Rail Symbol Detect Project: Automated BOM Generation of Railways Schematics

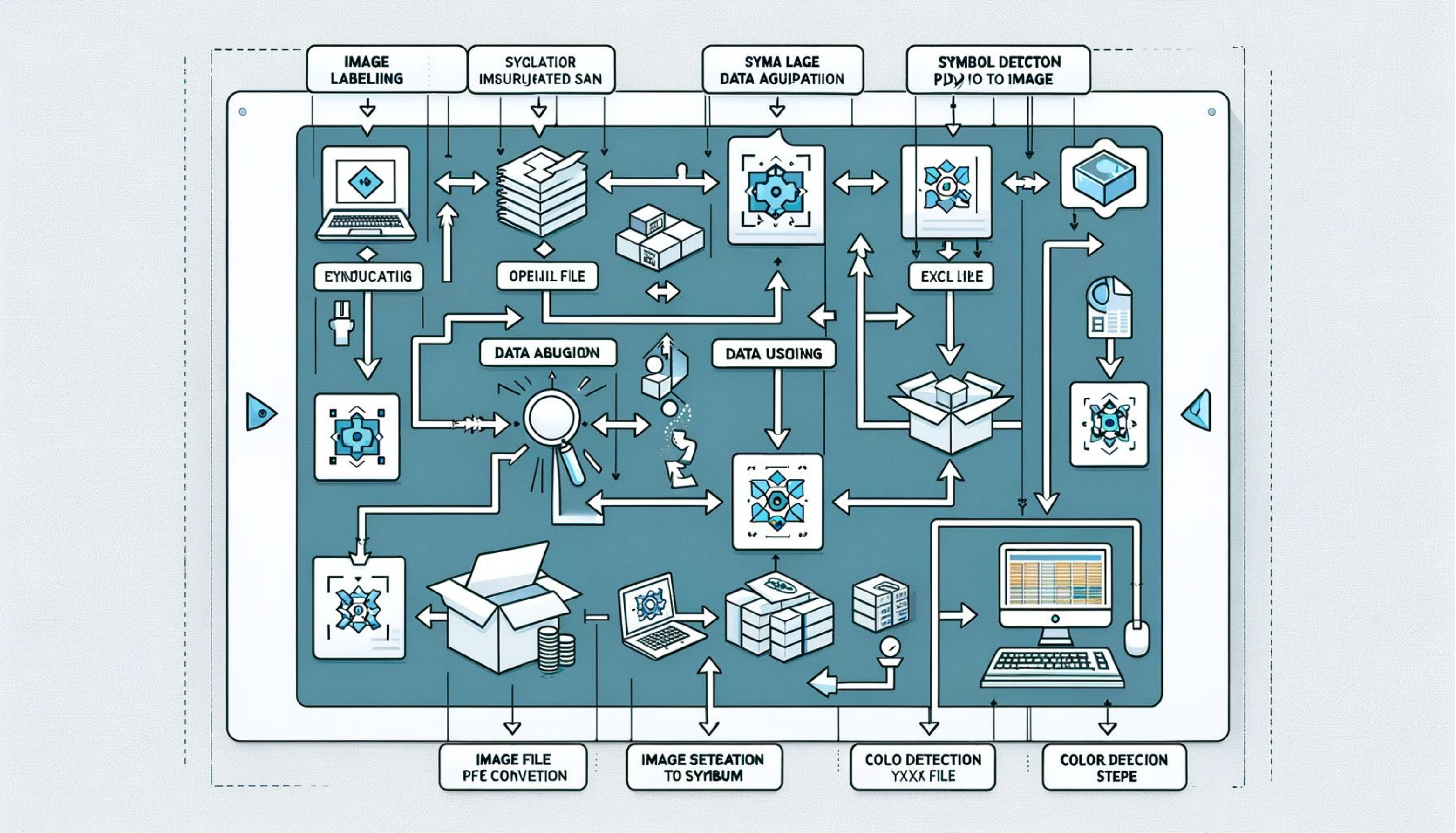
IGOR ADAMENKO

Introduction

- **Objective:** To automate the generation of a Bill of Materials (BOM) from railway route schematics, providing a robust and accurate mechanism to document rail equipment.

- **Approach:** Utilize MATLAB and image processing techniques: YOLOX object detector model training, Non-maximum Suppression, color detection algorithms, and more.

Part 1: Data Preparation and YOLOX Training



Challenges

- **Segmentation Requirement:** YOLOX demands images to be segmented into fixed sizes for training.
- **Manual Labeling:** Labor-intensive process of manually labeling symbols on each segmented image.
- **Scarce Data:** Limited availability of training data poses challenges in effectively training the YOLOX model.
- **Symbol Imbalance:** A disproportionate amount of symbols in the dataset affects the training efficiency and model accuracy.
- **Optimization Dilemma:** Determining the optimal training parameters for YOLOX training.
- **Computational Demand:** The training process is computationally intensive and time-consuming, requiring significant resources.

Process Workflow

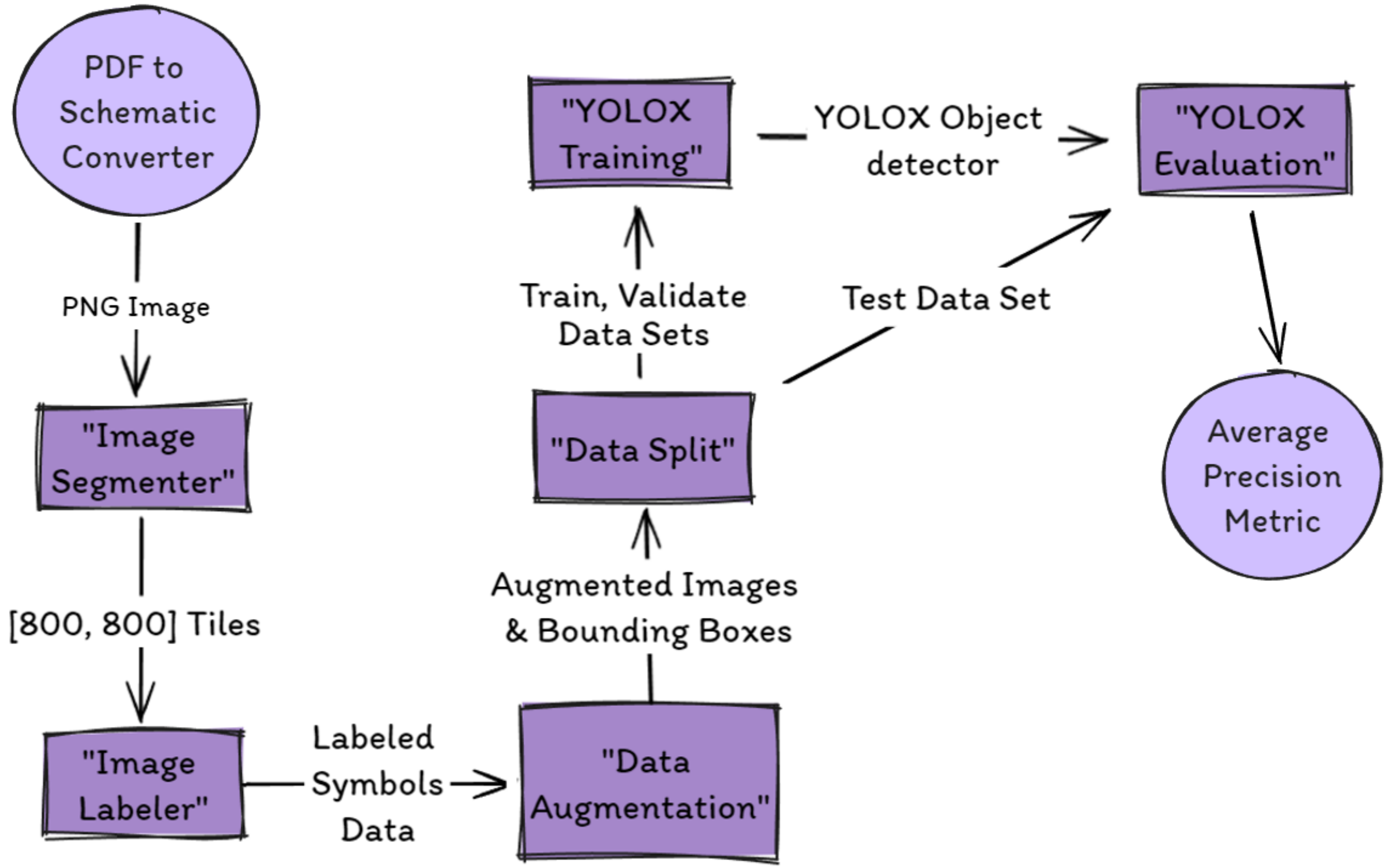
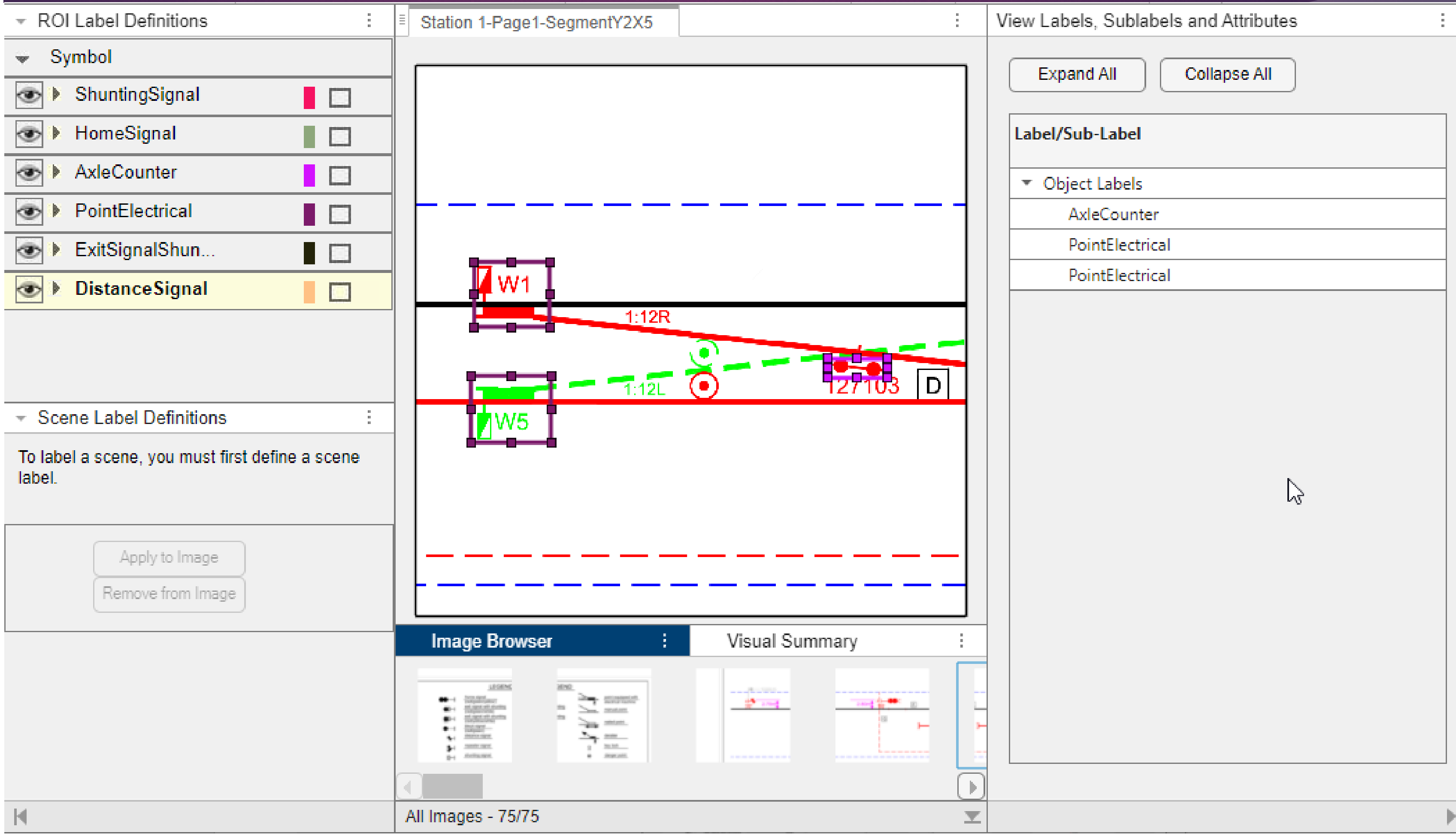


Image Labeler



Data Augmentation

- **Addressing Data Scarcity:** we expanded the dataset by synthetically augmenting existing images through flipping, rotating, and scaling.
- **Technique:** Applied one of the randomly chosen transformation matrices to the original images, diversifying the dataset.
- **Improvement:** These augmentation techniques significantly enhanced the model's detection accuracy, effectively compensating for the initial scarcity of data.

Data Augmentation: Comparison

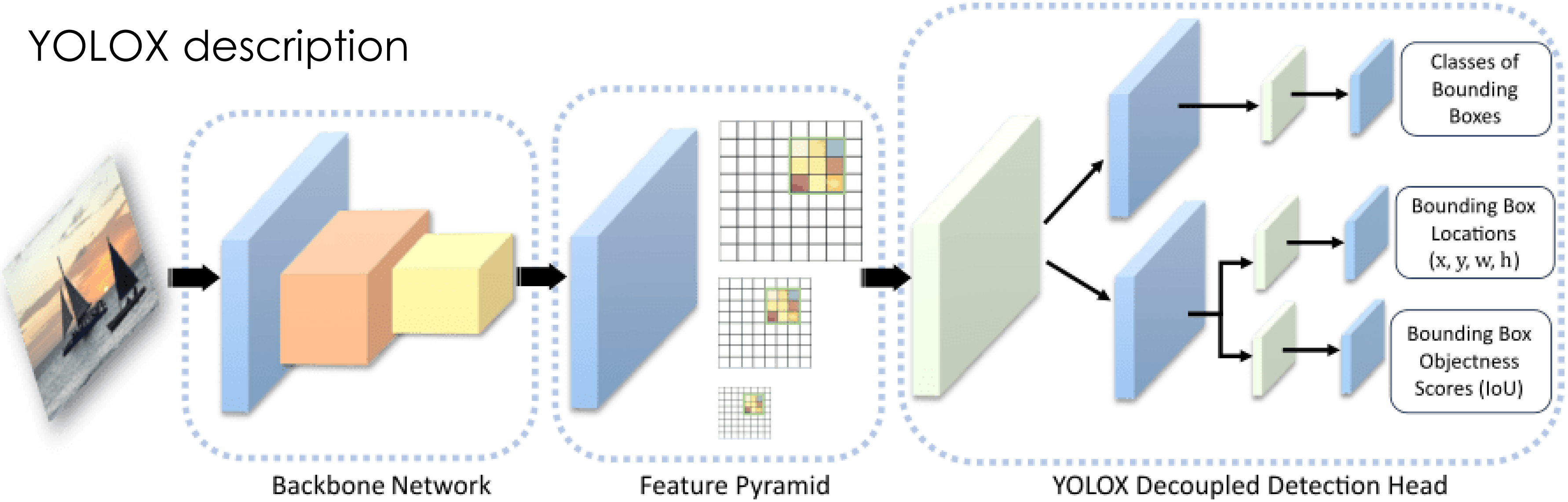
Before augmentation

Label	Count	ImageCount
AxleCounter	48	36
DistanceSignal	29	19
ExitSignalShunting	18	13
HomeSignal	20	14
PointElectrical	36	24
ShuntingSignal	23	16

After augmentation

Label	Count	ImageCount
AxleCounter	548	396
DistanceSignal	308	198
ExitSignalShunting	285	198
HomeSignal	308	198
PointElectrical	429	286
ShuntingSignal	319	209

YOLOX description



- **Input**: It starts with an image.
- **Feature Extraction**: Then, YOLOX uses a part of its network to understand different details in the image by breaking it down into features (like colors, edges, shapes).
- **Grid Overlay**: The network divides the image into grids at different sizes to help locate objects precisely.
- **Processing Layers**: Each piece of the grid goes through layers within the network that make sense of what's in that section, identifying potential objects.
- **Output**: Finally, for each object detected, YOLOX decides what it is (classification), where it is (drawing a bounding box around it), and how sure it is that the object is actually there (confidence).

Training YOLOX

- **Pre-trained Model Utilization**: A pre-trained model is a machine learning model that has been previously trained on a large, general dataset. It learned features such as edges, shapes, and textures.

- **Optimizing Training Options**: Fine-tunes training parameters to achieve optimal performance, balancing between accuracy and generalization.

- **Evaluating Model Performance**: average precision (AP) scores are used to assess and present the model's precision in detecting specific classes.

Training YOLOX: Monitoring Loss Metrics

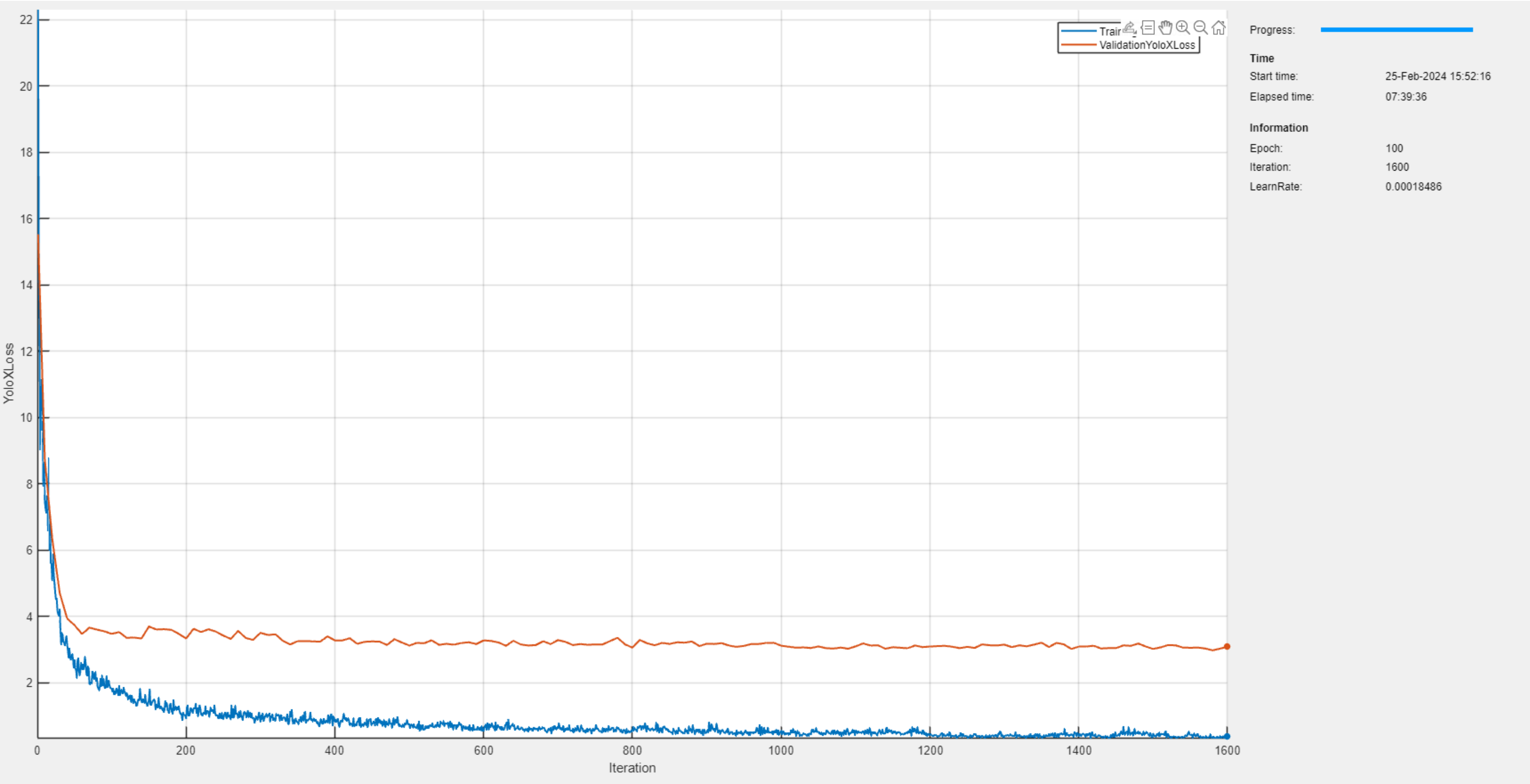
- **TrainingYoloXLoss**: indicates how well the model is learning to detect objects based on the training data. A lower loss value suggests better learning and fitting to the training data.

- **ValidationYoloXLoss**: helps in evaluating the model's generalization ability. A lower validation loss indicates that the model is performing well on unseen data.

Convergence of both losses to similar low values is ideal, indicating that the model is not overfitting (performing well only on the training data) or underfitting (performing poorly on both datasets).

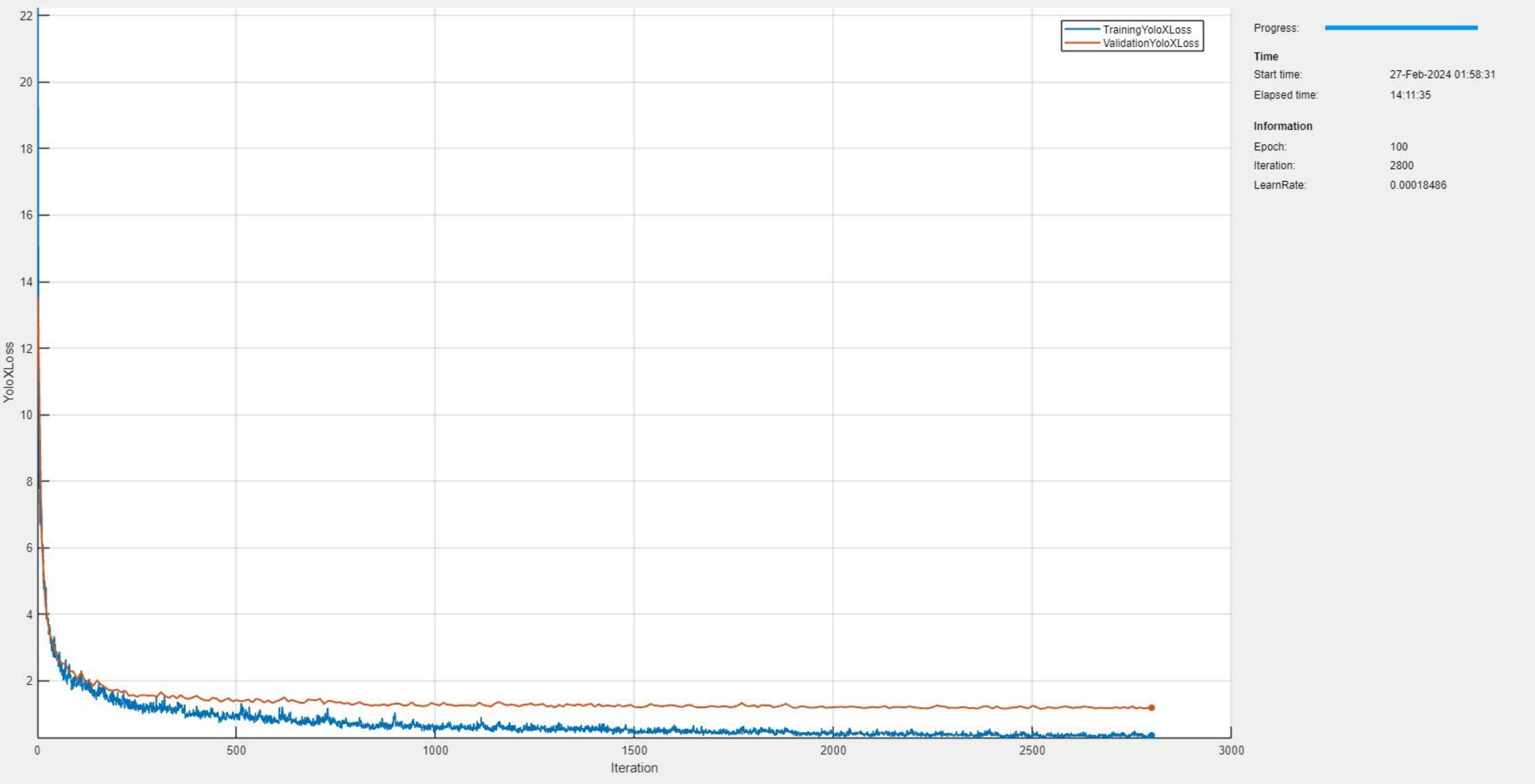
Training YOLOX: Monitoring Loss Metrics

Poorly selected image data set and options



Training YOLOX: Monitoring Loss Metrics

Well-selected image dataset and optimized training options



Training YOLOX: Evaluation

Poor model

classNames	averagePrecision
{'AxleCounter' }	0.85
{'DistanceSignal' }	0.23958
{'ExitSignalShunting'}	0
{'HomeSignal' }	0.38
{'PointElectrical' }	0.66667
{'ShuntingSignal' }	0.57

Better model

classNames	averagePrecision
{'AxleCounter' }	1
{'DistanceSignal' }	0.99841
{'ExitSignalShunting'}	0.95241
{'HomeSignal' }	1
{'PointElectrical' }	0.99983
{'ShuntingSignal' }	1

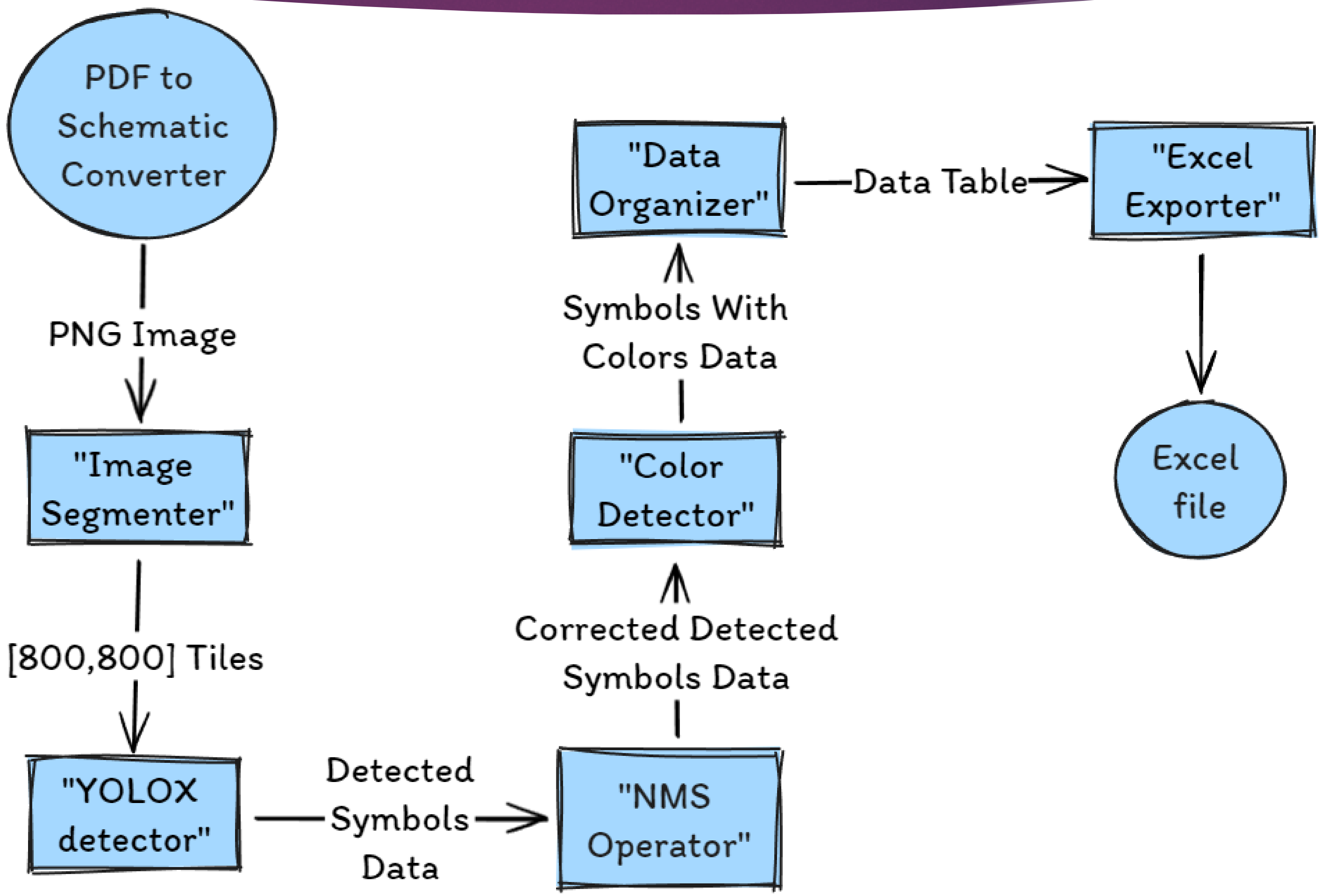
Part 2: Model Application and Further Processing

Challenges

- **Segmented Image Processing:** Images are segmented into tiles for analysis, risking symbol cropping. Overlapping segments are created to mitigate this but introduce the issue of multiple detections for the same symbol.

- **Color Detection:** Employed a mode algorithm for dominant color detection within symbol bounding boxes. Accuracy is compromised when bounding boxes are imprecise or when symbols contain multiple colors, potentially misidentifying the symbol's actual color.

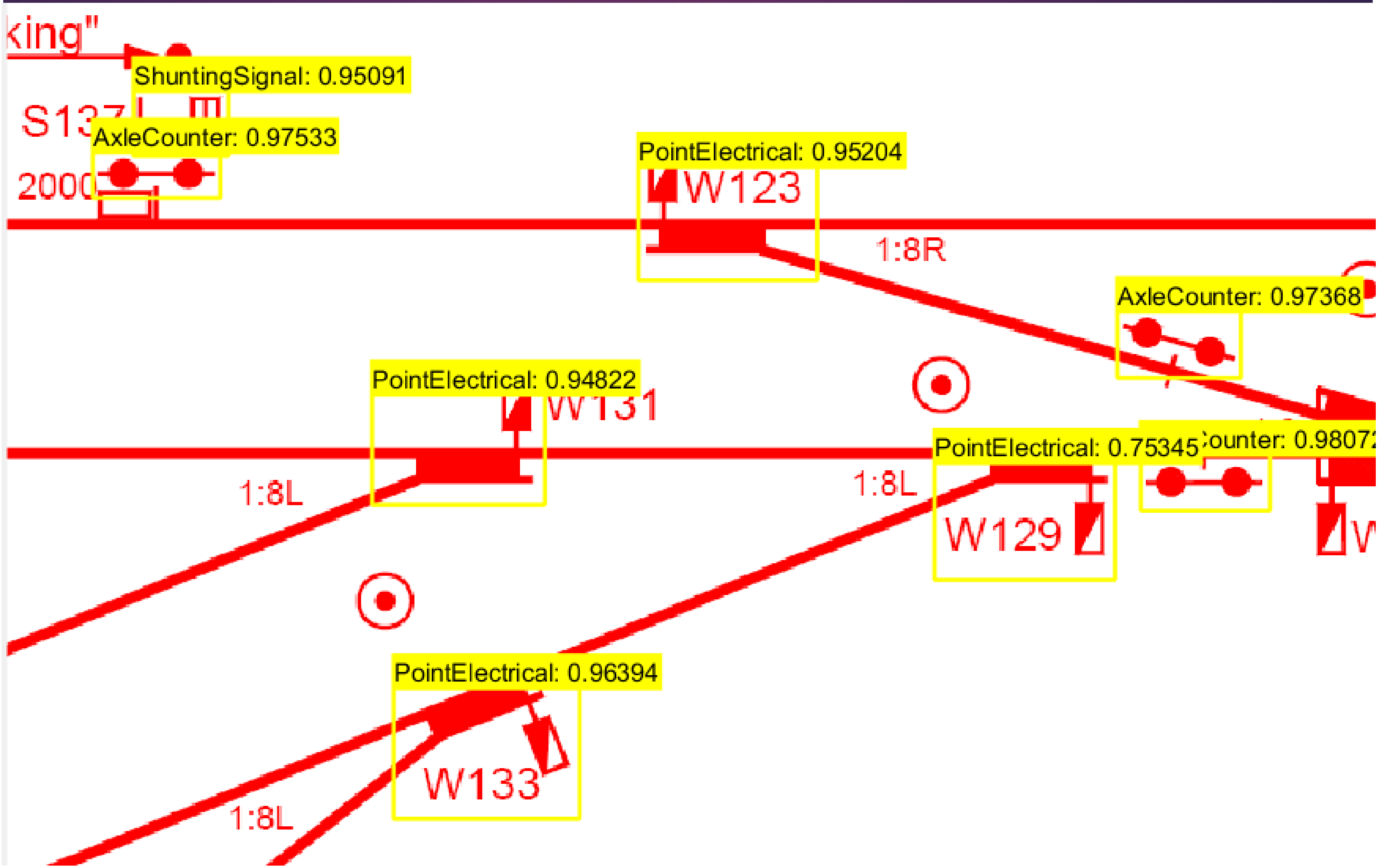
Process Workflow



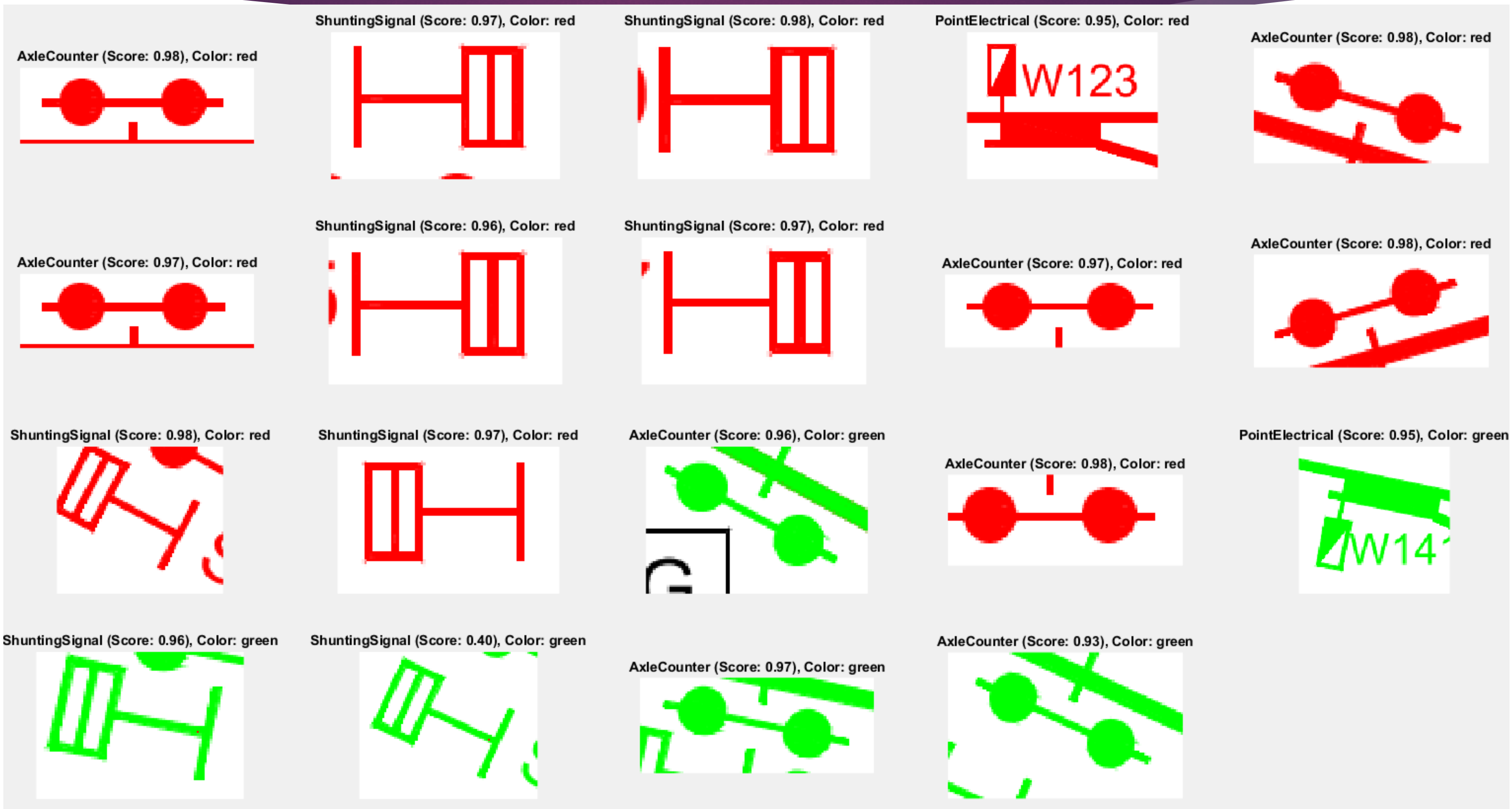
Processing

- **Utilizing NMS:** To address multiple detections from overlapping segments, Non-Maximum Suppression (NMS) is employed, ensuring a single detection per symbol across the full image.
- **Color Identification:** Symbols are individually cropped from the full image to identify the predominant color by comparing against predefined color ranges, adjusting for potential background colors like white and yellow.
- **Exporting to Excel**: We utilize ActiveX to export data into an Excel file, enabling dynamic insertion and resizing of images alongside their corresponding data entries directly within the Excel workbook.

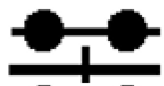
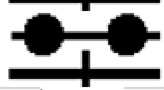
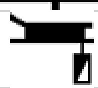
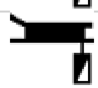
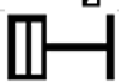
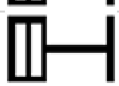
Results and Visualization



Results and Visualization

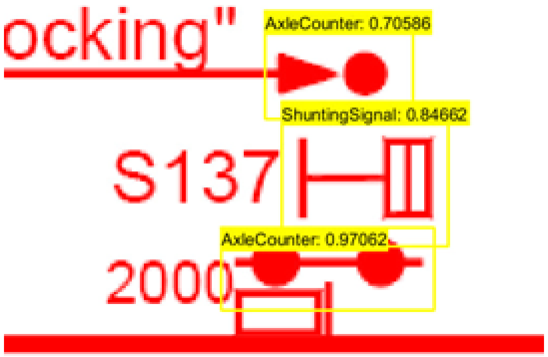


Results and Visualization

	A	B	C	D
1	Symbol	Name	Quantity	Color
2		Axle Counter	38	red
3		Axle Counter	3	green
4		Point Electrical	19	red
5		Point Electrical	1	green
6		Shunting Signal	26	red
7		Shunting Signal	2	green

Issues

- **Detection Inconsistencies:** Symbols occasionally go undetected or are incorrectly identified, affecting the accuracy of the results.
- **Limited Training Scope:** The model was trained primarily on frequently occurring symbols from the legend, excluding less common ones, which limits detection capabilities.
- **Symbol Variability:** The presence of composite symbols and symbols not included in the legend compromises the precision of detection, challenging the model's effectiveness.



Future Work

- **Model Enhancement:** With access to more schematics, the model can be refined to detect a wider variety of symbols, improving its versatility.

- **Accuracy Improvements:** Implementing additional processing techniques, such as correlating detections with known symbols or integrating multiple models for verification, to enhance detection accuracy.

Conclusion

- This project successfully automated the process of generating BOM (bill of materials) from railway schematics using advanced image processing and machine learning techniques.
- Identified issues requiring further resolution.
- Demonstrated machine learning's potential in engineering applications.
- Established a foundation for future advancements in automated documentation.
- Future efforts will focus on improving model versatility and accuracy for enhanced outcomes.