Evaluating OCR Performance on Ingredient Lists and Nutrition Panels from South African Food Packaging

Submission 49

No Institute Given

Abstract. This study evaluates four open-source OCR systems which are Tesseract, EasyOCR, PaddleOCR, and TrOCR on real-world food packaging images. The aim is to assess their ability to extract ingredient lists and nutrition facts panels. Accurate OCR for packaging is important for compliance and nutrition monitoring but is challenging due to multilingual text, dense layouts, and varied fonts. A dataset of 231 products (1,628 images) was processed by all four models to assess speed and coverage. From this, 60 products (113 images) were manually transcribed to create ground truth, which was used for accuracy evaluation. Metrics included Character Error Rate (CER), Word Error Rate (WER), BLEU, ROUGE-L, F1, coverage, and execution time. Tesseract achieved the best accuracy (CER = 0.912, BLEU = 0.245). EasyOCR provided a good balance between accuracy and multilingual support. PaddleOCR offered a near-complete coverage but was significantly slower due to CPU execution, caused by the model's GPU incompatibility, and TrOCR produced the weakest results despite GPU acceleration. These results provide a packaging-specific OCR benchmark, establish a baseline for future OCR research on packaging images, and highlight directions for improving recognition through layout-aware models and computer-vision-based text localization.

Keywords: Optical Character Recognition · Food Packaging · Ingredient Lists · Nutrition Facts Panels · Tesseract · EasyOCR · PaddleOCR · TrOCR.

1 Introduction

Consumers rely on printed information on food packaging such as ingredient lists, allergen warnings, and nutrition facts to make informed dietary decisions. While regulations require this information to be accurate and legible, packaging often reduces readability with cluttered layouts, small fonts, decorative text styles, and multilingual content for export. These design choices make it harder for people to read and create additional challenges for digital processing. Child-directed marketing practices, which often include illustrations and fantasy imagery, further reduce legibility, particularly in breakfast cereals and snacks [1]. For example, multilingual text and inconsistent formatting complicate both manual review and automated extraction, limiting the accuracy of nutrition apps, compliance checks, and large-scale health research [2,3]. This context highlights the need for robust OCR systems that can handle real-world packaging conditions rather than controlled document layouts [4].

Extracting printed text from food packaging is challenging because of the way information is presented. Ingredient lists and nutrition facts often use small fonts, column layouts, and multiple languages within a limited space. These conditions reduce recognition accuracy for OCR systems. Real-world images add complexity through glare, shadows, and curved or reflective surfaces that distort text [5]. Studies highlight that such design and imaging factors make packaging more complex than standard documents with uniform structures [2]. Seitaj and Elangovan [3] further note that mixed text styles and multilingual content frequently cause segmentation and recognition errors. Huang et al. [6] emphasize that many OCR models fail when text includes language switching within the same block, an issue common in packaging.

Although OCR systems perform well on scanned documents and scene-text benchmarks, their reliability on food packaging remains underexplored. Public datasets and competitions such as the International Conference on Document Analysis and Recognition (ICDAR) focus on either documents or natural scenes, which lack the dense tabular structures and multilingual content typical of packaging [7]. Large-scale benchmarks like TextOCR emphasize arbitrarily shaped text in outdoor images but do not include nutrition panels or ingredient lists [8]. Recent advances such as multilingual OCR networks and layout-aware frameworks [6,9] aim to improve recognition across scripts and structured regions, but these methods are rarely tested on packaging with language switching and fine-grained text. Existing packaging studies, including those by Guimarães et al. [2] and Rosyadi et al. [10], highlight the complexity of cluttered layouts and mixed fonts but rely on small datasets or single-engine evaluations. These limitations make it difficult to establish fair comparisons or draw conclusions about real-world performance, creating a clear need for systematic benchmarking under uncontrolled imaging conditions.

This paper addresses these gaps by presenting a comparative evaluation of four widely used open-source OCR systems: Tesseract, EasyOCR, PaddleOCR, and TrOCR on real-world food packaging images captured in South African retail environments. The study focuses on ingredient lists and nutrition facts panels using full packaging images and applies a standardized post-processing pipeline for normalization. Performance is assessed using character-level, word-level, and semantic metrics alongside coverage and execution time, establishing a domain-specific benchmark for packaging OCR.

This paper advances packaging OCR research in several ways. First, it evaluates Tesseract, EasyOCR, PaddleOCR, and TrOCR on real-world food packaging, focusing on ingredient lists and nutrition facts panels. Second, it introduces a benchmark that uses multiple metrics, including CER, WER, BLEU, ROUGE-L, F1, coverage, and execution time. Third, it applies a standardized post-processing workflow to normalize outputs across models, and finally, it offers insights into performance trade-offs and limitations under practical deployment conditions.

The remainder of the paper is organized as follows: Section 2 provides an overview of the OCR systems evaluated in this study. Section 3 presents the methodology, including dataset description, preprocessing, and experimental setup. Section 4 outlines the evaluation metrics and implementation details, while Section 5 reports and discusses the results, and finally, Section 6 concludes the paper and highlights potential directions for future research.

2 OCR Approaches

2.1 Tesseract OCR

Tesseract is one of the most widely used open-source OCR engines, initially developed by Hewlett-Packard and later maintained by Google. Early versions relied on rule-based segmentation, which restricted flexibility for non-standard layouts. Since version 4.0, Tesseract incorporates an LSTM-based recognition module, allowing sequence learning and improving accuracy on printed text compared to its earlier design [11]. This architecture is effective on clean, structured text, making Tesseract a strong baseline for document-oriented OCR tasks.

Despite these advances, Tesseract remains highly sensitive to noisy backgrounds, small fonts, and curved text regions. Its reliance on line segmentation causes recognition errors on dense tabular structures, such as nutrition facts panels, and on multilingual ingredient lists with mixed fonts and orientations [12]. Preprocessing steps such as grayscale conversion, contrast normalization, and denoising have been shown to reduce errors, yet challenges persist on real-world packaging images that combine decorative elements with text [2].

Tesseract has been tested in food-label recognition tasks. Saputra et al. [13] combined Tesseract with preprocessing and detection for nutrition-label extraction, reporting improved accuracy only after structured segmentation. These findings suggest that while Tesseract performs well on standardized layouts, it requires additional steps to maintain performance under uncontrolled packaging conditions.

2.2 EasyOCR

EasyOCR is an open-source OCR library implemented in Python and developed by JaidedAI, supporting over 80 languages [14]. Its architecture combines convolutional layers for feature extraction and bidirectional recurrent layers for sequence modeling in a CRNN-based recognition network. When detection is enabled, EasyOCR uses the CRAFT detector, although recognition can run independently for pre-cropped or full images.

The main strength of EasyOCR lies in its multilingual support and lightweight deployment, which enables fast inference on GPU-based systems. These properties make it attractive for scenarios involving multi-language content, such as food packaging [15]. However, studies have reported that EasyOCR struggles with small fonts, cluttered backgrounds, and irregular layouts that combine multiple languages in dense panels [2]. Even with multilingual capability, the absence of layout-specific optimizations limits its robustness on packaging images compared to controlled document text.

Prior evaluations show EasyOCR performing competitively on scene-text benchmarks but less consistently on structured domains. Flores et al. [15] confirmed that EasyOCR is more resilient than Tesseract under image distortions but still degrades when handling text mixed with graphical elements, a common condition in retail packaging.

2.3 PaddleOCR

PaddleOCR, developed by Baidu, is a modular OCR system designed for multilingual text and layout-aware recognition. It integrates DBNet-based detection with CNN-RNN recognition heads, while recent versions such as PP-OCRv5 incorporate lightweight models optimized for mobile and real-time inference. PaddleOCR also provides optional modules for document parsing, including PP-Structure, which can segment tables and key-value pairs [16,17].

These capabilities make PaddleOCR theoretically suitable for packaging tasks that include structured layouts like nutrition tables. Rosyadi et al. [10] demonstrated its practical use for ingredient-list extraction from smartphone images, reporting competitive accuracy but highlighting performance degradation in poor lighting and on reflective surfaces. While PaddleOCR supports angle classification for skew correction and multiple language packs, its effectiveness on unsegmented packaging images remains uncertain, as most prior work assumes either document-level or cropped-region inputs.

The inclusion of PaddleOCR in this study enables evaluation of a state-of-the-art modular system on full packaging images. This approach tests whether its advanced features for layout handling and multilingual recognition can compensate for the absence of explicit text-region detection in real-world conditions.

2.4 TrOCR

TrOCR is a transformer-based OCR model introduced by Microsoft, combining a Vision Transformer (ViT) encoder with an autoregressive language decoder inspired by GPT-2. This architecture formulates OCR as a sequence-to-sequence task, eliminating handcrafted segmentation and allowing the model to learn contextual relationships across an image [18]. TrOCR has demonstrated strong results on document and handwriting benchmarks, benefiting from large-scale pretraining and fine-tuning.

Despite its advantages, TrOCR lacks explicit layout modeling, which poses challenges for structured packaging images containing dense text blocks, small fonts, and multilingual content. Studies note that transformer-based OCR systems often exhibit output fragmentation when applied to cluttered or tabular layouts without preprocessing [19,20]. Line-level segmentation or slicing is often necessary to prevent recognition collapse in complex layouts.

This evaluation includes TrOCR to assess whether its transformer-based design provides measurable gains on real-world packaging or whether its reliance on full-image encoding limits accuracy without domain-specific tuning. Findings from prior work suggest significant trade-offs: while transformers excel at contextual reasoning, they underperform in environments where layout and structure dominate recognition complexity [19].

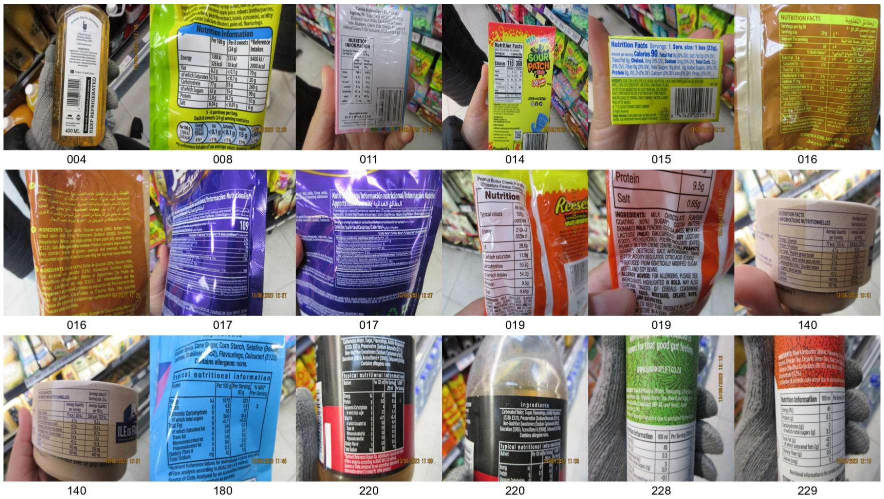

Fig. 1. Sample food packaging images from the SA NFP 2023 dataset.

3 Methodology

3.1 Dataset Description

The dataset used in this study was collected in 2023 by the Department of Nutrition and Dietetics at the University of the Western Cape as part of the South African Nutrition Facts Panel (NFP) project. Images were captured in retail environments using handheld mobile devices, introducing challenges such as glare, reflections, and curved surfaces that affect OCR accuracy. The full dataset folder selected for this work, *20230613_04_03*, contains 231 products and 1,628 images.

Each product in this folder includes multiple views (2–23) showing different sides of the packaging, such as the front panel, ingredient list, and NFP. Images follow a structured naming format that encodes the capture date, session ID, fieldworker ID, product ID, and image index as *YYYYMMDD_ XX_ YY_ ZZZ*

(N).jpg. In this format, *YYYYMMDD* indicates the date on which the image was captured. The *XX* component represents the session or batch identifier, while *YY* corresponds to the ID of the fieldworker who collected the data. The *ZZZ* portion is a unique product identifier, such as 001 or 002, and the *(N)* at the end denotes the image index, used to differentiate between various views of the same product packaging. Figure 1 illustrates examples of the packaging images used in this study with product numbers under each product, highlighting diverse conditions such as glare, curved surfaces, and multilingual ingredient lists, which reflect the real-world complexity of the evaluation dataset.

All 1,628 images were processed by the four OCR systems to measure execution time and coverage. For accuracy evaluation, a subset of 60 products (113 images) was manually transcribed to create ground truth for ingredient lists and NFP text. This design reflects practical deployment scenarios where processing speed is critical at scale, while detailed error metrics can only be computed for annotated data. The dataset differs significantly from available OCR benchmarks due to how packaging combines multilingual text, irregular layouts, and small fonts with decorative elements, making recognition more difficult than on documents or signage [2,21]. This reflects broader trends in the South African packaged food supply, where visual clutter and on-package marketing often compromise the clarity of mandatory labeling [1]. This complexity makes this dataset well-suited for benchmarking OCR models under realistic deployment conditions.

3.2 OCR Models Used

Four open-source OCR systems were evaluated in this study: Tesseract, EasyOCR, PaddleOCR, and TrOCR. These models represent distinct architectural paradigms—LSTM-based (Tesseract), CNN–RNN pipelines (EasyOCR and PaddleOCR), and transformer-based recognition (TrOCR). All systems were applied directly to full packaging images without prior region segmentation or keyword filtering to reflect realistic deployment conditions. This approach tests each model's ability to handle common packaging challenges, such as multilingual text, small fonts, irregular layouts, and dense tabular structures.

3.3 Implementation and Experimental Setup

The experiments were conducted on a high-performance workstation running *Windows 11 Enterprise (64-bit)* with an *Intel 64-based processor* (24 cores, 32 threads) operating at 3.2 GHz, 128 GB RAM, and an *NVIDIA GeForce RTX 3070 GPU* with 8 GB of VRAM. All OCR systems were implemented in *Python 3.11.9*. Key library versions were: *pytesseract 0.3.13* for Tesseract, *EasyOCR 1.7.2*, *PaddleOCR 2.7.0.3*, and *HuggingFace Transformers 4.53.3* for TrOCR. GPU acceleration was enabled for EasyOCR and TrOCR, while PaddleOCR ran on CPU due to backend compatibility limitations. Tesseract, by design, operates only on CPU [11,12].

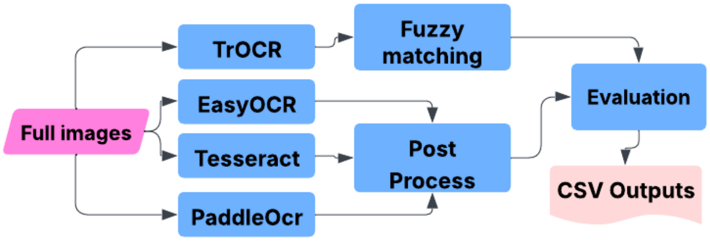

Fig. 2. OCR evaluation workflow from image input to Evaluations.

All four OCR systems were applied to full packaging images without region cropping or segmentation. Preprocessing for Tesseract included grayscale conversion, contrast enhancement, and non-local means denoising to enhance text clarity on cluttered layouts. EasyOCR was applied without additional custom steps beyond its internal resizing and normalization routines. PaddleOCR relied on its default PP-OCRv5 pipeline, which includes resizing, normalization, and angle classification for skew correction. TrOCR required RGB normalization because its Vision Transformer (ViT) encoder operates on three-channel inputs, and horizontal line-based slicing was applied before inference on dense areas such as nutrition facts tables to improve recognition accuracy compared to full-image inference [18].

Each system was executed with default models to avoid bias from fine-tuning. Tesseract employed its LSTM-based engine; EasyOCR used its CRNN recognizer with multilingual support; PaddleOCR ran with the lightweight PP-OCRv5 recognition model; and TrOCR utilized the pre-trained `microsoft/trocr-base-printed` configuration. Outputs were logged in CSV format with fields for `product_id`, `image_filename`, `raw_ocr_text`, and processing time. Figure 2 provides an overview of the OCR evaluation pipeline, showing how full packaging images were processed through the four OCR systems, followed by normalization and metric-based evaluation. JSON outputs were also generated during initial testing for grouping checks but were excluded from final evaluation to maintain consistency across models.

3.4 Data Post-Processing and Normalization

Raw OCR outputs were standardized to ensure comparability across models before evaluation. Predictions were initially stored in CSV format with fields such as `product_id`, `image_filename`, and `raw_ocr_text`, and then processed through a two-stage normalization pipeline.

For **Tesseract**, **EasyOCR**, and **PaddleOCR**, a keyword-based classification strategy was used to assign text to one of two categories, which are either *ingredients* or *nutrition facts panel (NFP)*. When multiple candidates existed for the same product and text type, keyword density served as the tie-breaker to retain the most relevant block. This approach reflects common practices for handling structured text in OCR workflows where specific sections must be isolated for downstream tasks [2,10].

TrOCR required a different approach because its transformer-based encoder–decoder architecture often produced fragmented predictions on dense packaging layouts. A fuzzy-matching method based on partial ratio scores was applied to identify text segments most strongly associated with target keywords. A variant of the keyword-density pipeline was also tested but kept producing incomplete matches, confirming that fuzzy matching was more effective for this model.

After classification, the following normalization steps were applied across all systems:

- Removal of special characters and redundant whitespace.
- Extraction of text following the first occurrence of relevant keywords (e.g., "ingredients:" or "nutrition information").
- Conversion to lowercase for uniform string comparison.

The normalized outputs were consolidated into standardized CSV files containing four fields: `product_id`, `image_filename`, `text_type`, and `ocr_text`. These files served as the input for all evaluation metrics described in Section 4.

4 Evaluation Setup

This section describes the metrics and procedures used to evaluate the OCR systems. Five complementary metrics were selected to capture both character-level and semantic accuracy: Character Error Rate (CER), Word Error Rate (WER), BLEU, ROUGE-L, and F1 score. These measures are widely used in OCR benchmarking, where CER and WER dominate structured-text evaluations [7,22]. In addition, processing time was recorded for all models to assess computational efficiency.

4.1 Character Error Rate (CER)

CER quantifies character-level accuracy by computing the Levenshtein distance between the predicted and reference text, normalized by the length of the ground truth [7,22]. It penalizes insertions, deletions, and substitutions equally. CER is defined as:

$$\mathrm{CER} = \frac{S + D + I}{N},$$

where S represents substitutions, D deletions, I insertions, and N the number of characters in the reference string.

4.2 Word Error Rate (WER)

WER applies the same principle as CER but operates at the word level, reflecting the preservation of semantic units in OCR output. It is particularly relevant for ingredient lists and nutrition facts panels [7]. WER is defined as:

$$\text{WER} = \frac{S + D + I}{N_w},$$

where N_w denotes the number of words in the reference text.

4.3 BLEU (Bilingual Evaluation Understudy)

BLEU measures n-gram overlap between predicted and reference text. Although originally introduced for machine translation, BLEU is widely applied in OCR research for evaluating structural fidelity [23]. The metric is calculated as:

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{k} w_n \log p_n\right),$$

where p_n is the modified n-gram precision, w_n the weight for each n-gram size, and BP the brevity penalty.

4.4 ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation)

ROUGE-L evaluates text similarity using the longest common subsequence (LCS), combining precision and recall to account for sequence order. This property makes ROUGE-L effective for structured text such as ingredient lists [24]. It is defined as:

$$\text{ROUGE-L}_{F_1} = \frac{(1 + \beta^2) \cdot R \cdot P}{R + \beta^2 P},$$

where R denotes recall, P precision, and β is typically set to 1 for equal weighting.

4.5 F1 Score

The F1 score, calculated at the word level, represents the harmonic mean of precision and recall, emphasizing exact word matches. It complements BLEU and ROUGE by focusing on strict token overlap, which is important for structured fields such as nutrition panels [7]. The formula is:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

4.6 Implementation Details

All metrics were implemented in Python 3.11.9 using the following libraries: `python-Levenshtein` for CER and WER, `nltk.translate.bleu_score` for BLEU (with smoothing method 4), `rouge_score` for ROUGE-L, and a custom implementation for F1 at the word level. Normalization steps prior to evaluation included keyword-based classification for Tesseract, EasyOCR, and PaddleOCR, fuzzy matching for TrOCR, removal of punctuation, case folding, and trimming redundant characters (see Section 3.4).

Accuracy metrics were computed using 113 manually transcribed images from 60 products, representing the ground truth subset. Processing time was measured across all 1,628 images to reflect performance under realistic workload conditions. This separation ensures accurate quality assessment without compromising fairness in efficiency analysis.

Despite the availability of newer neural-based similarity measures, BLEU and ROUGE-L remain widely adopted in OCR evaluation due to their interpretability and established benchmarks, building on their original definitions in [23,24].

5 Results and Discussion

This section presents the evaluation results for the four OCR systems: EasyOCR, Tesseract, PaddleOCR, and TrOCR. Performance was evaluated using character-level metrics (CER and WER), semantic metrics (BLEU, ROUGE-L, and F1), coverage, and execution time. The findings are interpreted in relation to dataset characteristics and model architectures to assess their suitability for food packaging text extraction in real-world conditions.

5.1 Overall Performance Summary

The evaluation used 60 products with 113 images, covering both ingredient lists and nutrition facts panels. These samples were selected from the full set of 231 products to maintain diversity while keeping the evaluation manageable. Table 1 presents the mean BLEU, ROUGE-L, F1 scores, and coverage for each OCR system.

Table 1. Overall semantic performance of OCR models. Best values in each column are in **bold**.

Model	**Coverage (%)**	**BLEU**	**ROUGE-L**	**F1**
EasyOCR	91.53	0.153	0.314	0.265
Tesseract	79.66	**0.245**	**0.391**	**0.345**
PaddleOCR	98.31	0.163	0.361	0.248
TrOCR	**100.00**	0.010	0.026	0.017

Tesseract achieved the highest BLEU, ROUGE-L, and F1 scores, showing the strongest semantic accuracy among all systems. PaddleOCR achieved the highest coverage among the convolution-based models and delivered competitive ROUGE-L performance. EasyOCR provided balanced results across all metrics, while TrOCR produced the lowest semantic scores despite full coverage, highlighting its limitations when handling dense layouts and multilingual content.

5.2 CER and WER Analysis

Character Error Rate (CER) and Word Error Rate (WER) evaluate recognition accuracy at the character and word levels. Table 2 summarizes the mean CER and WER values for all models, while Figure 3 and Figure 4 illustrate their distribution.

Table 2. CER and WER summary statistics per OCR model

Model	**CER Mean**	**WER Mean**
EasyOCR	1.075	7.408
Tesseract	**0.912**	**6.262**
PaddleOCR	0.985	6.857
TrOCR	1.049	7.243

Tesseract achieved the lowest mean CER (0.912) and WER (6.262), indicating strong recognition accuracy. It also displayed the least variability, as shown in Figure 3, which suggests consistent performance across different image conditions. TrOCR showed the widest error range, with WER values reaching 38.4, reflecting sensitivity to complex layouts. EasyOCR and PaddleOCR produced intermediate error rates with occasional outliers, often linked to small fonts and noisy backgrounds.

5.3 BLEU, ROUGE-L, and F1 Analysis

BLEU and ROUGE-L measure n-gram and sequence-based similarity, while F1 evaluates precision and recall at the token level. Figure 5 compares BLEU and ROUGE-L scores, and Figure 6 shows F1 performance.

Tesseract achieved the highest BLEU and ROUGE-L scores (0.245 and 0.391), confirming its advantage in preserving lexical and structural accuracy. EasyOCR and PaddleOCR delivered moderate scores, while TrOCR recorded very low BLEU (0.010) and ROUGE-L (0.026) values, indicating limited capability for structured text on packaging. These outcomes suggest that convolution-based models are currently more reliable than transformer-based architectures when applied to complex layouts without fine-tuning.

Although new semantic similarity metrics have emerged, BLEU and ROUGE remain standard for OCR text fidelity evaluation because of their interpretability and benchmarking history, as established by Papineni et al. [23] and Lin [24].

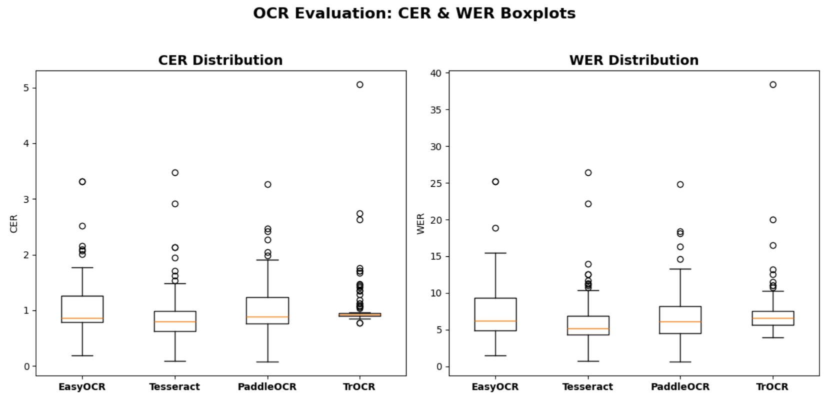

Fig. 3. Distribution of CER and WER across OCR models.

5.4 Coverage Analysis

Coverage refers to the percentage of products for which an OCR model successfully extracted text for both ingredient lists and nutrition facts panels. TrOCR achieved full coverage at 100%, followed by PaddleOCR at 98.31%, EasyOCR at 91.53%, and Tesseract at 79.66%. The complete coverage recorded by TrOCR indicates that the model consistently returned text outputs, although these outputs often lacked semantic accuracy. In contrast, the lower coverage observed for Tesseract is associated with its sensitivity to image noise and stylized fonts, which occasionally resulted in empty outputs.

5.5 Execution Time Comparison

Average processing times for each model are shown in Table 3. Tesseract was the fastest system, with an average of 0.58 seconds per image, making it well-suited for large-scale deployments on CPU-based environments. EasyOCR followed with an average time of 0.81 seconds per image, benefiting from GPU acceleration on the NVIDIA RTX 3070. PaddleOCR was the slowest, averaging 6.24 seconds per image due to its execution in CPU-only mode, which was required because of PaddlePaddle GPU compatibility limitations. TrOCR averaged 2.20 seconds per image despite running on GPU, reflecting the higher computational cost of its Transformer-based architecture.

5.6 Discussion

The evaluation shows that Tesseract achieved the highest overall accuracy across character-level metrics (CER and WER) and semantic measures (BLEU, ROUGE-L, and F1). Its LSTM-based recognition pipeline appears well-suited for structured text such as ingredient lists and nutrition panels. This outcome aligns

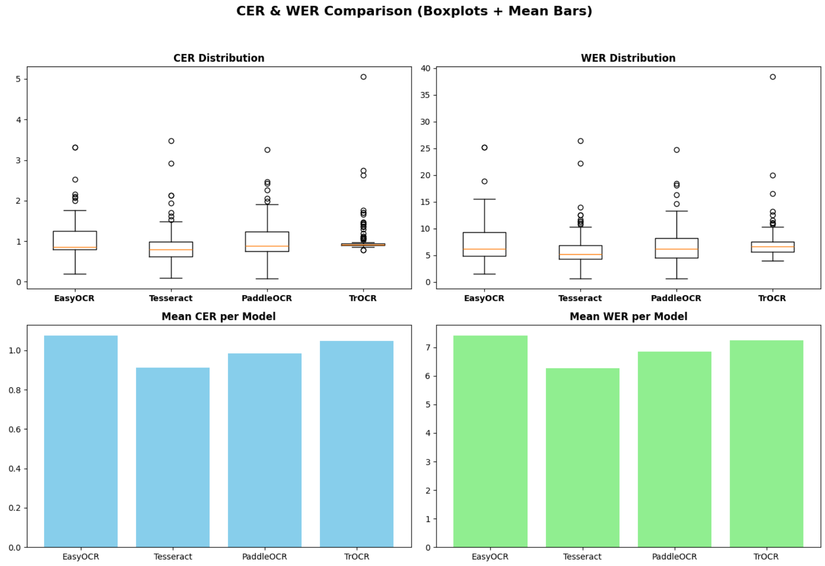

Fig. 4. CER and WER results shown with boxplots and mean values for each model.

Table 3. Average execution time per image and hardware utilization.

Model	**Total Time (s)**	**Avg Time/Image (s)**	**Hardware Used**
EasyOCR	1,322.10	0.81	GPU (RTX 3070)
Tesseract	**949.53**	**0.58**	CPU Only
PaddleOCR	10,161.16	6.24	CPU Only
TrOCR	3,573.58	2.20	GPU (RTX 3070)

with the findings of Smith [11], who reported that Tesseract remains effective on printed text when combined with basic preprocessing.

EasyOCR and PaddleOCR produced competitive results. PaddleOCR obtained the highest coverage but had the longest processing times because it ran exclusively on CPU during these experiments. Although the system supports GPU acceleration, compatibility issues with CUDA prevented its use in this setup, which is a known limitation documented in practice [10]. This constraint, combined with its multi-stage architecture, contributed to slower execution. EasyOCR offered a more balanced trade-off between speed and accuracy, particularly in GPU-enabled environments, making it suitable for multilingual packaging scenarios [14].

TrOCR recorded the weakest results despite GPU acceleration and a transformer-based architecture. Similar limitations were noted by Li et al. [18], who emphasized that transformer-based OCR models often require domain-specific fine-tuning to manage irregular layouts and dense structures effectively. Tesseract's advantage over TrOCR can be attributed to its optimization for printed text

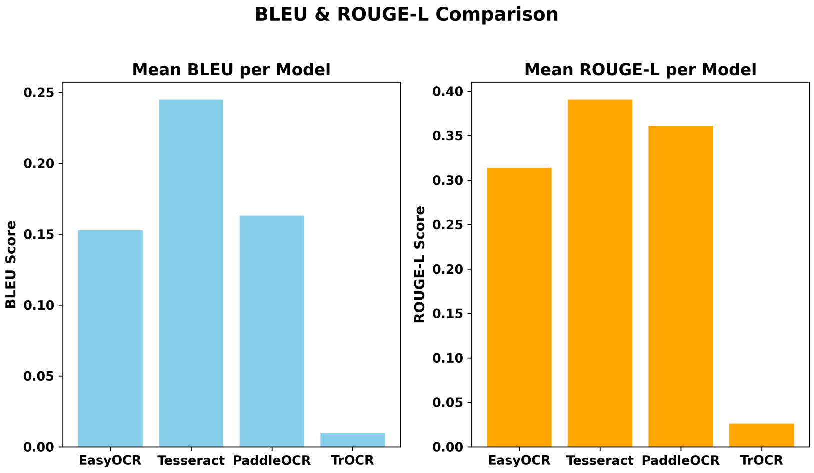

Fig. 5. BLEU and ROUGE-L comparison across OCR models.

and its ability to process structured layouts with minimal adjustments [11,12]. TrOCR, by contrast, was primarily developed for scanned documents and handwriting rather than multi-column packaging layouts [18]. These design characteristics likely explain the fragmented outputs and low semantic accuracy observed in this study.

The results have practical implications for real-world deployment. High OCR accuracy is essential for regulatory compliance, as errors in ingredient or nutritional information may lead to health and legal risks [2]. Recent evaluations of label compliance in South Africa have shown that many food packages do not present nutrition information clearly, even to human readers [4]. This is especially important in products marketed to children, where visual distractions and misleading branding can obscure key information [1]. Reliable text extraction also enables the development of digital nutrition databases, mobile health applications, and large-scale dietary research [3]. Based on these findings, Tesseract is most suitable for CPU-based environments that require high accuracy, EasyOCR is recommended for GPU-enabled scenarios with multilingual requirements, and PaddleOCR is appropriate where advanced layout handling is needed and GPU resources are available. TrOCR may only be suitable in specialized use cases that allow model fine-tuning and benefit from transformer interpretability.

6 Conclusion

This study evaluated four open-source OCR systems—Tesseract, EasyOCR, PaddleOCR, and TrOCR—on food packaging images collected from South African retail environments, focusing on ingredient lists and nutrition facts panels. Performance was measured using character-level metrics (CER and WER), semantic

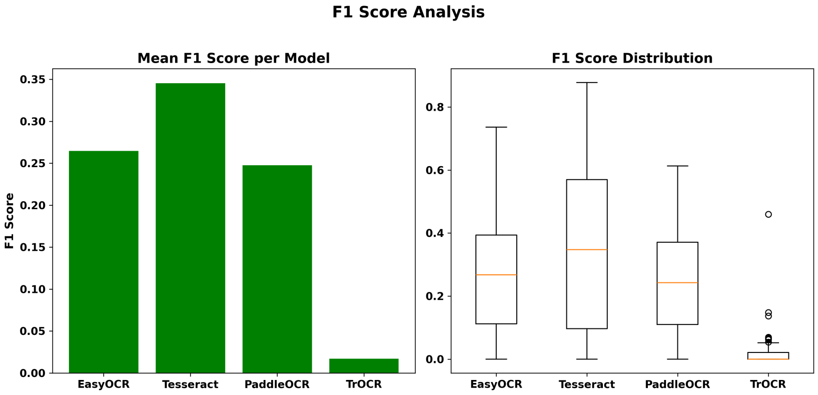

Fig. 6. F1 score distribution across OCR models.

measures (BLEU, ROUGE-L, and F1), as well as coverage and execution time. Tesseract achieved the highest accuracy across all metrics despite its older design, while EasyOCR provided balanced performance with strong multilingual capability. PaddleOCR delivered the highest coverage but required significantly longer processing times because it was limited to CPU execution in this setup. TrOCR, although GPU-accelerated, recorded the lowest accuracy, highlighting the challenges of applying transformer-based models to complex packaging layouts without domain-specific tuning.

The results provide a reference point for OCR performance on packaging data and can assist in selecting suitable systems for applications such as regulatory compliance, nutrition databases, and mobile health platforms. Future work should focus on fine-tuning OCR models for packaging text, integrating region-aware detection methods, and improving post-processing to better handle multilingual content and complex layouts.

References

1. Khan, A.S., Frank, T., Swart, R.E.: Child-directed marketing on packaged breakfast cereals in South Africa. Public Health Nutrition 26(10), 2139–2148 (2023). https://doi.org/10.1017/S1368980023001507
2. Guimarães, V., Silva, P., Rocha, J.: A review of recent advances and challenges in grocery label detection and recognition. Applied Sciences **13**(5), 2871 (2023). https://doi.org/10.3390/app13052871
3. Seitaj, H., Elangovan, V.: Information extraction from product labels: A machine vision approach. International Journal of Artificial Intelligence & Applications **15**(2), 57–76 (2024). https://doi.org/10.5121/ijaia.2024.15204
4. Abdool Karim, S., Frank, T., Khan, A.S., Tlhako, M.G., Joni, S.K., Swart, E.C.: An assessment of compliance with proposed regulations to restrict on-package marketing of packaged foods to improve nutrition in South Africa. BMC Nutrition **11**(1), 17 (2025). https://doi.org/10.1186/s40795-025-01007-3

5. Rodin, D., Orlov, N.: Fast Glare Detection in Document Images. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), vol. 7, pp. 6–9. IEEE (2019). https://doi.org/10.1109/ICDARW.2019.60123
6. Huang, J., Pang, G., Kovvuri, R., Toh, M., Liang, K.J., Krishnan, P., Yin, X., Hassner, T.: A multiplexed network for end-to-end, multilingual OCR. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4547–4557. IEEE (2021). https://doi.org/10.1109/CVPR46437.2021.00452
7. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., Gomez i Bigorda, L., Robles Mestre, S., Mas, J., Fernandez Mota, D., Almazan, J., de las Heras, L.P.: ICDAR 2013 robust reading competition. In: Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 1484–1493. IEEE (2013). https://doi.org/10.1109/ICDAR.2013.221
8. Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., Hassner, T.: TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8798–8808. IEEE (2021). https://doi.org/10.1109/CVPR46437.2021.00869
9. Shu, Y., Zeng, W., Li, Z., Zhao, F., Zhou, Y.: Visual Text Meets Low-level Vision: A Comprehensive Survey on Visual Text Processing. CoRR abs/2402.03082 (2024). https://doi.org/10.48550/arXiv.2402.03082
10. Rosyadi, A.W., Kurniawan, A., Hidayat, A.: Ingredients identification through label scanning using PaddleOCR and ChatGPT. RESTI Journal **8**(6), 758–767 (2024). https://doi.org/10.29207/resti.v8i6.6119
11. Smith, R.: An overview of the Tesseract OCR engine. In: Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR), pp. 629–633. IEEE (2007). https://doi.org/10.1109/ICDAR.2007.4376991
12. Sporici, D., Cuşnir, E., Boiangiu, C.A.: Improving the accuracy of Tesseract 4.0 OCR engine using convolution-based preprocessing. Symmetry **12**(5), 715 (2020). https://doi.org/10.3390/sym12050715
13. Saputra, M.E., Susanto, A., Carmelita, J.B.: Implementation of Tesseract OCR and bounding box for text extraction on food nutrition labels. Bulletin of Information Technology and Systems (BITS) **6**(3), 1403–1412 (2024). https://doi.org/10.47065/bits.v6i3.6107
14. JaidedAI: EasyOCR: Ready-to-Use OCR with 80+ Languages Supported. GitHub Repository (2020). https://github.com/JaidedAI/EasyOCR
15. Flores, M., Valiente, D., Alfaro, M., Fabregat-Jaén, M., Payá, L.: Evaluation of Open-Source OCR Libraries for Scene Text Recognition in the Presence of Fisheye Distortion. (2024). https://hdl.handle.net/11000/36840
16. Du, Y., Cui, C., Zhang, D., others: PP-OCR: A practical ultra lightweight OCR system. arXiv preprint arXiv:2009.09941 (2020). https://arxiv.org/abs/2009.09941
17. Cui, C., Lin, M., Sun, T., others: PaddleOCR 3.0 Technical Report. arXiv preprint arXiv:2507.05595 (2025). https://arxiv.org/abs/2507.05595
18. Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), vol. 37, no. 11, pp. 13094–13102. AAAI Press (2023). https://ojs.aaai.org/index.php/AAAI/article/view/26538
19. Nagaonkar, S., Raghunathan, V., Kumar, P., others: Benchmarking vision-language models on OCR in dynamic video environments. arXiv preprint arXiv:2502.06445 (2025). https://arxiv.org/abs/2502.06445

20. Baek, J.-C., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Lee, H.: What is wrong with scene text recognition model comparisons? Dataset and model analysis. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp.4715–4723. IEEE (2019). https://doi.org/10.1109/ICCV.2019.00481
21. Olejniczak, K., Šulc, M.: Text Detection Forgot About Document OCR. In: Sablatnig, R., Kleber, F. (eds.) 26th Computer Vision Winter Workshop (CVWW 2023), CEUR Workshop Proceedings, vol. 3349, pp. 1–7. CEUR-WS.org, Krems, Austria (2023). https://ceur-ws.org/Vol-3349/paper2.pdf
22. Neudecker, C., Baierer, K., Gerber, M., Clausner, C., Antonacopoulos, A., Pletschacher, S.: A Survey of OCR Evaluation Tools and Metrics. In: Proceedings of the 6th International Workshop on Historical Document Imaging and Processing (HIP '21), pp.13–18. ACM (2021). https://doi.org/10.1145/3476887.3476888
23. Papineni, K., Roukos, S., Ward, T., Zhu, W.-J.: BLEU: a Method for Automatic Evaluation of Machine Translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp.311–318 (2002). https://aclanthology.org/P02-1040.pdf
24. Lin, C.Y.: ROUGE: A Package for Automatic Evaluation of Summaries. In: Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), pp.74–81. ACL (2004). https://aclanthology.org/W04-1013.pdf