

Multi-Class Image Classification Using YOLO

Objective

The aim of this project is to develop a multi-class image classification model capable of categorizing images into four distinct classes. The model leverages YOLOv8 for feature extraction and fine-tuning, combined with PyTorch for training and evaluation. The goal is to achieve high performance using metrics such as accuracy, precision, and recall.

Approach

1. Dataset Selection

The dataset consists of images divided into four categories:

- Blurred Aadhar
- Clear Aadhar
- Government Documents
- Others

2. Data Splitting

The dataset was manually split into training (80%) and testing (20%) subsets:

- Training Set: Used for model optimization.
- Testing Set: Used to evaluate the performance of the trained model.

3. Data Preprocessing

- Images were resized to 224x224 pixels.
- Data augmentation techniques were applied, including:
 - Random rotations
 - Horizontal flips

- Normalization to a standard range.

4. Model Selection and Feature Extraction

- YOLOv8 was chosen for its efficient backbone and classification head.
- The pretrained yolov8n-cls.pt model was fine-tuned on the selected dataset for better class-specific feature extraction.

5. Training Configuration

The model was trained using the following configuration:

- Epochs: 10
- Batch Size: 16
- Image Size: 224x224
- Optimizer: Stochastic Gradient Descent (SGD)
- Learning Rate Scheduler: Adjusted the learning rate dynamically.
- Device: CPU (adaptable to GPU for faster computation).

6. Evaluation Metrics

To measure the model's performance, the following metrics were employed:

1.Accuracy: Measures overall correctness.

2.Precision: Measures the proportion of true positives among predicted positives.

3.Recall: Measures the proportion of true positives among actual positives.

Code Implementation

The complete implementation is structured for clarity and reproducibility. Below is a summary of the steps taken:

1.Model Initialization:The YOLO model was initialized using the pretrained yolov8n-cls.pt.

2.Model Training:The model was fine-tuned on the training dataset using 10 epochs and a batch size of 16.

3.Model Evaluation:The trained model was evaluated on the testing dataset to compute confusion matrix-based metrics.

4.Metrics Calculation:Precision, recall, and accuracy were computed for each class as well as overall.

Conclusion

This project successfully implemented a multi-class classification model using YOLOv8. The model achieved an **accuracy of 76.2%**, with consistent **precision** and **recall** values across all classes. The use of YOLO as a feature extractor streamlined the pipeline, demonstrating its versatility beyond object detection tasks.