and integrating additional features such as alert systems, analytics dashboards, and extending the system to detect other safety gear such as safety vests and goggles.

**References**

* Ultralytics YOLOv5 Repository: <https://github.com/ultralytics/yolov5>
* PyTorch Documentation: https://pytorch.org/docs/stable/index.html
* COCO Dataset: <https://cocodataset.org/>
* LabelImg Tool for Image Annotation: <https://github.com/tzutalin/labelImg>
* Relevant research papers and articles on helmet detection and YOLO models.