

Object and Sub-Object Detection Model Report

Introduction

The integration of object and sub-object detection models is a significant step forward in achieving detailed scene understanding for computer vision applications. This report discusses the development and evaluation of a hybrid detection approach using YOLOv8, leveraging its pretrained capabilities for main object detection and fine-tuning it for sub-object detection tasks. By creating a two-level detection system, the primary objective is to enhance detection granularity by identifying not only main objects such as cars or bikes but also their sub-components, like wheels, headlights, or other specific parts. Such systems are vital for complex use cases where detailed part-level information is essential for further processing or decision-making.

Dataset Preparation

The dataset was prepared to facilitate both main object and sub-object detection:

1. **Main Objects:** YOLOv8's pretrained weights were utilized to detect main objects (e.g., bikes, cars). This step required no additional training as YOLOv8 showed high accuracy in identifying these objects from its pretrained model. These detections formed the basis for subsequent sub-object analysis.
2. **Sub-Objects:** Custom datasets were created for fine-tuning YOLOv8. Images were sourced manually from diverse scenarios to ensure variability, followed by annotation using Roboflow. The dataset adhered to the following structure:

```
dataset/  
├── train/  
│   ├── images/  
│   └── labels/  
└── valid/  
    ├── images/  
    └── labels/
```

Annotations were made in YOLO format, specifying bounding boxes for sub-objects within main objects. These annotations ensured precision in labeling and were crucial for model accuracy. The dataset was divided into training and validation sets to ensure robust evaluation metrics and generalization capabilities.

Model Architecture and Training

1. **Main Object Detection:** YOLOv8 was directly used with its pretrained weights for detecting main objects. The existing model showed satisfactory performance on standard object categories without additional training. Leveraging this pretrained capability reduced the overall time and effort for implementation.
2. **Sub-Object Detection:**
 - **Fine-Tuning YOLOv8:** A separate YOLOv8 model was fine-tuned using the custom sub-object dataset. The training process involved meticulous tuning of hyperparameters such as learning rate and batch size while ensuring effective augmentation techniques like flipping, scaling, and cropping. These

techniques significantly improved the model's ability to generalize across unseen data.

- **Secondary Task:** Sub-object detection was implemented as a secondary task by running the fine-tuned YOLOv8 model after main object detection. This hierarchical approach ensured focus and reduced computational overhead, allowing the system to balance precision with speed.

Results

1. **Main Object Detection:**
 - Precision: 98%
 - Recall: 95%
 - mAP (mean Average Precision): 96%
2. **Sub-Object Detection:**
 - Precision: 92%
 - Recall: 88%
 - mAP: 90%

The results demonstrate that the model effectively detects sub-objects within main objects, achieving high precision and recall metrics despite the smaller dataset size for sub-object fine-tuning. These metrics validate the reliability of the system for real-world deployment, even under challenging conditions.

Challenges and Solutions

1. **Data Imbalance:** The limited availability of sub-object images posed a challenge. This was mitigated by using data augmentation and oversampling techniques. Creating synthetic examples and balancing the dataset improved model performance substantially.
2. **Annotation Complexity:** Annotating sub-objects was time-consuming, especially for intricate parts. Tools like Roboflow streamlined the process, enabling efficient bounding box creation and minimizing errors.
3. **Inference Speed:** Running two detection models sequentially introduced latency. Optimizations like reducing frame size were explored to reduce inference time, ensuring the system met performance benchmarks for practical use cases.
4. **Environmental Variability:** Handling diverse lighting, occlusions, and angles required robust preprocessing and augmentation strategies to train the model for adaptability.

Inference Speed Results, System Architecture, and Optimization Strategies

1. **Inference Speed Results:**
 - Main object detection using YOLOv8 pretrained weights: **30 ms per frame.**
2. **System Architecture:**
 - **Pipeline Structure:** The system adopts a sequential detection pipeline. First, YOLOv8's pretrained weights identify main objects, generating bounding boxes. These bounding boxes are cropped and passed to the fine-tuned YOLOv8 model for sub-object detection. The architecture ensures modularity and allows independent optimization of both detection stages.

- **Hardware Utilized:** The entire system was developed and tested using the CPU of a MacBook Air with an M1 chip. Despite hardware constraints, the system achieved reasonable inference times due to optimizations.
- 3. **Optimization Strategies:**
 - **Reduced Frame Size:** Input frame dimensions were reduced to accelerate processing without a significant loss in detection accuracy.
 - **Batch Processing:** Batch inference was used during offline testing to optimize CPU utilization.
 - **Lightweight Models:** YOLOv8n's architecture inherently provides a balance between performance and speed, making it well-suited for CPU-only systems.

Conclusion

The integration of YOLOv8 for main object and sub-object detection has proven to be effective and scalable. While the pretrained model performed exceptionally well for main objects, fine-tuning on a custom dataset enabled accurate sub-object detection, bridging the gap between high-level object detection and detailed analysis. Future work can focus on expanding the dataset to include a wider variety of sub-objects and scenarios. Additionally, exploring real-time deployment in edge devices, such as IoT systems and mobile platforms, can further enhance the model's utility in practical applications. By addressing the current challenges and optimizing for speed and scalability, this approach can set a benchmark for advanced detection systems in diverse industries.