**AI-based Fraud Management System for UID Aadhaar**

1. **Upload Document**
2. **CLassification Model**
3. **a) Use detection algorithm e.g YOLO**

**b) We can directly send the image to ANY ocr API or tesseract**

**YOLOv9**

**Yolov8**

**After the OCR it will send us raw texts from the uploaded document in this case AAdhar**

**Name or Address or sometime as per the need UID as well DOB or phone number**

**Extracted name from AAdhar is Manas**

**DB - Database name is Manash**

**False**

**Extracted Address - Journal Street, Delhi, India, 900900**

**Existing database Address - Mahin Street, Delhi, India, 909099 False**

1. **How we can build this project in terms of UI or API as well**
2. **You do some research on what are the classification model we can use in this project**

**And why do u think this classification model is better**

1. **How we can detect the image bounding box for image detection algorithm and what are the detection algorithm are there we can use**

### **1. Project Overview**

1. **Document Upload: User uploads an Aadhaar document image.**
2. **Classification Model: Identifies the document type to ensure correct processing.**
3. **Object Detection: Detects and extracts sections (name, address, UID) for further processing.**
4. **OCR: Extracts text from detected areas to get raw data.**
5. **Matching Logic: Compares the extracted details with database records to detect mismatches or fraud.**

### **2. Core Components and Model Options**

#### **2.1 Document Classification**

* **EfficientNet or ResNet models: Both are commonly used for document type classification. EfficientNet is faster and uses fewer resources, while ResNet offers higher accuracy.**
* **Why Classification: Helps to verify if the document uploaded is Aadhaar and directs it for further processing.**

#### **2.2 Detection Model**

* **YOLOv8 or YOLOv9: Detects and extracts bounding boxes for key sections (name, address, UID). YOLO models are fast and accurate for real-time detection.**
* **Alternative Options: Faster R-CNN (higher accuracy, slower) or SSD (balance of speed and accuracy).**

#### **2.3 OCR (Optical Character Recognition)**

* **Tesseract OCR: Open-source and customizable, suitable for basic text extraction.**
* **Google Vision API or AWS Textract: High accuracy, especially for complex images. Requires API setup and may have costs.**

### **3. Matching and Verification Logic**

1. **Name Matching: Exact match or phonetic similarity (e.g., Levenshtein Distance).**
2. **Address Matching: Match components like street name, city, and pincode separately.**
3. **UID Matching: Requires an exact match to flag inconsistencies.**
4. **DOB and Phone Number: Simple exact match for verification.**