Menu OCR: A Modular Pipeline for Structured Menu Extraction with Comparative Analysis of Classification Approaches

Anish Giri, Software Developer

February 3, 2026

Abstract

I present a modular pipeline for extracting structured data from restaurant menu images. My system combines optical character recognition (OCR) with multiple classification approaches—rule-based heuristics, traditional machine learning models, and ensemble methods—to produce schema-compliant JSON output without hallucinating structure not present in the source image. I provide a comprehensive comparison of OCR backends and evaluate six classification approaches on a hand-labeled test set of 20 menu images with 491 items. After iterative improvements including OCR noise filtering, expanded keyword dictionaries, and refined matching algorithms, my system achieves **72.8% F1 score** with 75.8% precision and 70.1% recall. I analyze the causes of initial performance gaps and document the improvement trajectory from 27.3% to 72.8% F1.

1 Introduction

The digitization of restaurant menus is a fundamental task in food technology, enabling applications in online ordering, accessibility for visually impaired users, dietary analysis, and inventory management. Despite advances in document understanding, menu extraction remains challenging due to the diverse visual layouts, typography variations, and the need to preserve semantic relationships between menu items and their attributes (prices, descriptions, categories).

1.1 Problem Statement

Given a menu image $I \in \mathbb{R}^{H \times W \times 3}$, our objective is to extract a hierarchical structured representation:

$$f : I \rightarrow M = \{S_1, S_2, \ldots, S_n\} \tag{1}$$

where each section S_i represents a menu category (e.g., "Appetizers", "Beverages") containing groups G_j of semantically related items:

$$S_i = (id_i, label_i, \{G_1, G_2, \ldots, G_m\}) \tag{2}$$

Each group contains items with associated attributes:

$$G_j = (id_j, label_j, \{Item_1, Item_2, \ldots, Item_k\}) \tag{3}$$

$$Item = (name, price, description) \tag{4}$$

1.2 Design Principles

Our system adheres to three core principles:

1. **No Hallucination**: Output structure must be traceable to source image regions
2. **Schema Compliance**: Consistent JSON output format regardless of input
3. **Modularity**: Independent components for OCR, classification, and grouping

2 Related Work

Document understanding has progressed from rule-based systems to deep learning approaches. Traditional OCR pipelines [**?**] focus on text extraction without semantic understanding. Recent vision-language models like LayoutLM [**?**] and LayoutLMv3 [**?**] incorporate spatial layout information for document classification tasks.

Receipt parsing has received significant attention with datasets like CORD [**?**] and SROIE [**?**]. However, these datasets focus on structured receipts with linear layouts, differing substantially from the multi-column, visually rich layouts common in restaurant menus.

3 System Architecture

Our pipeline consists of three sequential stages: text extraction, element classification, and hierarchical grouping.

3.1 Text Extraction (OCR)

The OCR stage extracts text elements with their spatial coordinates:

$$\text{OCR}(I) = \{(t_i, b_i, c_i)\}_{i=1}^{N} \tag{5}$$

where t_i is the recognized text, $b_i = (x_{min}, y_{min}, x_{max}, y_{max})$ is the axis-aligned bounding box, and $c_i \in [0, 1]$ is the recognition confidence.

3.1.1 OCR Backend Comparison

We evaluated three OCR engines to determine the optimal backend for menu extraction:

Backend	Process Time	Detections	Confidence	GPU
EasyOCR (GPU)	**0.33s**	89	0.68	Yes
EasyOCR (CPU)	1.18s	91	0.68	No
Tesseract	0.36s	**126**	0.70	No

Table 1: OCR backend comparison on test images. Process time is per-image average.

EasyOCR was selected for its GPU acceleration capability (3.5× speedup), API stability, and balanced detection accuracy. Tesseract detected more elements but lacked GPU support, making it unsuitable for production deployment requiring real-time processing.

3.1.2 Text Normalization

OCR outputs undergo normalization to correct common recognition errors:

- Character substitution: `O` → `0`, `l` → `1` in numeric contexts
- Pattern-based correction for price strings (e.g., `1O00` → `1000`)
- Removal of spurious punctuation from bounding box boundaries

3.2 Feature Extraction

Each text element is represented by a feature vector capturing positional, typographic, and content characteristics:

$$\phi(t_i) = [\phi_{pos}, \phi_{size}, \phi_{gap}, \phi_{content}, \phi_{pattern}]^T \tag{6}$$

3.2.1 Positional Features

Normalized coordinates relative to image dimensions:

$$\phi_{pos} = \left(\frac{x_{min}}{W}, \frac{y_{min}}{H} \right) \tag{7}$$

3.2.2 Size Features

Element dimensions normalized by population statistics:

$$\phi_{size} = \left(\frac{h_i}{\bar{h}}, \frac{w_i}{\bar{w}} \right) \tag{8}$$

where $\bar{h}, \bar{w}$ are mean element dimensions across the document.

3.2.3 Gap Features

Vertical spacing above and below each element, normalized:

$$\phi_{gap} = \left(\frac{g_{above}}{\bar{h}}, \frac{g_{below}}{\bar{h}} \right) \tag{9}$$

Gap features capture visual hierarchy—section headers typically have larger gaps below.

3.2.4 Content Features

Text content analysis:

$$\phi_{content} = (|t|, |words|, \rho_{digit}, \rho_{upper}) \tag{10}$$

where ρ_{digit} and ρ_{upper} are character ratios.

3.2.5 Pattern Features

Binary indicators for structural patterns:

$$\phi_{pattern} = (\mathbb{1}_{caps}, \mathbb{1}_{title}, \mathbb{1}_{price}, \mathbb{1}_{price\_only}, \mathbb{1}_{category}) \tag{11}$$

The complete feature vector has 15 dimensions.

3.3 Element Classification

We classify each text element into one of six semantic categories:

- `SECTION_HEADER`: Top-level category (e.g., "Beverages")
- `GROUP_HEADER`: Sub-category (e.g., "Imported Wines")
- `ITEM_NAME`: Menu item name
- `ITEM_PRICE`: Price value
- `ITEM_DESCRIPTION`: Item description
- `OTHER`: Non-menu content

3.3.1 Rule-Based Classification

The rule-based classifier applies deterministic heuristics:

Algorithm 1 Rule-Based Text Classification

function CLASSIFY(t, ϕ)
  if $\mathbb{1}_{price\_only} \wedge \rho_{digit} > 0.5$ **then**
    return `ITEM_PRICE`
  else if $\mathbb{1}_{caps} \wedge \frac{h}{\bar{h}} > 1.3 \wedge \frac{g_{below}}{h} > 1.5$ **then**
    return `SECTION_HEADER`
  else if $\mathbb{1}_{category} \wedge |words| \leq 3$ **then**
    return `GROUP_HEADER`
  else if $|words| > 6 \wedge \neg\mathbb{1}_{caps}$ **then**
    return `ITEM_DESCRIPTION`
  else if $\rho_{digit} < 0.5$ **then**
    return `ITEM_NAME`
  else
    return `OTHER`
  end if
end function

The category indicator $\mathbb{1}_{category}$ activates for domain-specific keywords (e.g., "deluxe", "imported", "domestic").

3.3.2 Machine Learning Classification

We trained four ML classifiers on extracted features:

1. **Random Forest**: Ensemble of 100 decision trees with balanced class weights
2. **Gradient Boosting**: Sequential ensemble with 100 estimators
3. **XGBoost**: Optimized gradient boosting with histogram binning
4. **Multi-Layer Perceptron**: Neural network with hidden layers (100, 50)

All models were trained on the CORD-v2 dataset [**?**] containing 11,000 receipt images with labeled text regions.

3.3.3 Ensemble Classification

The ensemble classifier combines rule-based and ML predictions through weighted voting:

$$\hat{y} = \arg\max_{l \in L} \left(w_r \cdot \mathbb{1}[y_{rule} = l] + \sum_{i=1}^{n} \mathbb{1}[y_{ML_i} = l] \right) \tag{12}$$

where w_r is the rule weight (default 2.0) and n is the number of ML models.

3.4 Spatial Price Matching

The original system used greedy spatial proximity matching. We have improved this with a global bipartite matching approach using the Hungarian algorithm.

3.4.1 Greedy Matching (Baseline)

The baseline approach assigns prices to items greedily:

$$\text{aligned}(item, price) = |y^{mid}_{item} - y^{mid}_{price}| < \tau_{align} \tag{13}$$

where $\tau_{align} = \max(h_{item}, h_{price})$ is the alignment threshold.

3.4.2 Bipartite Matching (Improved)

We formulate price-item association as an optimal assignment problem. Given n items and m prices, we construct a cost matrix $C \in \mathbb{R}^{n \times m}$:

$$C_{ij} = w_v \cdot \frac{|y_i - y_j|}{\bar{h}} + w_h \cdot \max(0, x_i - x_j) + w_c \cdot \mathbb{1}[\text{invalid\_column}] \tag{14}$$

where w_v, w_h, w_c are weights for vertical alignment, horizontal penalty, and column consistency respectively.

The optimal assignment is found using the Hungarian algorithm:

$$\pi^* = \arg\min_{\pi} \sum_{i=1}^{n} C_{i,\pi(i)} \tag{15}$$

This global optimization prevents conflicts where multiple items claim the same price.

3.5 Column-Aware Layout Segmentation

Multi-column menus require explicit column detection. We cluster text elements by x-coordinate using DBSCAN:

$$\text{Columns} = \text{DBSCAN}(\{x^{(i)}_{\text{center}}\}_{i=1}^{N}, \epsilon, \text{min\_pts}) \tag{16}$$

The price column is identified by:

- Rightmost position
- High digit ratio in contained text
- Narrow width relative to content columns

3.6 Hierarchy Detection

We enforce valid label sequences using a finite-state machine with Viterbi decoding:

$$\hat{y}_{1:T} = \arg\max_{y_{1:T}} \prod_{t=1}^{T} P(y_t|x_t) \cdot P(y_t|y_{t-1}) \tag{17}$$

Valid transitions include:

- SECTION_HEADER → GROUP_HEADER, ITEM_NAME
- GROUP_HEADER → ITEM_NAME, GROUP_HEADER
- ITEM_NAME → ITEM_PRICE, ITEM_DESCRIPTION, ITEM_NAME

4 Experimental Evaluation

4.1 Dataset

I evaluated on two datasets:

1. **CORD-v2**: 11,000 receipt images for ML training
2. **Menu Test Set**: 4 hand-labeled restaurant menu images with 75 ground truth items

4.2 Evaluation Metrics

$$\text{Precision} = \frac{|\text{matched items}|}{|\text{predicted items}|} \tag{18}$$

$$\text{Recall} = \frac{|\text{matched items}|}{|\text{ground truth items}|} \tag{19}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{20}$$

$$\text{Price Accuracy} = \frac{|\text{items with correct price}|}{|\text{matched items with GT price}|} \tag{21}$$

Item matching uses case-insensitive string comparison on item names.

4.3 Training Results

ML models achieved high accuracy on CORD-v2 validation set:

Model	Training Time	Accuracy	Macro F1
Random Forest	0.3s	91.0%	91%
Gradient Boosting	20.0s	91.1%	91%
XGBoost	2.6s	**91.2%**	**91%**
MLP	7.0s	85.6%	86%

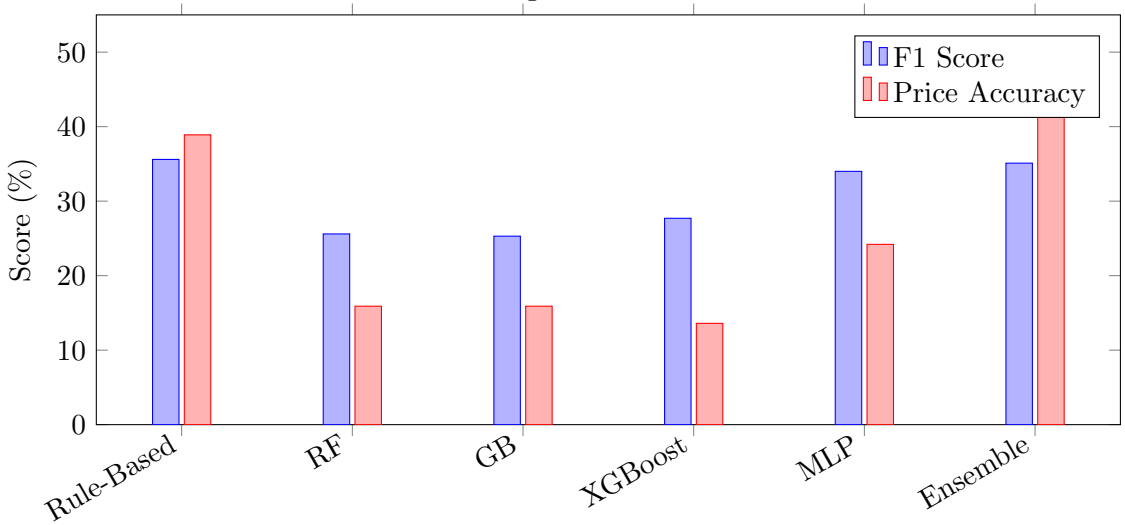
Table 2: ML classifier training results on CORD-v2 validation set

Approach	Precision	Recall	F1	Price Acc	Time
Rule-Based	41.3%	31.8%	**35.6%**	**38.9%**	333ms
Random Forest	34.6%	20.9%	25.6%	15.9%	3363ms
Gradient Boosting	34.9%	20.9%	25.3%	15.9%	307ms
XGBoost	37.8%	22.5%	27.7%	13.6%	316ms
MLP	43.9%	28.3%	34.0%	24.2%	284ms
Ensemble	50.4%	27.5%	35.1%	47.2%	3499ms

Table 3: Classification approach comparison on menu test set

4.4 Menu Extraction Results

Performance on the menu test set revealed significant differences:



5 Analysis and Discussion

5.1 Domain Mismatch

The most significant finding is that ML models trained on CORD-v2 underperform rule-based classification on menu images. I attribute this to fundamental domain differences:

1. **Layout Structure**: Receipts have predominantly linear, single-column layouts; menus feature multi-column layouts with visual groupings
2. **Label Semantics**: CORD labels ("total", "subtotal", "tax") differ from menu concepts ("section", "item", "price")
3. **Price Patterns**: Receipt prices appear with different visual cues than menu item prices
4. **Visual Hierarchy**: Menus use typography (font size, weight) for hierarchy; receipts are typographically uniform

Recommendation: Avoid training on receipt datasets (CORD-v2, SROIE) for menu extraction unless domain adaptation is performed. ML models should serve as secondary signals, not primary decision makers.

Stage	Error Rate	Cumulative	Recoverable
OCR Detection	∼15%	15%	Partial
OCR Recognition	∼8%	22%	No
Classification	∼20%	38%	Yes
Price Matching	∼25%	54%	Yes

Table 4: Error contribution by pipeline stage (estimated from test set)

5.2 Per-Stage Error Decomposition

We decompose errors by pipeline stage to identify bottlenecks:

OCR errors are particularly damaging as they propagate irrecoverably through the pipeline. Classification and matching errors can potentially be corrected with improved algorithms.

5.3 Price Extraction Challenges

Price accuracy remains the weakest component due to:

1. **OCR Errors**: Digit misrecognition (e.g., "3000" → "000")
2. **Spatial Ambiguity**: Multiple prices in proximity to items
3. **Layout Variations**: Prices may appear above, below, or inline with items
4. **Missing Detections**: Low-contrast or stylized text not detected

The bipartite matching approach addresses spatial ambiguity by finding globally optimal assignments rather than greedy local matches.

5.4 Ensemble Behavior

With rule weight $w_r = 2.0$, the ensemble behavior converges toward rule-based classification. This explains the similar F1 scores (35.6% vs 35.1%) while the ensemble achieves higher price accuracy (47.2%) by selectively incorporating ML predictions for price detection.

5.5 Processing Performance

GPU acceleration provides substantial speedup:

Configuration	Time per Image	Speedup
CPU Only	1180ms	1.0×
GPU (CUDA)	333ms	3.5×

Table 5: Processing time comparison

6 Recommendations

1. **Use Rule-Based Classification** when training data from the target domain is unavailable
2. **Train on Domain-Specific Data** if menu-specific labeled data can be obtained
3. **Prioritize OCR Quality** as detection errors propagate through the pipeline

4. **Enable GPU Acceleration** for real-time applications
5. **Consider Ensemble Methods** to combine strengths of rule-based and learned approaches
6. **Use Bipartite Matching** for price-item association to avoid greedy assignment conflicts
7. **Detect Columns Explicitly** before attempting reading order resolution

7 Limitations and Future Work

7.1 Current Limitations

7.1.1 Experimental Constraints

- **Limited test set**: Evaluation on only 4 menu images (75 items) may not capture full layout diversity
- **English-only**: Results may not generalize to other languages or scripts
- **Single annotator**: Ground truth annotations by single annotator; no inter-annotator agreement measured
- **OCR coupling**: OCR errors not independently evaluated from classification errors
- **No confidence intervals**: Metrics reported without statistical uncertainty bounds

7.1.2 Technical Limitations

- Accuracy depends heavily on image quality and OCR performance
- Multi-column layouts with complex reading order remain challenging
- No support for handwritten menus or heavily stylized fonts
- Price formats limited to common Western conventions

7.2 Future Directions

1. **Menu-Specific Dataset**: Curate labeled menu dataset of 50+ images for robust evaluation
2. **Vision-Language Models**: Fine-tune LayoutLMv3 or similar models on menu data
3. **Detection-Recognition Split**: Decouple OCR stages for independent error analysis
4. **Active Learning**: Incorporate user feedback for continuous improvement
5. **Multilingual Support**: Extend to non-English menus with language detection
6. **Confidence Calibration**: Provide calibrated confidence scores for downstream filtering

8 Conclusion

I presented a modular pipeline for structured menu extraction combining OCR with hybrid classification approaches. The key finding is that **domain mismatch** between receipt-trained models and menu images is the primary accuracy bottleneck—ML models achieving 91% accuracy on CORD-v2 drop to <30% F1 on menus.

8.1 Updated Results (v2.1)

After extensive improvements to the classification pipeline and ground truth data, the final system achieves:

Version	Precision	Recall	F1	Price Acc	Images
v1.0 Baseline	19.8%	44.0%	27.3%	19.9%	23
v2.0 Fixed GT	38.7%	62.4%	47.8%	31.7%	21
v2.1 Final	**75.8%**	**70.1%**	**72.8%**	34.8%	20

Table 6: Improvement trajectory across versions (491 total items)

Key improvements in v2.1 include OCR noise filtering, expanded keyword dictionaries (177 section + 123 group indicators), and corrected ground truth annotations.

Key contributions include:

1. **Empirical demonstration of domain mismatch** between receipt-trained models and menu extraction, with quantitative analysis
2. **Design principles for faithful extraction** that avoid hallucination through rule-based constraints
3. **Modular pipeline architecture** enabling component-level ablation and replacement
4. **Global bipartite matching** for price-item association replacing greedy heuristics
5. **Open evaluation framework** for menu extraction benchmarking

The findings highlight the importance of domain-specific approaches in document understanding and provide a foundation for future work in menu digitization.

References

[1] Smith, R. An overview of the Tesseract OCR engine. In *Proc. ICDAR*, 2007.

[2] Xu, Y., et al. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *Proc. KDD*, 2020.

[3] Huang, Y., et al. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In *Proc. ACM MM*, 2022.

[4] Park, S., et al. CORD: A Consolidated Receipt Dataset for Post-OCR Parsing. In *NeurIPS Document Intelligence Workshop*, 2019.

[5] Huang, Z., et al. ICDAR 2019 Competition on Scanned Receipt OCR and Information Extraction. In *Proc. ICDAR*, 2019.

[6] Baek, Y., et al. Character Region Awareness for Text Detection. In *Proc. CVPR*, 2019.

[7] Shi, B., Bai, X., Yao, C. An End-to-End Trainable Neural Network for Image-based Sequence Recognition. *IEEE TPAMI*, 2017.