

OCR Research results

Kolobok team

June 2025

1 Introduction

During this week sprint, we were mainly focused on the OCR pipeline (recognizing tire mark, manufacturer, and size). Below is the summary of our work.

2 Research

Our investigation of existing solutions yielded the following sources:

- [Grid Dynamics](#) - a blog post about the implementation of an application with similar to our functionality. Their pipeline consists of detection of the tire ring, then unwrapping it and applying OCR.
- [Roboflow](#) - a blog post with a comparison results of different OCR models on different datasets, including tire marks. [1](#) shows their results.
- [MMOCR](#) - an open-source library by OpenMMLab, which contains a collection of modern OCR pipelines, including text detection and recognition.
- [Tesseract](#) - an open-source end-to-end OCR engine by Google.

Domain	DocTR	Tesseract	Surya	EasyOCR	Gemini	Claude	GPT-4
Hazmat Labels	28.1%	14.4%	29.9%	<u>58.7%</u>	<u>78.1%</u>	87.3%	56.2%
Odometer	0.0%	0.0%	6.0%	54.0%	78.4%	84.6%	13.3%
Tire Serial Numbers	46.4%	8.3%	50.6%	<u>81.0%</u>	86.5%	92.4%	<u>92.2%</u>
Scale Readings	0.0%	0.0%	0.0%	<u>29.8%</u>	23.5%	46.6%	<u>24.8%</u>
Price Tag	<u>98.6%</u>	78.7%	<u>88.8%</u>	<u>92.6%</u>	<u>97.5%</u>	99.3%	0.0%
Book Titles	40.3%	8.2%	28.4%	65.0%	75.0%	100.0%	100.0%
Cargo Container IDs	26.3%	7.1%	37.9%	50.3%	71.5%	74.1%	57.2%
VIN Numbers	58.3%	2.9%	22.3%	<u>86.8%</u>	76.8%	97.6%	<u>78.8%</u>
License Plate Numbers	0.0%	36.5%	50.4%	<u>79.5%</u>	85.0%	95.0%	20.0%
Named Logos	21.3%	17.1%	18.0%	<u>80.0%</u>	<u>83.2%</u>	100.0%	100.0%

Figure 1: Comparison of different OCR models on different datasets, including tire marks by Roboflow blog post

3 Experiments

In order to evaluate existing approaches, we collected a small dataset of 15 images with side view of tires from parking near the 6th dormitory. Then we evaluated the accuracy of different models and different image preprocessing approaches. Our conducted experiments include the following DL pipelines:

- **Tesseract**
- **DBNet++ + ABINet** (MMOCR)
- **PSENet + ABINet** (MMOCR)
- **TextSnake + ABINet** (MMOCR)
- **PANet + ABINet** (MMOCR)
- **GPT-4o-mini**

In MMOCR models, we use a combination of two models: text detector (first) and text recognition (second). We evaluated the pipeline with varying text detection models, while preserving the same recognition model. We argue that our pipeline quality is more dependent on the detection model (for example, TextSnake is better at detecting curved text, while DBNet++ is better at finding bounding boxes of text), while recognition models from MMOCR model zoo show similar performance on same datasets. Therefore, to get an accurate comparison results, we keep the recognition model constant for all experiments (Note: ABINet is the latest model in MMOCR repository).

We also evaluated proposed algorithms on different preprocessing approaches:

- **Raw** - no preprocessing
- **Polar unwrapping** - unwrapping the tire image to a flat view (see [2](#))
- **CLAHE** - contrast limited adaptive histogram equalization (see [3](#))



Figure 2: Unwrapped tire image example



Figure 3: CLAHE-enhanced tire image example

4 Evaluation details

We conduct an evaluation session, where the image is first preprocessed, then passed through the OCR pipeline, and finally the output is compared with the ground truth.

Evaluation metric: the number of correctly recognized texts. We have 3 labels for each image: tire mark, manufacturer, and size, giving 45 text labels in total. For each pipeline, we evaluate the number of correctly recognized texts.

For Tesseract and MMOCR models, we do not use any postprocessing of output, which gives an advantage to GPT-4o-mini, which can apply its knowledge of the language to get more accurate results. To compensate for this bias, we evaluate those models in the following way:

- All of the detected texts are compared with all GT labels
- If the detected text has a levenstein ratio with any GT label more than 0.75, it is considered as correctly recognized

This approach gives more tolerance to the OCR models without language knowledge, as spelling mistakes can easily be corrected later.

GPT-4o-mini is evaluated more strictly:

- The model is instructed to output `JSON` data with 3 fields
- Each field is compared with GT label, and considered correct only if there is a perfect match

5 Results

Model	Raw	Polar Unwrapping	CLAHE
Tesseract	4	9	8
DBNet++ + ABINet	6	10	15
PSENet + ABINet	5	11	14
TextSnake + ABINet	7	12	16
PANet + ABINet	3	8	9
GPT-4o-mini	37	45	45

6 Conclusion

As our experiments have shown, the most accurate approach is to use GPT-4o-mini with unwrapping preprocessing with optional CLAHE. We may consider using an approach less dependent on external APIs, but it would require proper tuning of the OCR pipelines, which is not a strict requirement, as GPT-4o-mini API cost is very low.