PixMind

Computer Vision PSM Project
Presentation 1 - Project Status

Presented by: Omar Hawas
Project Team:

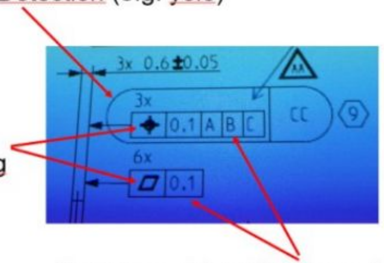
- Omar Hawas
- Alina
- Shaheryaar
- Ahmed
- Areeb Khan
- Mugesh

Agenda

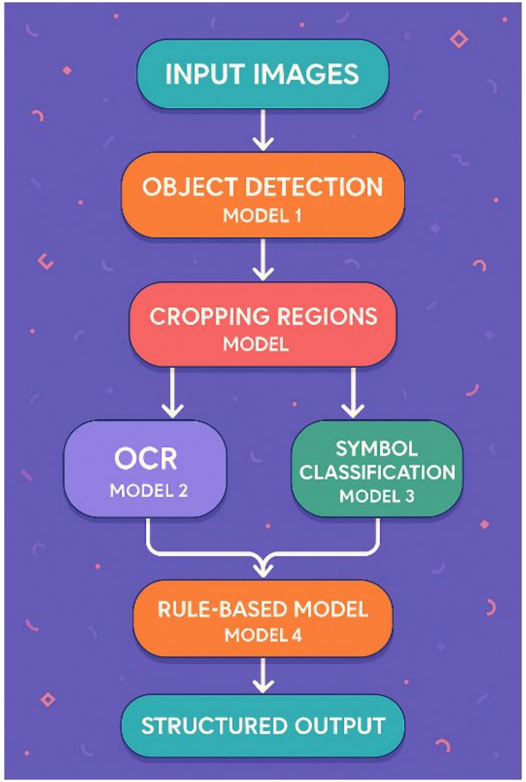
- Project’s goals
- Project flow
- Goals achieved
- Models Used
- Assumptions
- Constraints
- Documentation
- Further Steps to be taken

Project Goals

N°	Target Description	Success Criteria (Measurability)
Z1	Automate evaluation of mechanical drawings to reduce manual checks.	YOLO model detects relevant drawing area (bounding boxes). Manual check no longer required for initial analysis.
Z2	Detect the rectangle that contains technical info like values, tolerances, and part IDs.	Model identifies and localizes rectangles with >90% accuracy on the test set.
Z3	Use OCR to extract relevant text/numbers from the detected rectangles.	OCR extracts text with >85% accuracy and maps it into a structured format (e.g., JSON).
Z4	Identify critical tolerances and feasibility issues based on extracted data and predefined rules.	System highlights values outside manufacturing specs automatically. Flagging works in >90% of cases.
Z5	Re-evaluate updated drawings automatically and highlight changes.	Updated parts are compared to previous versions; all changes are clearly marked.
Z6	Develop a user interface for reviewing detected data and making manual corrections if needed.	Web interface displays detected values and allows edits. 100% of processed files are accessible via the UI.
Z7	Log system decisions and create a summary report for each drawing processed.	Each processed drawing has a log file containing extracted data, tolerances, and the final feasibility decision.



Project's Flow (Architecture)



Pipeline Stages Breakdown

1. Input Images

- Raw images are collected from various sources.
- These images are prepared for analysis and passed to the first model.

2. Object Detection – `Model 1`

- **YOLO-based model** detects and classifies regions of interest (ROIs).
- Outputs bounding boxes for each object in the image.

3. Cropping Regions

- Detected ROIs are cropped and separated for parallel processing.
- These cropped regions are inputs to the next two models.

4. OCR Extraction – `Model 2`

- Applies optical character recognition to extract any text from the cropped regions.
- Uses Tesseract or a deep-learning OCR engine.
- Outputs structured text.

5. Symbol Classification – `Model 3`

- Classifies non-text elements such as icons, logos, or graphical symbols.
- A CNN-based classifier identifies relevant symbol categories.

6. Rule-Based Logic – `Model 4`

- Merges the results from OCR and symbol classification.
- Applies business or logical rules to validate and organize the data.

7. Structured Output

- Final output is exported in structured formats like: .xlsx
- This output is ready for analysis, reports

What Has been achieved so far

✅Collecting Synthetic Data (Alternative for not receiving the data from the customer)

✅Building up different scripts for Data annotation(Symbol Detection and Data classification)

✅Verify the annotation of the Data collected

✅Building up both models (OCR Extraction, Symbol Classification)

✅Formatting the data as pdf instead of png as it eases the process of the OCR and the text extraction

✅ Successfully Build up the CI/CD Integration and Deployment set up for the Project.

Model Used and Rationale

The classification model currently in use is a Convolutional Neural Network (CNN), chosen for the following reasons:

- CNNs perform well on visual/textual patterns common in scanned document images.
- The model architecture allows for end-to-end learning from preprocessed document images or OCR-extracted text representations.
- It can be fine-tuned or extended for multi-label classification as required.

Alternatives considered included traditional machine learning (e.g., SVM, Random Forest) which were found to underperform on noisy OCR inputs. Future iterations may explore Transformer-based architectures (e.g., BERT or LayoutLM) once more realistic datasets are available.

Assumptions About Expected Results

The current assumptions being tested using synthetic data are:

- The synthetic documents are a close enough approximation of the real data structure to allow for meaningful model pretraining.
- The OCR engine can extract structured and semi-structured text with reasonable accuracy, especially when synthetic image quality is high.
- Classification results from synthetic data training can be used as a baseline or reference for future comparison once real data becomes available.

These assumptions will be re-evaluated after real data is acquired.

Constraints

❌ **NOT Receiving the main data out from the and mainly relying on the synthetic data that we have created**

❌Computer power in order to train the models we have (Classification models, OCR Extraction)

Documentation

✅Initial Situation and Motivation

✅Constraints

✅Project & Goals

✅Deliverables

✅System View

✅Problem Statement

✅Basic Research (Glossary)

❌Feature Engineering **(Reason: Not receiving the real data yet)**

❌Labeling the data **(Reason: Not receiving the real data yet)**

❌Requirements and constraints for the model **(Reason: Not receiving the real data yet)**

❌Decide your ML approach **(Reason: Not receiving the real data yet)**

Further Steps to be taken

1. Receive the Data from the Customer
2. Annotate it with the Python Scripts that are already Pre-implemented
3. Verify the Results of the annotation
4. Train the Models that are already Pre-Built with the data
5. Compare the results with the expected ones and adjust the models (if errors in detecting the text or the text presented)