

## ORM Project Documentation

This documentation provides a detailed overview of the **ORM (Object Recognition Model)** project, which involves building an object detection system using **YOLO (You Only Look Once)**. The project includes data preprocessing, model training, inference, and real-time object detection through a user-friendly **Streamlit** interface. Each phase is documented to give a clear understanding of the workflow and implementation details.

### 1. ORM\_Practice.ipynb

- **Purpose:**
  - Test and refine object detection.
- **Key Components:**
  - **Sample Inputs:**
    - Runs model on sample images and videos.
    - Allows testing of different confidence thresholds.
  - **Parameter Optimization:**
    - Adjusts hyperparameters (learning rate, epochs, batch size) for better performance.
  - **Debugging:**
    - Identifies misclassified objects.
    - Fine-tunes model for improved accuracy.
  - **Performance Tuning:**
    - Monitors FPS and processing time.
    - Reduces latency by optimizing model architecture.

## 2. ORM\_EDA.ipynb

- **Purpose:**
  - Exploratory Data Analysis (EDA).
- **Key Components:**
  - **Statistical Summary:**
    - Provides mean, median, mode, variance, and standard deviation.
  - **Visualization:**
    - Generates histograms, box plots, and scatter plots.
    - Helps identify patterns and outliers.
  - **Correlation Matrix:**
    - Shows relationships between different features.
    - Helps identify multi-collinearity.
  - **Outlier Detection:**
    - Uses IQR and Z-score methods to detect and handle outliers.

## 3. ORM\_Preprocessing.ipynb

- **Purpose:**
  - Data cleaning and preprocessing.
- **Key Components:**
  - **Data Loading:**
    - Loads raw data from CSV/Excel files.
    - Handles large datasets efficiently.
  - **Handling Missing Values:**
    - Imputes missing values using statistical methods (mean/median/mode).
    - Drops rows/columns if missing data exceeds a threshold.
  - **Data Type Conversion:**
    - Ensures consistency in data types.
    - Converts categorical data to numerical using encoding.
  - **Scaling:**
    - Applies Min-Max scaling or Standard scaling for consistency.

#### 4. ORM\_Model\_Training.ipynb

- **Purpose:**
  - Train the YOLO model.
- **Key Components:**
  - **Data Splitting:**
    - Splits data into training and validation sets.
    - Uses stratified sampling to maintain class balance.
  - **YOLO Configuration:**
    - Loads YOLO configuration for training.
    - Adjusts batch size, learning rate, and number of epochs.
  - **Training:**
    - Runs the model on the training set.
    - Displays live loss and accuracy updates.
  - **Performance Metrics:**
    - Evaluates model using precision, recall, and F1-score.

#### 5. ORM\_Inference.ipynb

- **Purpose:**
  - Model inference and evaluation.
- **Key Components:**
  - **Model Loading:**
    - Loads trained YOLO model for inference.
    - Uses GPU for faster inference.
  - **Prediction:**
    - Tests model on new data.
    - Generates bounding boxes for detected objects.
  - **Evaluation:**
    - Compares predictions with ground truth.
    - Generates confusion matrix and classification report.
  - **Plotting:**
    - Plots accuracy and loss curves.
    - Highlights improvement areas.

## 6. ORM\_App.py

- **Purpose:**
  - Implements object detection using YOLO and Streamlit.
  - Provides three modes: Image, Video, and Webcam.
- **Key Components:**
  - **Model:** Loads YOLO model (best.pt) for object detection.
    - Uses pre-trained weights to identify objects in different input formats.
    - Ensures quick and accurate detection through GPU acceleration.
  - **Modes:**
    - **Image Mode:** Allows users to upload an image and detects objects within the image.
      - The result is displayed alongside the original image.
    - **Video Mode:** Processes a video frame-by-frame and applies object detection.
      - Displays live progress using a progress bar.
    - **Webcam Mode:** Captures real-time video feed and applies object detection.
      - Handles frame processing and model inference in real-time.
  - **Streamlit UI:**
    - Provides an intuitive interface for selecting modes and setting parameters.
    - Includes buttons to start/stop webcam and close the application.

Link for the demo: [https://drive.google.com/file/d/1scyJi\\_ePuc6KOUtoFvJTc-YOUWmdOjM8/view?usp=drive\\_link](https://drive.google.com/file/d/1scyJi_ePuc6KOUtoFvJTc-YOUWmdOjM8/view?usp=drive_link)