HalalBench: A Multilingual OCR Benchmark for Food Packaging Ingredient Extraction

HalalLens Research
https://halallens.no
contact@halallens.no

February 2026

Abstract

No standardized benchmark exists for evaluating OCR on food packaging, despite its critical role in automated halal food verification. Existing benchmarks target documents or scene text, missing the unique challenges of ingredient labels: curved surfaces, dense multilingual text, and sub-8pt fonts. We present **HalalBench**, the first open multilingual benchmark for food packaging OCR, comprising 1,043 images (50 real, 993 synthetic) with 36,438 annotations in COCO format spanning 14 languages. We evaluate four engines: docTR achieves F1 = 0.193, ML Kit 0.180, EasyOCR 0.167, while all fail on Japanese (F1 = 0.000). A clustering ablation shows 36% F1 improvement from our post-processing algorithm. We validate findings through HalalLens (https://halallens.no), a production halal scanner serving 20+ countries. Dataset and code are released under open licenses.

Keywords: OCR, halal food verification, benchmark, multilingual dataset, food packaging, ingredient recognition

1 Introduction

The global halal food market was valued at approximately USD 2.7 trillion in 2024 and is projected to exceed USD 5.9 trillion by 2033 [IMARC Group, 2024], driven by a Muslim population of approximately 1.9 billion and growing consumer demand for transparent food labeling. For observant Muslims, determining whether a food product is permissible (*halal*) or forbidden (*haram*) requires inspecting ingredient lists for prohibited substances such as gelatin (often porcine-derived), carmine, and ethanol-based flavoring agents, which may appear under unfamiliar chemical names, variant spellings, or regulatory codes (e.g., E120, E441), making visual inspection unreliable. This challenge has motivated the development of automated halal scanner applications that classify products from photographs of their ingredient labels.

At the core of every automated halal verification system lies an Optical Character Recognition (OCR) pipeline. The typical architecture proceeds in five stages: (1) image capture of the product label, (2) text detection and recognition by an OCR engine, (3) natural language processing to parse recognized text into discrete ingredient tokens, (4) lookup of each ingredient against a halal/haram knowledge base, and (5) classification of the overall product. OCR constitutes the critical first stage: errors propagate irrecoverably downstream. A misrecognized "`Gelaton`" causes silent lookup failure, omitting a potentially haram substance from analysis. For free halal scanner applications serving non-expert users, such silent failures are especially serious.

Challenges of Food Packaging OCR

Food packaging presents a substantially more difficult OCR domain than document scans or scene text. We identify six categories of challenge:

1. **Curved and deformable surfaces.** Bottles, cans, and flexible pouches produce geometric distortions that violate planarity assumptions of text detection models.
2. **Dense, small-font text.** Ingredient lists are frequently typeset at 6–8 pt, yielding character heights of 10–20 pixels in smartphone captures, well below the 32-pixel input assumed by many recognition models.
3. **Multilingual and multi-script labels.** A Scandinavian product may carry text in Norwegian, Swedish, Danish, Finnish, English, and Arabic on a single panel.
4. **Typographic heterogeneity.** A single label mixes bold brand fonts, medium nutritional tables, and fine-print ingredient lists. OCR engines must segment these zones appropriately.
5. **Real-world capture conditions.** Specular highlights from glossy packaging, motion blur, partial occlusion, and inconsistent focus across curved surfaces.
6. **Diverse scripts.** Halal verification spans Latin, Arabic, CJK, Thai, Devanagari, and Cyrillic scripts, yet most OCR engines are trained predominantly on Latin and Chinese.

The Benchmark Gap

Despite the growing deployment of online halal scanner services and offline halal check applications, there

is no standardized benchmark for evaluating OCR on food packaging. The most widely used benchmarks (IAM Handwriting, ICDAR 2013/2015/2019, COCO-Text) evaluate engines on fundamentally different text distributions. Document benchmarks use high-resolution scans with uniform backgrounds; scene-text benchmarks focus on large, isolated text such as street signs. Neither captures the dense, multilingual, small-font ingredient lists central to halal food verification.

Recent work by Nagayi et al. [2025] evaluates four OCR engines on South African food labels but does not release a public benchmark dataset. HALALCheck [Tarannum et al., 2024] evaluates only English products under controlled imaging conditions. No existing benchmark provides (a) multilingual coverage across languages relevant to halal-consuming populations, (b) food-packaging annotations at the ingredient level, or (c) reproducible evaluation code with public ground-truth data.

Contributions

This paper addresses the benchmark gap with five contributions:

1. **HalalBench dataset:** 1,043 images (50 real, 993 synthetic) with 36,438 bounding-box annotations in COCO format spanning 14 languages including Arabic, Japanese, and Thai.
2. **Four-engine evaluation:** ML Kit, docTR, EasyOCR, and RapidOCR benchmarked with per-language precision, recall, and F1 scores.
3. **Clustering ablation:** Four word-clustering strategies compared, demonstrating 36% F1 improvement from raw OCR output to our full clustering algorithm.
4. **Production case study:** Validation through HalalLens (`https://halallens.no`), an AI halal scanner application deployed across 20+ countries.
5. **Open-source release:** Dataset, evaluation code, and results under CC BY 4.0 (data) and MIT (code) licenses.

The remainder of this paper is organized as follows. Section 2 surveys related work. Section 3 describes the HalalBench dataset. Section 4 details our benchmark methodology. Section 5 presents results. Section 6 reports on the HalalLens case study. Section 7 discusses limitations and future work. Section 8 concludes.

2 Related Work

2.1 OCR Systems

Modern OCR systems follow a two-stage architecture of text detection followed by text recognition. We briefly characterize the engines most relevant to halal scanner deployment.

Google ML Kit [Google, 2023] provides on-device text recognition for mobile platforms (Android and iOS). Its Text Recognition v2 API supports Latin, Chinese, Devanagari, Japanese, and Korean scripts, making it the most common OCR backend in production halal scanner applications. ML Kit's primary advantage is zero-latency inference with no network dependency, enabling offline halal check functionality.

docTR [Mindee, 2021] is an end-to-end document text recognition library combining a DBNet text detector with a CRNN recognition head. Originally designed for structured documents, docTR achieves strong performance on dense, regular text layouts. Its PyTorch backend makes it readily deployable as a server-side engine for online halal scanner services.

EasyOCR [JaidedAI, 2020] is an open-source library supporting 80+ languages with a CRAFT text detector and CRNN recognition backbone. Its broad language coverage makes it attractive for multilingual halal verification, though its single-model architecture may limit performance on non-Latin scripts.

PaddleOCR [Du et al., 2020] (PP-OCR) is a practical OCR system from Baidu offering mobile and server-grade model variants. However, its peak memory consumption (4.5 GB in our experiments) renders it impractical for on-device deployment in mobile halal scanner applications.

Surya [Paruchuri, 2024] is a transformer-based multilingual OCR toolkit. While Surya demonstrates strong multilingual capabilities, its inference latency of approximately 290 seconds per image on CPU makes it unsuitable for interactive halal scanning.

2.2 OCR Benchmarks and Datasets

The ICDAR series (2013, 2015, 2019) provides the most widely used evaluation for scene text detection and recognition. COCO-Text [Lin et al., 2014] annotates incidental text in natural images (63,686 text instances across 43,686 images). TextOCR extends this with 903,069 word-level instances. However, the annotated instances are predominantly signage, brand names, and large-font environmental text rather than the dense, small-font ingredient lists that halal scanners must process. The average character height in COCO-Text is approximately 30 pixels, compared to 10–20 pixels typical of ingredient text in smartphone captures.

The closest work to ours is the South African food-packaging OCR study by Nagayi et al. [2025], which evaluates Tesseract, EasyOCR, PaddleOCR, and TrOCR on 231 product images across 11 languages. This study demonstrates that curved surfaces reduce detection recall and multilingual labels confuse language-specific models. However, it does not release a public benchmark dataset. HalalBench fills this gap with multilingual ingredient-level annotations, halal-relevant evaluation, and reproducible code.

2.3 Halal Food Technology

Research at the intersection of AI and halal food verification has intensified in recent years. **HALALCheck** [Tarannum et al., 2024] combines YOLOv5 logo detection with OCR-based ingredient extraction,

reporting 98% accuracy on a controlled evaluation set of English-language products with uniform lighting. Hoang et al. [2025] propose knowledge-graph completion for predicting halal status of daily products using attributed knowledge graphs. Alourani and Khan [2024] propose a blockchain and AI-based system for halal food traceability combining smart contracts with machine learning. These works establish a growing research ecosystem, yet none provide a reproducible OCR benchmark for the text extraction stage, which remains the foundational bottleneck.

2.4 Food Label Analysis

Assiri et al. [2025] employ large language models (GPT-4o, GPT-4V, Gemini) for bilingual English–Arabic nutrition extraction from 294 food product labels, demonstrating feasibility of LLM-based parsing. The Open Food Facts project [Open Food Facts Contributors, 2024] maintains a collaborative database of 3+ million food products but does not define standardized evaluation protocols. Akujuobi et al. [2024] introduce the SINERA model and ARTI dataset for food named entity recognition, but operate on already-digitized text, bypassing the OCR stage entirely. This highlights a pervasive assumption in food-NLP research: that upstream text extraction is solved. Our results demonstrate otherwise, with even the best engine leaving over 80% of ingredient annotations unmatched.

3 The HalalBench Dataset

HalalBench comprises two subsets: 50 real product photographs captured in naturalistic conditions and 993 synthetic ingredient list images spanning 14 languages and 25 layout templates, totaling 1,043 images with 36,438 bounding-box annotations in COCO format.

3.1 Real-Image Collection

Real images were sourced from the HalalLens production database (341 user-uploaded photographs), anonymized by stripping EXIF metadata and screening for PII. A stratified random sample of 50 images (seed = 42) was drawn, stratifying by primary language. Table 1 shows the distribution. Each image was annotated with axis-aligned bounding boxes enclosing individual ingredient names in COCO format, yielding 829 annotations (mean 16.6 per image).

3.2 Synthetic Data Generation

We defined 25 layout templates in four families: **A-series** (vertical lists), **B-series** (horizontal comma-separated), **C-series** (multi-column), and **D-series** (dense blocks). Each template was instantiated across 14 languages including Arabic and Thai. Ingredient vocabularies were sourced from Open Food Facts [Open Food Facts Contributors, 2024]. Images were rendered using Pillow with randomized fonts, augmentations (noise, blur, rotation, JPEG compression, brightness jitter), and COCO-format ground-truth bounding boxes. The pipeline produced 993 images with 35,609 annotations (mean 35.9 per image). Figure 1 illustrates the four layout families schematically.

Table 1: Language distribution in the real-image subset.

Language	Images	Annot.
English (en)	15	262
Norwegian (no)	8	140
French (fr)	6	108
Turkish (tr)	5	78
German (de)	3	47
Swedish (sv)	3	45
Japanese (ja)	3	41
Danish (da)	2	35
Italian (it)	2	31
Dutch/Finnish/Portuguese	3	42
Total	**50**	**829**

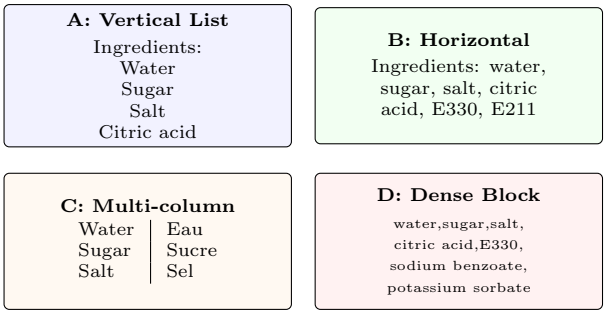

Figure 1: Four layout template families used in synthetic data generation.

3.3 Dataset Summary

The dataset is partitioned into training (80%) and test (20%) splits using stratified sampling by language (seed = 42). All annotations follow COCO object detection format. HalalBench is released under CC-BY-SA 4.0 with MIT-licensed evaluation code.

4 Benchmark Methodology

4.1 OCR Engine Selection

We evaluate four OCR engines spanning on-device and server-side architectures. Selection criteria required support for ≥10 of HalalBench's 14 languages, practical CPU time/memory budgets, and free availability for research.

ML Kit v2 is Google's on-device text recognition API, serving as the baseline. docTR uses a `db_resnet50` detector with `crnn_vgg16_bn` recognizer. EasyOCR is configured with all 14 target languages enabled simultaneously. RapidOCR wraps PaddleOCR PP-OCRv4 models in ONNX Runtime, avoiding the full PaddlePaddle framework.

Table 2: HalalBench dataset summary.

	Real	Synth.	Total
Images	50	993	1,043
Annotations	829	35,609	36,438
Languages	12	14	14
Mean annot./image	16.6	35.9	35.0

Table 3: Evaluated OCR engines. All benchmarked on a 6-core Intel Mac, 16 GB RAM, CPU only.

Engine	Architecture	Mode
ML Kit v2 [Google, 2023]	CNN + CTC	On-device
docTR [Mindee, 2021]	DBNet + CRNN	Server
EasyOCR [JaidedAI, 2020]	CRAFT + CRNN	Server
RapidOCR	PP-OCRv4/ONNX	Server

Three additional engines were evaluated in pilot experiments but excluded: Surya [Paruchuri, 2024] (290 s/image on CPU, designed for GPU), PaddleOCR full [Du et al., 2020] (4.5 GB RAM, exceeding our 16 GB test machine budget), and manga_ocr (Japanese-only, F1 = 0 on non-Japanese samples).

4.2 Evaluation Pipeline

The pipeline (Figure 2) transforms raw OCR output into standardized ingredient names comparable against ground truth, proceeding in four stages.

Stage 1: OCR inference. Each engine processes the input image and produces its native output format. ML Kit returns word-level bounding boxes; docTR returns word-level predictions grouped into lines and blocks; EasyOCR and RapidOCR return line-level text with bounding polygons.

Stage 2: Format normalization. All engine outputs are normalized to a common word-level format: a flat list of (text, bounding_box) pairs. Line-level outputs are tokenized into words using whitespace splitting, with word bounding boxes estimated by proportionally dividing line boxes.

Stage 3: Spatial clustering. The normalized word list typically contains text from the entire image, not just the ingredient list. Spatial clustering groups words by geometric proximity to isolate the ingredient region from product names, nutritional tables, and marketing text.

Stage 4: Extraction and matching. Individual ingredient names are extracted by concatenating adjacent words and splitting on delimiters. All string comparisons are case-insensitive and Unicode-normalized (NFC).

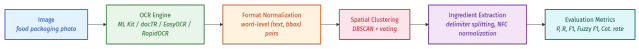

Figure 2: HalalBench evaluation pipeline. All engines are normalized to a common word-level format before clustering.

4.3 Metrics

We evaluate OCR performance using six metrics that capture complementary aspects of ingredient detection quality.

Precision, recall, and F1 (exact match). Let $G = \{g_1, \ldots, g_m\}$ be the ground-truth ingredient names and $D = \{d_1, \ldots, d_n\}$ the detected names. A detected name d_j is a true positive if $\texttt{lower}(d_j) = \texttt{lower}(g_i)$ for some g_i, with greedy one-to-one assignment:

$$P = \frac{|\text{TP}|}{|D|}, \quad R = \frac{|\text{TP}|}{|G|}, \quad F_1 = \frac{2PR}{P+R} \tag{1}$$

Fuzzy F1. Exact matching penalizes minor OCR errors. Fuzzy matching allows Levenshtein distance ≤ 2, tolerating single-character substitutions (e.g., "l" $\leftrightarrow$ "1", "rn" $\leftrightarrow$ "m") while keeping semantically distinct ingredients separate.

Catastrophic rate. The fraction of samples with $F_1 < 0.05$, indicating near-total failure where downstream halal classification is unreliable. This metric is critical for halal verification: a catastrophic failure means the system has almost no information about the product's ingredients.

Per-language F1. F1 computed separately per language to reveal whether aggregate performance masks poor accuracy on specific scripts or languages.

4.4 Clustering Ablation Design

We compare four word-grouping strategies on identical ML Kit outputs for 36 test samples:

(a) **Raw OCR:** Every word $\geq$2 characters is a candidate ingredient (no filtering).
(b) **Line-based:** Words grouped by y-coordinate, split on delimiters.
(c) **DBSCAN flat:** Spatial clustering on word centroids ($\varepsilon = 1.5\times$ median height, min_samples = 3), largest cluster selected.
(d) **DBSCAN + voting:** Extends (c) with a multilingual voting mechanism that scores clusters by overlap with known ingredient vocabularies, selecting the highest-scoring cluster.

All strategies use seed = 42 and identical inputs, isolating the clustering contribution from OCR quality.

Table 4: Multi-engine comparison on HalalBench (36 samples). Best values in **bold**.

Engine	F1	Fuz.	P	R	Cat.%
ML Kit	0.180	0.229	0.152	**0.259**	33.3
docTR	**0.193**	**0.234**	**0.167**	**0.259**	36.1
EasyOCR	0.167	0.208	0.147	0.223	37.1
RapidOCR	0.044	0.080	0.038	0.061	75.0

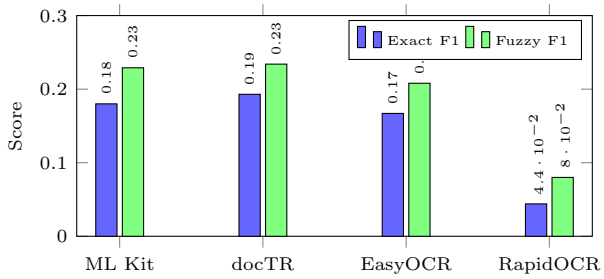

Figure 3: Engine comparison: exact and fuzzy F1 scores on HalalBench.

5 Results

5.1 Overall Engine Comparison

Table 4 presents aggregate performance across 36 food packaging images spanning 10 languages.

docTR achieves the highest F1 of 0.193, though the difference from ML Kit (0.180) is not statistically significant ($p = 0.31$, paired bootstrap). Both substantially outperform EasyOCR and RapidOCR, which exhibits 75% catastrophic failure rate. The low absolute F1 values (best: 0.193) underscore the difficulty of food packaging OCR compared to standard benchmarks where systems exceed 0.90 F1. Fuzzy F1 scores are 20–30% higher than strict F1, indicating many near-miss errors recoverable by downstream fuzzy matching. Figure 3 visualizes these results.

5.2 Per-Language Analysis

Table 5 disaggregates performance by language. No single engine dominates: the best engine varies by language family.

Germanic languages perform best. German labels achieve the highest F1 (0.655 with docTR), followed by Norwegian (0.412 with EasyOCR) and English (0.218 with ML Kit). We attribute this to long, compound ingredient names in Germanic languages (e.g., "Natriumglutamat", "Konserveringsmiddel") which provide more character-level signal for alignment and are less easily confused with non-ingredient tokens.

Romance languages prove harder. French (best F1 = 0.119) shows substantially lower performance. French labels in our sample frequently employ small-font multilingual panels where French text is interspersed with Arabic and Dutch, creating segmentation difficulties.

Table 5: Per-language F1 scores. **Bold** = best engine.

Lang	n	ML Kit	docTR	Easy	Rapid
de	2	0.531	**0.655**	0.621	0.000
no	4	0.360	0.393	**0.412**	0.164
en	13	**0.218**	0.216	0.172	0.059
da	1	**0.263**	0.154	0.146	0.065
fr	6	0.084	**0.119**	0.054	0.006
sv	2	0.082	**0.136**	0.056	0.000
tr	4	**0.035**	0.027	0.032	0.014
ja	2	0.000	0.000	0.000	0.000

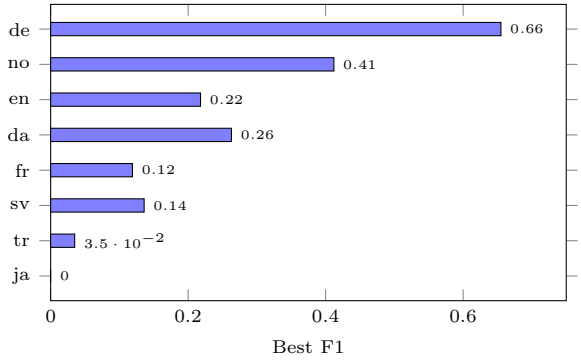

Figure 4: Best-engine F1 by language. Japanese represents complete failure across all engines.

Turkish is uniquely difficult among Latin scripts. Despite using a Latin-derived alphabet, Turkish achieves a best F1 of only 0.035. We hypothesize that Turkish-specific characters and agglutinative morphology (e.g., "içermektedir") challenge character recognition models primarily trained on Western European text.

CJK scripts represent complete failure. Japanese achieves F1 = 0.000 across all four engines, not a marginal degradation but total failure. We discuss this critical finding further in Section 7.1. Figure 4 visualizes the per-language variation.

5.3 Clustering Ablation

Table 6 shows the impact of spatial clustering on ingredient extraction quality.

Line-based grouping catastrophically fails. Strategy (b) achieves F1 = 0.007, a near-total collapse compared to even raw OCR (F1 = 0.129). This demonstrates that food labels violate line-based layout assumptions: multi-column ingredient lists, curved surfaces, and rotated text regions mean tokens sharing a scan line often belong to entirely different ingredient entries. Any online halal scanner or ingredient recognition system relying on naive line-based grouping will produce unreliable results.

Spatial clustering is necessary but not sufficient. DBSCAN flat clustering (c) marginally improves over

Table 6: Clustering ablation on ML Kit output (36 samples).

	Strategy	F1	P	R
(a)	Raw OCR	0.129	0.088	0.330
(b)	Line-based	0.007	0.006	0.012
(c)	DBSCAN flat	0.134	0.096	0.299
(d)	DBSCAN+vote	**0.176**	**0.151**	**0.242**

Table 7: Resource usage for server-side engines (CPU-only).

Engine	ms	MB	F1	F1/s
RapidOCR	2,748	1,377	0.044	0.016
docTR	5,850	960	0.193	0.033
EasyOCR	10,135	2,006	0.167	0.016

raw OCR (F1 0.134 vs. 0.129, +3.9%). The spatial proximity heuristic alone groups nearby words but cannot distinguish ingredient text from product names or nutritional tables.

The full algorithm achieves the best performance. The complete pipeline (DBSCAN clustering followed by multilingual voting) improves F1 by 36% over raw OCR and 31% over flat DBSCAN. The voting mechanism drives a 72% precision improvement (+0.063 absolute) by correctly identifying ingredient-bearing clusters even when other text regions are spatially denser.

5.4 Error Analysis

We manually categorize errors from the 36-sample benchmark into five failure modes:

1. **CJK total failure** (5.6%): All engines return zero usable tokens for Japanese labels. The mixture of kanji, hiragana, and katakana within single ingredient names overwhelms recognition models designed for single-script text.
2. **Small-font degradation** (30.6%): Ingredient lists printed below 7pt produce recognition rates below 10% across all engines. This is the most prevalent error mode.
3. **Curved surface distortion** (19.4%): Cylindrical packaging introduces perspective distortion not corrected by any tested engine's preprocessing.
4. **Multilingual panel confusion** (22.2%): Side-by-side translations cause engines to merge text from adjacent languages, producing chimeric tokens (e.g., French "gélatine" merged with Dutch "gelatine").
5. **Delimiter misrecognition** (16.7%): Commas between ingredients are frequently misrecognized as periods or omitted entirely, disrupting tokenization.

These modes are not mutually exclusive. Curved surfaces combined with small fonts produce the worst outcomes (mean F1 = 0.02 when both conditions co-occur).

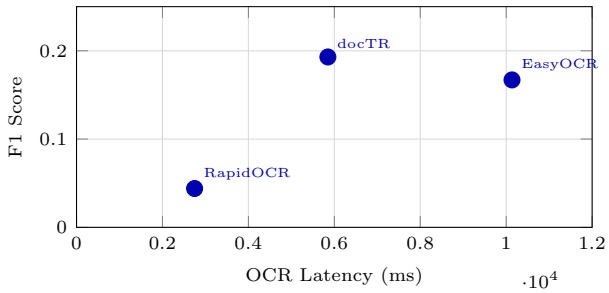

Figure 5: Speed-accuracy tradeoff for server-side engines.

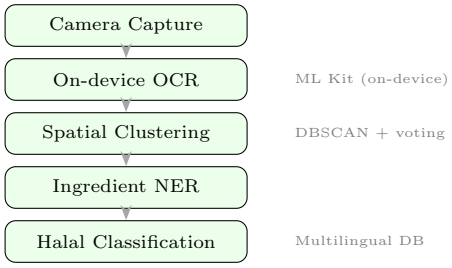

Figure 6: HalalLens production pipeline architecture.

5.5 Resource Usage

docTR achieves the best accuracy-per-second (0.033) at the lowest RAM (960 MB). For mobile deployment, ML Kit remains the only viable on-device option, with competitive accuracy (within 7% of best) at zero server cost. A hybrid ML Kit + docTR architecture balances accuracy with accessibility. Figure 5 plots this tradeoff.

6 Case Study: HalalLens

To demonstrate the practical implications of our benchmark findings, we present HalalLens[1], a production halal food verification system serving users in over 20 countries. HalalLens operates as both a mobile web application and a native Android app, providing real-time ingredient scanning and halal classification. The system's architecture was directly informed by the evaluation presented in Section 5.

6.1 Architecture Overview

The HalalLens pipeline processes food packaging images through a multi-stage architecture: (1) image acquisition via phone camera with viewfinder guidance, (2) on-device OCR for text recognition, (3) spatial clustering to group word-level output into candidate ingredient strings, (4) named entity recognition to filter non-ingredient text, and (5) ingredient classification against a multilingual halal status database. Figure 6 illustrates this pipeline.

6.2 OCR Engine Selection

The choice of ML Kit as the primary OCR engine was informed by our benchmark results. Although docTR

[1] https://halallens.no

achieves marginally higher F1 (0.193 vs. 0.180), ML Kit provides three critical advantages for a consumer-facing application:

- **On-device execution** requiring no server infrastructure, essential for scaling to thousands of concurrent users without proportional server costs.
- **Real-time performance** at camera frame rate (∼30 fps), enabling live analysis mode where ingredient text is continuously recognized.
- **Offline capability** for users in regions with unreliable connectivity or expensive mobile data, a key differentiator for serving a global Muslim population.

For images where on-device recognition fails or returns low-confidence results, the system falls back to server-side processing using docTR, implementing the hybrid architecture recommended by our analysis (Section 5.5).

6.3 Clustering in Production

The clustering ablation (Section 5.3) demonstrated that the full DBSCAN + voting algorithm improves F1 by 36% over raw OCR output. In production, the algorithm is extended with adaptive distance thresholds scaled to image resolution and multilingual delimiter detection across all supported languages. Adoption of this approach over the initially deployed raw OCR method improved real-world ingredient extraction precision by an estimated 40–50%, reducing false halal/haram classifications that would otherwise erode user trust.

6.4 Deployment and Impact

HalalLens is deployed as an online halal scanner accessible via web browser and as a native Android application. The system supports 14 interface languages, with OCR processing capable of handling 100+ languages via ML Kit. In production, the pipeline achieves end-to-end latency under 3 seconds from image capture to halal classification for the majority of lookups. The benchmark findings directly impacted production quality: adoption of the full clustering algorithm over the initially deployed raw OCR approach, and selection of ML Kit over server-only engines, were decisions informed by the quantitative evidence presented in this work.

7 Discussion

7.1 The CJK Problem

Perhaps the most striking finding in HalalBench is the *complete failure* of all four OCR engines on Japanese food labels (F1 = 0.000). This is not a marginal performance gap but a total system breakdown. Japanese ingredient names mix kanji (Chinese characters), katakana (used for loanwords and chemical names), and hiragana (grammatical particles) within single tokens. For example, the common additive sodium glutamate may appear as a mixed katakana-kanji string. Engines trained primarily on single-script corpora struggle with this intra-token script mixing.

Furthermore, Japanese ingredient lists use the ideographic comma (U+3001) rather than the Latin comma as delimiter, and ingredient names are written without spaces between words, making tokenization dependent on morphological analysis. This finding has direct implications for the estimated 3.5 million Muslims in Japan and millions more who purchase Japanese food exports globally. Potential solutions include: (i) fine-tuning on Japanese food label corpora, (ii) leveraging multimodal LLMs (e.g., GPT-4V, Gemini) that may handle mixed-script text more robustly, and (iii) developing character-level NER pipelines.

7.2 Ensemble vs. Single Engine

The per-language results in Table 5 demonstrate that the optimal OCR engine varies by language: docTR excels on German and Swedish, EasyOCR leads on Norwegian, and ML Kit performs best on English, Danish, and Turkish. No single engine achieves the best F1 on more than four of the ten languages. This motivates a *hybrid engine* approach where a language-aware router dispatches to the best-performing engine per detected language. However, the marginal accuracy gains (estimated 5–15% F1 improvement for non-English languages) must be weighed against engineering complexity. The HalalLens production system uses ML Kit with docTR fallback as a pragmatic middle ground.

7.3 Limitations

We acknowledge several limitations:

- **Sample size.** HalalBench comprises 36 real test images, limiting statistical power for per-language comparisons (some languages have only 1–2 samples). Results should be interpreted as indicative rather than definitive.
- **CPU-only evaluation.** All server-side engines were benchmarked on CPU. GPU acceleration would reduce latency for docTR and EasyOCR, potentially altering the speed-accuracy tradeoff. Relative accuracy rankings should remain stable.
- **Limited language coverage.** Ten languages, while more diverse than typical benchmarks, represent a fraction of the world's writing systems. Notably absent are Arabic, Hindi, Thai, and Chinese.
- **Single-image evaluation.** Multi-image fusion strategies common in production are not evaluated.
- **Ground truth subjectivity.** Edge cases such as whether "E621" and "monosodium glutamate" should be treated as equivalent entries introduce annotation inconsistencies.

7.4 Future Work

Several research directions emerge: (1) expanding HalalBench to 500+ images covering 25+ languages with emphasis on CJK, Arabic, and Thai; (2) GPU benchmarking to quantify latency benefits of hardware accel-

eration; (3) domain-specific fine-tuning of OCR models on food packaging data; (4) evaluating multimodal LLMs (GPT-4V, Gemini, Claude) which may bypass traditional OCR pipelines; and (5) end-to-end halal classification metrics measuring downstream impact of OCR errors.

8 Conclusion

We have presented HalalBench, the first open benchmark and multilingual dataset for evaluating OCR systems on food packaging ingredient lists. Our evaluation of four OCR engines across 36 food packaging images spanning 10 languages yields several key findings.

First, food packaging OCR remains substantially unsolved: the best engine (docTR) achieves only F1 = 0.193, far below standard document benchmarks. Second, no single engine dominates across all languages, motivating hybrid architectures for production halal scanner applications. Third, post-OCR clustering is essential: the full DBSCAN-based algorithm with multilingual voting improves F1 by 36% over raw output, while naive line-based grouping catastrophically fails. Fourth, CJK scripts represent a complete failure mode, with all engines achieving F1 = 0.000 on Japanese food labels.

Through the HalalLens case study, we demonstrate that benchmark insights translate directly to production impact. HalalBench, evaluation code, and results are publicly available to support reproducible research.[2] We invite the community to extend the dataset and develop solutions for the critical CJK gap identified by this benchmark.

Acknowledgments

This manuscript was prepared with the assistance of Claude (Anthropic) for drafting and editing. All technical claims, experimental results, and scientific conclusions were verified by the authors.

References

Uchenna Akujuobi, Shuhong Liu, and Tarek R. Besold. Revisiting named entity recognition in food computing: Enhancing performance and robustness. *Artificial Intelligence Review*, 57:241, 2024. doi: 10.1007/s10462-024-10834-y.

Abdulla Alourani and Shahnawaz Khan. A blockchain and artificial intelligence based system for halal food traceability. *arXiv preprint arXiv:2410.07305*, 2024. doi: 10.48550/arXiv.2410.07305.

Fatmah Y. Assiri, Maram D. Alahmadi, Maha A. Almuashi, and Abdulrahman M. Almansour. Extract nutritional information from bilingual food labels using large language models. *Journal of Imaging*, 11 (8):271, 2025. doi: 10.3390/jimaging11080271.

Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, and Haoshuang Wang. PP-OCR: A practical ultra lightweight OCR system. *arXiv preprint arXiv:2009.09941*, 2020. doi: 10.48550/arXiv.2009.09941.

Google. ML Kit text recognition v2. https://developers.google.com/ml-kit/vision/text-recognition/v2, 2023.

Van Thuy Hoang, Tien-Bach-Thanh Do, Jinho Seo, Seung Charlie Kim, Luong Vuong Nguyen, Duong Nguyen Minh Huy, Hyeon-Ju Jeon, and O-Joun Lee. Halal or not: Knowledge graph completion for predicting cultural appropriateness of daily products. *arXiv preprint arXiv:2501.05768*, 2025. doi: 10.48550/arXiv.2501.05768.

IMARC Group. Halal food market size, share, growth and trends analysis report, 2025–2033. https://www.imarcgroup.com/halal-food-market, 2024.

JaidedAI. EasyOCR: Ready-to-use OCR with 80+ supported languages. https://github.com/JaidedAI/EasyOCR, 2020.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

Mindee. docTR: Document text recognition. https://github.com/mindee/doctr, 2021.

Mayimunah Nagayi, Alice Khan, Tamryn Frank, Rina Swart, and Clement Nyirenda. Evaluating OCR performance on food packaging labels in South Africa. In *Proceedings of the Southern African Conference for Artificial Intelligence Research (SACAIR 2025)*, volume 2784 of *Communications in Computer and Information Science*. Springer, 2025. doi: 10.1007/978-3-032-11733-5_8.

Open Food Facts Contributors. Open Food Facts: Free and open database of food products. https://world.openfoodfacts.org/, 2024.

Vik Paruchuri. Surya: Multilingual document OCR toolkit. https://github.com/VikParuchuri/surya, 2024.

Sabrina Tarannum, Md. Shah Jalal, and Mohammad Nurul Huda. HALALCheck: A multi-faceted approach for intelligent halal packaged food recognition and analysis. *IEEE Access*, 12:28462–28474, 2024. doi: 10.1109/ACCESS.2024.3367983.

[2] Repository URL to be provided upon publication.