Pragament Tech Solutions - Project Proposal for AIML Python-Based OMR Sheet Bubble Detection

Project Overview

We seek engineers with experience in Artificial Intelligence and Machine Learning (AI&ML) for a project focused on automating Optical Mark Recognition (OMR) for detecting and interpreting human-marked data on documents. This document outlines the project details, requirements, and benefits.

Project Name

AIML-Powered OMR Sheet Bubble Detection and Alignment Tool

Objective

The objective of this project is to develop an AI&ML-based tool for automated bubble detection and interpretation on OMR sheets, aiming to enhance accuracy and efficiency in processing large volumes of scanned student paper images.

Scope of Work

1. **Dataset analysis:**
    - https://github.com/Pragament/OMR_dataset/tree/main/images
2. **Automated Bubble Detection:**
    - Detect and interpret bubbles marked on OMR sheets.
    - Generate a CSV file with data mapping roll numbers and detected responses for each question.
3. **Sheet Alignment and Marker Detection:**
    - Identify the four corners of the OMR sheet using corner markers.
    - Automatically correct misalignments and fix scanned page orientation.
4. **Template Input for Customization:**
    - Accept human inputs to define custom areas like roll number sections or other specific regions on the sheet.
5. **Data Integration:**
    - Process datasets including PDF files of scanned student paper images and output CSV files containing the detected responses.
6. **Error Handling and Review:**
    - Implement error correction mechanisms to address incomplete or misaligned scans.
    - Provide options for human review and feedback for improved learning.

7. **TensorFlow Lite Integration:**
    - Develop a TensorFlow Lite version of the model to enable seamless integration with a Flutter app.
    - Ensure the Flutter app can process PDF files of scanned student paper images and generate CSV files containing detected responses without requiring Python code within the app.
8. **Reporting and Dashboards:**
    - Develop tools to visualize detection accuracy, error rates, and processed sheet statistics.

Candidate Requirements

Technical Skills:

- Proficiency in Python and AI&ML frameworks such as TensorFlow, PyTorch, or scikit-learn.
- Experience with image processing libraries like OpenCV and PIL.
- Strong knowledge of data handling tools, including pandas and NumPy.

Experience:

- Proven track record in developing AI&ML solutions for document or image processing.
- Familiarity with OMR systems and bubble detection technologies.

Knowledge:

- Deep understanding of supervised learning, computer vision, and document alignment techniques.
- Expertise in preprocessing and augmenting datasets for model training.

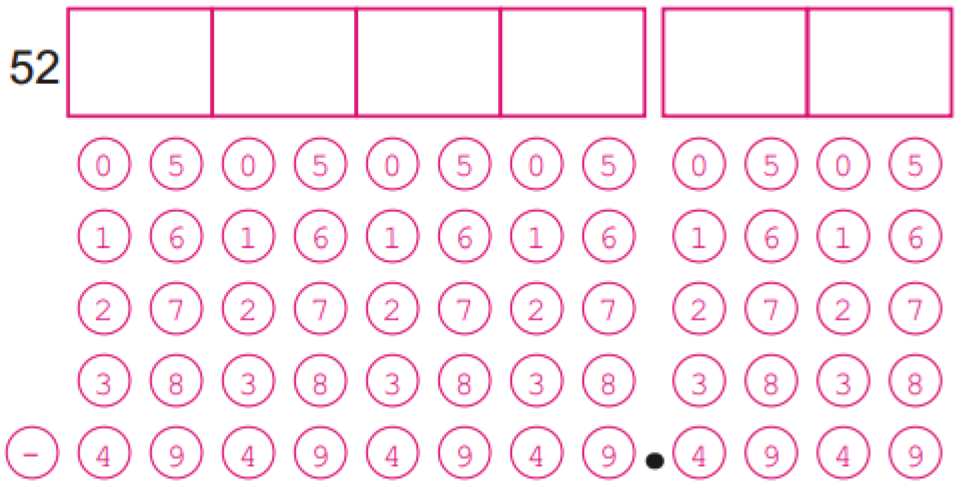
Problem-Solving Skills:

- Ability to troubleshoot issues related to sheet misalignment, marker detection, and template mismatches.
- Strong optimization skills to improve model accuracy and efficiency.

Compensation for each milestone:

1. ₹3000 TensorFlow Lite version of the model(minimum 2 OMR templates like NEET & JEE Main, Bubble Detection Accuracy: ≥ 60%) to enable seamless integration with a Flutter/Android Dart/Java/Kotlin app. Ensure the Android app can process PDF files of scanned student paper images and generate CSV files containing detected responses without requiring Python code within the app. **REQUIRED** milestone for any other milestone compensation.

2. ₹1000 Python Visualization Tools and Techniques - Refer https://github.com/Udayraj123/OMRChecker/wiki/%5Bv1%5D-Rich-Visuals for ideas
3. ₹1000 Python Error Analysis Techniques
4. ₹4000 Good metrics:
    - Bubble Detection Accuracy: ≥ 85%
    - Alignment Precision: Corner markers detected correctly in ≥ 90% of sheets.
    - Error Rate: < 10% misaligned or undetected sheets.
    - Human Review Integration: Human-in-the-loop system successfully corrects ≥ 90% of detected errors.
5. ₹9000(₹4000 Good + ₹5000 Very Good) Very Good metrics:
    - Bubble Detection Accuracy: ≥ 92%
    - Alignment Precision: Corner markers detected correctly in ≥ 95% of sheets.
    - Error Rate: < 5% misaligned or undetected sheets.
    - System Efficiency: Average processing time reduced by ≥ 15% compared to initial implementation.
6. ₹15000(₹4000 Good + ₹5000 Very Good + ₹6000 Awesome) Awesome metrics:
    - Bubble Detection Accuracy: ≥ 97%
    - Alignment Precision: Corner markers detected correctly in ≥ 98% of sheets.
    - Error Rate: < 2% misaligned or undetected sheets.
    - System Robustness: Performance maintains accuracy across diverse templates and scan quality variations.
7. ₹5000 Numeric type OMR with Very Good metrics.



Certification:

Official certification upon successful completion.

Internship Opportunity:

Potential internship opportunities based on performance.

Some tools and techniques that can help visualize results and perform error analysis to identify model weaknesses in the AIML-powered OMR Sheet Bubble Detection project:

Visualization Tools and Techniques

1. **Error Heatmaps**:
   - Use heatmaps to visualize the areas where the model struggles, such as incorrect bubble detection or misaligned sheets.
   - **Tool**: Matplotlib or Seaborn in Python.
2. **ROC Curves and Precision-Recall Curves**:
   - Analyze the trade-offs between true positive rates and false positive rates or precision and recall.
   - **Tool**: scikit-learn's built-in plotting functions.
3. **Confusion Matrices**:
   - Summarize prediction accuracy and errors for bubble detection tasks.
   - **Tool**: scikit-learn for generating and visualizing confusion matrices.
4. **Bounding Box Visualization**:
   - Overlay predicted corner markers or detected bubbles on the original OMR sheet images for direct comparison.
   - **Tool**: OpenCV or Matplotlib.
5. **Error Trend Dashboards**:
   - Create dashboards to track detection errors over time or across datasets.
   - **Tool**: Dash or Streamlit for interactive data visualization.
6. **Interactive Annotation Review**:
   - Build an interface for humans to review and correct errors in detected data.
   - **Tool**: Label Studio for annotation and error review.

Error Analysis Techniques

1. **Failure Case Categorization**:
   - Group errors into categories, such as misaligned sheets, incorrect bubble detection, or missed bubbles, to identify patterns.
2. **Dataset Inspection**:
   - Visualize and inspect dataset variations to identify outliers or challenging cases.
   - **Tool**: pandas for data handling and Matplotlib/Seaborn for visualization.

3. **Overlay Analysis**:
    - Compare actual vs. predicted bounding boxes for bubbles or markers to measure alignment errors.
4. **Model Performance Comparison**:
    - Test performance across subsets (e.g., noisy scans, low-quality images) to pinpoint weaknesses.
    - **Tool**: Jupyter Notebooks for analysis.
5. **Data Augmentation Testing**:
    - Assess model robustness by applying synthetic transformations like rotation or noise and visualizing performance changes.
6. **Confidence Scoring Visualization**:
    - Plot confidence scores of predictions to identify low-confidence outputs and areas for model improvement.

These tools and techniques will not only help pinpoint specific issues but also make it easier to present and interpret results for stakeholders working on the project.