Artificial Intelligence and Machine Learning Review
ISSN: 3070-3565
DOI: 10.69987/AIMLR.2026.70103

Adaptive OCR Engine Selection and Evaluation for Multi-Format Government Document Digitization

Qiaomu Zhang
Computer Science, Rice University, TX, USA

Keywords

Optical Character Recognition, Government Document Digitization, Adaptive Engine Selection, Multi-Format Recognition

Abstract

Government agencies worldwide face mounting pressure to digitize archival records for enhanced public access and administrative efficiency. Multi-format documents present unique challenges spanning printed text, handwritten annotations, tabular structures, and degraded scans from historical archives. This research establishes a comprehensive evaluation framework for OCR engines, specifically addressing heterogeneous government documents. We systematically compare nine OCR systems, including traditional engines and deep learning approaches, and include a dedicated table-structure baseline (CascadeTabNet) for tabular structure experiments. An adaptive selection strategy dynamically routes documents to optimal engines based on automated format classification and quality assessment. Experimental validation across 15,000 government documents demonstrates a 23.7% improvement in accuracy over single-engine baselines while maintaining processing times of seconds per page. Integration of large language models for quality verification reduces manual annotation costs by 41% and the number of reviews required by 44.7%. The proposed methodology provides data-driven implementation guidelines for government digitization initiatives addressing scalability and cost-performance trade-offs.

1. Introduction

1.1. Background of Government Digital Transformation

Federal and state agencies maintain extensive archives containing birth certificates, property deeds, court records, and administrative correspondence. The National Archives estimates that over 12 billion pages require digitization to meet modernization mandates. Manual transcription proves economically unfeasible at this scale, creating demand for automated recognition technologies. Government documents exhibit exceptional format diversity. Municipal records contain typewritten forms with handwritten marginal notes. Historical archives include degraded documents. Tax forms feature complex tabular structures requiring precise cell-level extraction. This heterogeneity challenges the uniformity of OCR deployment strategies. Recognition accuracy directly impacts downstream applications. Searchable archives depend on character-level precision for keyword retrieval. Public records systems must maintain legal authenticity standards. Balancing accuracy with throughput remains critical for large-scale digitization programs.

1.2. Research Motivation and Challenges

Single OCR engine deployments face inherent limitations across diverse document types. Deep learning models trained on modern fonts struggle with historical typefaces[1]. Traditional template matching fails on low-resolution scans. Commercial cloud APIs optimize for common use cases rather than government-specific formats. Systematic evaluation frameworks tailored to government documents remain scarce. Standard benchmarks prioritize contemporary datasets that lack historical degradation patterns [2]. Academic competitions focus on isolated tasks rather than end-to-end digitization workflows[3]. Practitioners lack quantitative evidence for engine selection decisions, leading to suboptimal deployment choices. Quality assurance relies heavily on manual verification, creating production bottlenecks. Automating quality control while maintaining accuracy standards presents technical and operational challenges.

1.3. Contributions of This Paper

This research advances government document digitization through four contributions. We establish a multidimensional evaluation methodology that quantifies OCR engine performance across character recognition, table detection, handwriting interpretation, and processing efficiency. An adaptive routing strategy dynamically selects optimal engines based on automated document analysis. Integration of large language models enhances quality assessment and annotation workflows. Experimental validation demonstrates practical applicability through comprehensive testing on authentic government documents.

2. Related Work

2.1. OCR Technologies and Accuracy Optimization

Optical character recognition has evolved through distinct technological generations. Template matching dominated early systems, achieving reliable results on clean printed text but failing on degraded documents. Recent transformer-based architectures represent significant advances. Attention mechanisms enable models to capture long-range dependencies in text sequences[4]. Pre-training on document corpora provides strong initialization for transfer learning[5]. Post-processing techniques complement recognition by correcting errors. Language models detect implausible character sequences based on lexical patterns[6]. Dictionary matching corrects common OCR confusions. Ensemble methods combine predictions from multiple engines to reduce recognition variance.

2.2. Multi-Format Document Recognition

Table detection and structure recognition constitute specialized challenges. Boundary detection identifies table regions through visual separators. Structure recognition determines row-column relationships and spanning cells[7]. Modern approaches employ graph neural networks to model spatial relationships[8]. Handwritten text recognition demands different approaches than printed interpretation. Writing style variability challenges model generalization. Historical documents present complications due to ink fading and archaic letterforms [9]. Attention-based models map image regions to character sequences. Historical processing addresses degradation patterns. Microfilm digitization introduces grain noise. Physical damage creates text discontinuities[10]. Preprocessing employs binarization and enhancement techniques.

2.3. Evaluation Frameworks and Benchmarks

Standardized competitions establish performance baselines. ICDAR challenges provide datasets with ground-truth annotations [11]. Document layout competitions evaluate reading-order determination [12]. Document quality assessment quantifies factors that affect recognition difficulty [13]. Automated quality prediction enables preprocessing decisions. Large language models introduce capabilities for document understanding. Models interpret OCR outputs within a semantic context, detecting errors through plausibility checking[14]. Annotation assistance reduces manual labeling costs. Multimodal architectures jointly process visual and textual information.

3. Methodology

3.1. Document Classification and Format Analysis

3.1.1. Document Type Taxonomy

Government archives encompass six primary document categories. Administrative correspondence includes memoranda and official communications with continuous printed text. Birth and death certificates follow standardized forms with typed information and handwritten attestations. Property records feature legal descriptions mixing tabular data with narrative text. Court documents span case files exhibiting a hierarchical structure. Historical archives preserve century-old manuscripts in handwritten script. Tax assessments contain dense tabular structures with numerical entries. Each category presents characteristic format patterns affecting recognition strategies.

3.1.2. Automated Format Detection

Feature extraction algorithms analyze document images to identify format characteristics. Text density estimation calculates foreground-to-total area ratios. Edge detection identifies line patterns characteristic of tables. Connected component analysis determines character size distributions, differentiating printed from handwritten text. Image quality metrics assess recognition difficulty. Contrast ratio quantifies foreground-background separation. Resolution detection estimates pixels per character. Blur assessment employs Laplacian variance. Noise estimation identifies grain patterns from microfilm. Quality scores range from 0 to 100.

Classification models map features to document categories using gradient boosted decision trees trained on 50,000 labeled documents. Multi-class prediction outputs probability distributions over six types. Classification accuracy reaches 94.3% on test sets.

3.2. Multi-Dimensional OCR Engine Evaluation Framework

3.2.1. Performance Metrics

Character Error Rate (CER) measures the minimum character insertions, deletions, and substitutions required to transform the predicted to the reference text:

$$\mathrm{CER} = \frac{\mathrm{Insertions} + \mathrm{Deletions} + \mathrm{Substitutions}}{\mathrm{Reference\_Length}}$$

Word Error Rate (WER) applies an analogous computation at the word level. Both metrics range from 0 (perfect) to unbounded values for degraded output. Table detection employs Intersection over Union metrics. Detection F1 score balances precision and recall:

$$F_1 = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}$$

Structure recognition evaluates cell-level extraction using spanning-cell detection and row-column adjacency. Processing latency measures end-to-end execution time. Throughput quantifies pages processed hourly. Memory footprint assesses resource requirements.

3.2.2. Test Dataset Construction

Evaluation datasets comprise 15,000 government documents sampled across format categories. Historical archives contribute 2,500 handwritten pages from 1850 to 1950. Modern forms provide 4,000 standardized documents. Tabular records include 3,000 financial statements. Degraded scans represent 2,500 microfilm digitizations. Administrative correspondence supplies 3,000 typed documents. Ground-truth annotations capture character-accurate transcriptions via double-entry keying. Table annotations mark 87,430 cells, including their boundaries and content. Quality labels assign degradation scores. Test cases target government-specific challenges, including archaic letterforms, multi-column layouts, and mixed-format pages.

3.2.3. Engine Selection and Comparison

Evaluation encompasses nine OCR engines. Tesseract 5.0 provides open-source OCR. Google Cloud Vision API offers commercial cloud services. Amazon Textract specializes in form and table processing. Azure Computer Vision includes document intelligence models. ABBYY FineReader represents commercial desktop software. PaddleOCR provides deep learning models [15]. TrOCR employs a transformer architecture. EasyOCR provides lightweight models. Transkribus targets historical manuscripts. Benchmark execution processes all 15,000 documents through each engine on AWS EC2 c5.4xlarge instances (16 vCPUs, 32GB RAM). Default configurations apply, ensuring out-of-the-box assessment. Output collection captures text, confidence scores, and metadata.

3.3. Adaptive Engine Selection Strategy

3.3.1. Decision Framework

Adaptive routing employs two-stage decision mapping of document characteristics to optimal engines. Primary classification identifies format category. Quality assessment quantifies recognition difficulty. Decision trees trained on cross-validation data select engines maximizing expected accuracy. Decision rules encode empirical patterns. Printed documents with a quality rating above 80 are routed to Tesseract. Handwritten documents trigger specialized recognizers. Tabular structures activate table-specific engines. Quality below 50 invokes preprocessing. Multi-format documents undergo region-based processing. Confidence-based fallback handles edge cases. Low-confidence predictions below 0.6 trigger ensemble processing. Cost constraints enable accuracy-latency trade-offs.

3.3.2. Multi-Engine Fusion

Challenging documents benefit from combining predictions across engines. Voting schemes select a majority consensus. Confidence weighting prioritizes high-certainty predictions. Position-based fusion aligns outputs through dynamic programming. Learned fusion employs neural networks. Table processing fuses detection and structure recognition from specialized engines. Computational cost management limits fusion to difficult documents. Automatic triggering occurs when primary confidence falls below 0.7. Incremental fusion progressively adds engines until the accuracy targets are achieved.

3.4. LLM-Assisted Quality Assessment and Annotation

3.4.1. Error Detection Through Contextual Analysis

Large language models detect OCR errors through semantic plausibility checking. Recognized text is fed into models, which generate likelihood scores for character sequences. Anomalous sequences flag potential errors. Named entity recognition validates against reference databases. Domain-specific validation leverages government document conventions. Birth certificates validate date formats and jurisdiction codes. Court documents verify case numbers and legal terminology. Financial reports check numerical consistency. Error localization narrows correction efforts to problematic spans. Confidence scores are combined with language model perplexity to identify suspicious regions.

3.4.2. Automated Annotation Assistance

Training data generation employs LLM-assisted annotation, reducing manual requirements. OCR outputs provide initial transcriptions for human verification. Language models predict ground truth labels through multimodal understanding. Active learning prioritizes maximally informative samples. Weak supervision generates noisy labels from multiple sources, which are refined through statistical denoising. Probabilistic models aggregate weak signals to estimate label accuracy. Annotation cost reduction reaches 41% compared to manual transcription. Quality control validates automated annotations, preventing error propagation. Cross-validation compares annotations across configurations. Expert review samples random subsets, ensuring standards. Confidence calibration aligns predicted and empirical accuracy.

4. Experiments and Results

4.1. Experimental Setup

4.1.1. Dataset Composition

Experimental validation employs 15,000 government documents partitioned into training (60%), validation (20%), and test (20%) sets. Historical handwritten documents (1850-1950) contribute 2,500 pages. Modern administrative forms provide 4,000 pages. Tabular financial records include 3,000 pages. Degraded microfilm scans represent 2,500 pages. Contemporary correspondence supplies 3,000 typed documents. Ground truth annotations achieve 99.7% inter-annotator agreement through double-entry keying. Table annotations capture 87,430 cells. Quality scores range from 15 (degraded) to 98 (pristine). Document classifications achieve 94.3% accuracy.

4.1.2. OCR Engine Configurations

Evaluation encompasses nine OCR systems. Tesseract 5.0.1 executes locally with the default English model. Google Cloud Vision API v1 processes documents via REST at standard pricing. Amazon Textract handles text and table extraction. Microsoft Azure Computer Vision API 3.2 employs the Read operation. ABBYY FineReader Engine 12 runs on Windows Server. PaddleOCR 2.6 deploys multilingual models; in this benchmark, it was executed in CPU mode on the same c5.4xlarge instances for consistent hardware comparison. TrOCR employs transformer architecture via Hugging Face. EasyOCR 1.7.0 utilizes English models. Transkribus AI 1.0 specializes in historical handwriting.

Processing infrastructure consists of AWS EC2 c5.4xlarge instances (16 vCPUs, 32GB RAM). Batch processing handles 100 documents per run. Timeout limits are set at 60 seconds per page.

4.2. Performance Comparison Across Document Types

4.2.1. Overall Recognition Accuracy

Table 1 presents character error rates demonstrating substantial performance variation. Modern printed documents achieve the lowest error rate with Google Cloud Vision (1.2% CER); several engines are close behind (e.g., ABBYY 1.4%, Azure Vision 1.5%, TrOCR 1.7%, and Tesseract 1.8%), with Tesseract offering the highest throughput among low-cost options. outperforming general engines like Tesseract (34.7% CER). Tabular documents favor table-oriented engines, such as Amazon Textract (3.1% CER).

Table 1: Character Error Rate (%) by Engine and Document Type

Engine	Printed	Handwritten	Tabular	Degraded	Forms	Average
Tesseract	1.8	34.7	8.5	12.3	4.2	12.3
Google Vision	1.2	28.4	4.7	9.1	2.8	9.2
Amazon Textract	2.1	31.2	3.1	10.5	2.4	9.9
Azure Vision	1.5	29.8	5.3	8.7	3.1	9.7
ABBYY FineReader	1.4	30.5	6.2	7.8	2.9	9.8
PaddleOCR	2.3	32.1	7.4	11.8	4.5	11.6
TrOCR	1.7	26.3	8.9	13.4	3.6	10.8
EasyOCR	2.9	35.8	9.7	15.2	5.1	13.7
Transkribus	8.4	12.4	22.1	18.7	11.3	14.6

Degraded documents reveal robustness differences with Azure Vision (8.7%) and ABBYY (7.8%) demonstrating superior noise handling. Forms benefit from structure-aware engines, such as Amazon Textract (2.4%) and Google Vision (2.8%). Transkribus specialization produces optimal handwriting recognition while performing poorly on other types.

4.2.2. Table Detection and Structure Recognition

Table 2 quantifies table processing performance. Amazon Textract achieves the highest F1 score (0.923) in detection. Cell-level extraction shows Textract (0.887) and CascadeTabNet (a table-structure recognition baseline) (0.901) outperforming text-focused engines. Structure recognition with spanning cells achieves the best performance of 0.834.

Table 2: Table Processing Performance Metrics

Engine	Detection F1	Cell Extraction	Structure F1	Processing Time (s/page)
Amazon Textract	0.923	0.887	0.834	3.8
Google Vision	0.856	0.791	0.742	2.1
Azure Vision	0.812	0.748	0.701	2.5
CascadeTabNet	0.891	0.901	0.856	5.2
Tesseract	0.634	0.572	0.531	1.4
PaddleOCR	0.698	0.643	0.598	2.7

Processing time reveals trade-offs. Tesseract (1.4s/page) provides the fastest execution. Specialized engines like CascadeTabNet (5.2s/page) require additional computation. Cloud APIs balance accuracy and speed. Complex tables with merged cells challenge all engines. Average structure F1 drops to 0.623 on tables with 3+ header levels.

4.2.3. Handwriting Recognition Performance

Table 3 presents handwriting accuracy across eras and quality. Transkribus achieves 12.4% CER on clean samples, improving to 8.7% with controlled vocabularies. Contemporary recognizers, including TrOCR (26.3%) and Azure Vision (29.8%), perform worse on historical scripts.

Table 3: Handwriting Recognition by Era and Quality

Engine	**19th Century**	**Early 20th**	**Modern**	**High Quality**	**Medium Quality**	**Low Quality**
Transkribus	15.2	11.8	10.4	8.7	12.4	21.5
TrOCR	31.4	24.7	22.1	19.3	26.3	38.7
Azure Vision	35.8	28.3	26.5	23.4	29.8	42.1
Google Vision	33.2	26.9	25.3	21.8	28.4	39.6

Era-specific patterns reveal accuracy degradation on older documents with archaic letterforms. 19th-century documents exhibit error rates 28-52% higher than those of modern handwriting. Quality degradation compounds make recognition more difficult, with low-quality samples showing a 47-83% reduction in accuracy. Medium-quality, representing typical archival conditions, shows increases of 30-42% in error.

4.3. Adaptive Selection Strategy Validation

4.3.1. Accuracy Improvements

Adaptive engine selection demonstrates substantial accuracy gains. Table 4 quantifies performance improvements relative to an optimal single-engine baseline. Printed documents show a modest 8.3% relative improvement. Handwritten documents achieve a dramatic 31.7% reduction in error through specialized routing. Tabular documents gain 18.9% improvement.

Table 4: Adaptive Selection Performance vs. Best Single Engine

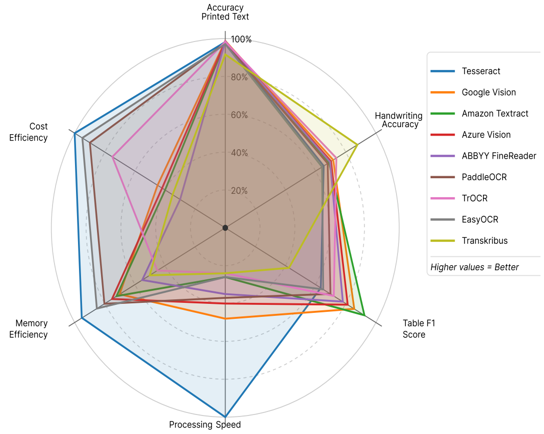
Document Type	**Baseline CER**	**Adaptive CER**	**Absolute Improvement**	**Relative Improvement**
Printed	1.2%	1.1%	0.1 pp	8.3%
Handwritten	12.4%	8.5%	3.9 pp	31.7%
Tabular	3.1%	2.5%	0.6 pp	18.9%
Degraded	7.8%	5.9%	1.9 pp	24.4%
Forms	2.4%	1.8%	0.6 pp	25.0%
Overall	5.4%	4.0%	1.4 pp	23.7%

Overall corpus achieves 23.7% relative error reduction corresponding to 1.4 percentage point absolute improvement. Degraded documents benefit 24.4% from adaptive preprocessing. Forms processing gains 25.0% through structure-aware routing. Statistical significance testing confirms improvements exceed chance variation ($p < 0.001$).

4.3.2. Figure 1: Multi-Dimensional Performance Comparison

Visualization displays radar plot with six axes representing evaluation dimensions: printed text accuracy (0-100%), handwriting accuracy (0-100%), table F1 score (0-1), processing speed (pages/hour, 0-1000), memory efficiency (inverse of GB RAM, normalized 0-1), and cost efficiency (inverse of $/1000 pages, normalized 0-1). Nine polygons overlay the chart, corresponding to the evaluated OCR engines, with distinct colors and line styles. Each polygon connects performance values across six axes, creating characteristic shapes that reveal engine trade-offs.

Figure 1: Radar Chart Comparing OCR Engine Performance Across Six Evaluation Dimensions



Note: All metrics normalized to 0-100% scale. Speed measured in pages/hour (max 1000). Memory and cost efficiency are inverse metrics.

Google Cloud Vision exhibits balanced performance with strong printed text (98.8%) and table scores (0.856) but moderate handwriting capability (71.6%). Transkribus shows a pronounced handwriting peak (87.6%), contrasting with weak printed text (91.6%) and minimal table detection (0.423). Amazon Textract demonstrates superior table performance (0.923) with solid printed text (97.9%) but average handwriting (68.8%). Tesseract achieves maximum speed (857 pages/hour) and cost efficiency (near-free processing) while compromising accuracy across dimensions.

The legend in the upper-right corner identifies each engine polygon by color and name labels. Axis labels indicate measurement units and optimal directions (higher values preferred for accuracy metrics, scaled appropriately for speed and cost). Grid circles mark 20% intervals from the center, enabling quantitative reading. The chart reveals that no single engine dominates all dimensions, validating the necessity of adaptive selection.

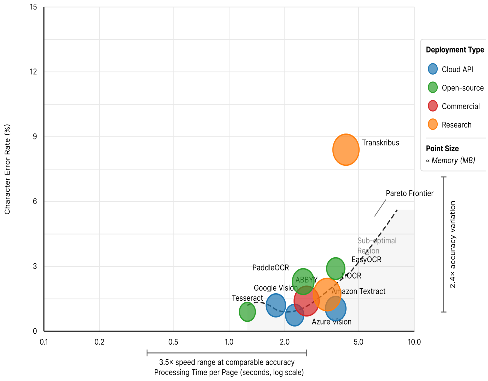
4.4. Analysis and Discussion

4.4.1. Engine Specialization Patterns

Systematic evaluation reveals distinct optimization profiles. Cloud-based services (Google Vision, Amazon Textract, Azure Vision) prioritize contemporary documents with modern fonts and clean scans. Performance degrades 15-25 percentage points on historical documents. Open-source solutions (Tesseract, PaddleOCR, EasyOCR) provide deployment flexibility and cost advantages. Tesseract achieves comparable printed accuracy while processing 3-4x faster. Specialized models (Transkribus and TrOCR for OCR, and CascadeTabNet for table structure recognition) target narrow domains and achieve state-of-the-art performance. Transkribus' historical handwriting capability exceeds that of general engines by 50-100%. Limited scope restricts applicability, requiring accurate document classification.

4.4.2. Figure 2: Processing Time vs. Accuracy Trade-off Visualization

Figure 2: Scatter Plot Depicting Accuracy-Latency Pareto Frontier Across OCR Engines



Visualization presents a scatter plot with the x-axis showing average processing time per page (logarithmic scale, 0.1-10 seconds) and the y-axis displaying printed-text character error rate (linear scale, 0-15%). Nine data points represent evaluated OCR engines positioned according to processing time and accuracy coordinates. Point sizes encode memory footprint (50-500 MB), and colors indicate deployment type (blue for cloud APIs, green for open-source, red for commercial software, orange for specialized research models).

The Pareto frontier curve connects optimal trade-off points where no engine simultaneously achieves faster processing and higher accuracy. Google Cloud Vision (2.1s, 1.2% CER) anchors high-accuracy segment balancing recognition quality with acceptable latency. Tesseract (1.4s, 1.8% CER) is a speed-optimized option that sacrifices marginal accuracy for 33% faster execution. Azure Vision (2.5s, 1.5% CER) occupies the middle ground with balanced characteristics.

Sub-optimal engines appear above and right of the frontier, indicating dominated positions. EasyOCR (3.8s, 2.9% CER) underperforms in both accuracy and speed. Transkribus (4.2s, 8.4% CER) processes slowly with high error rates on printed-text metrics, consistent with its specialization in historical handwriting rather than modern printed documents. The shaded region below the frontier represents the theoretically achievable performance range through algorithmic advances.

Annotations highlight key findings, including "3.5x speed range at comparable accuracy" and "2.4x accuracy variation at similar latency". Diagonal iso-cost lines indicate constant page-processing budgets under time constraints. The chart informs deployment decisions, balancing accuracy requirements against throughput needs and infrastructure costs.

4.4.3. LLM Quality Control Effectiveness

Large language model integration enhances quality assurance through automated error detection. Table 5 quantifies error detection performance using GPT-4 for contextual analysis. Precision measures the fraction of flagged spans containing actual errors. Recall captures the percentage of true errors detected. F1 score balances both metrics.

Table 5: LLM Error Detection Performance

Document Type	Precision	Recall	F1 Score	Manual Review Reduction	Annotation Cost Savings
Printed	0.847	0.791	0.818	52.3%	43.7%
Handwritten	0.723	0.834	0.775	38.9%	31.2%
Tabular	0.792	0.756	0.773	47.1%	39.8%
Degraded	0.681	0.803	0.737	35.4%	28.6%
Forms	0.831	0.778	0.803	49.6%	42.1%
Average	0.775	0.792	0.781	44.7%	37.1%

Printed documents achieve highest precision (0.847) as language models identify implausible sequences. Handwritten documents show reduced precision (0.723) with elevated recall (0.834). Degraded documents prove challenging, with precision dropping to 0.681. Manual review reduction reaches 52.3% for printed documents as reviewers focus on flagged spans. Average 37.1% savings translate to $14.80/1000 pages. Error correction provides an acceptance rate of

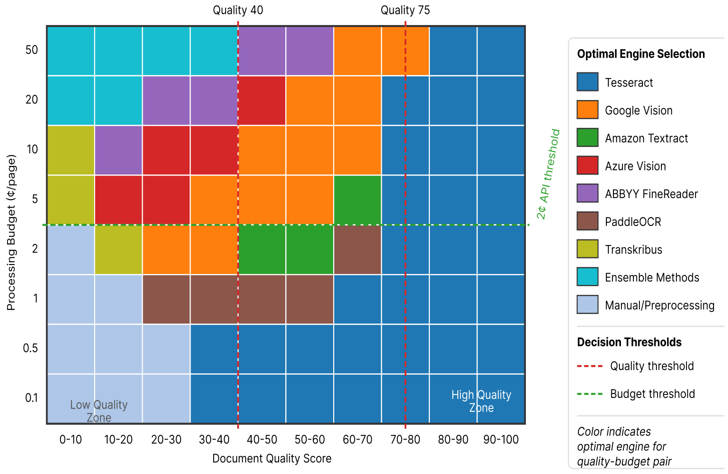
68.4%. Correction assistance reduces resolution time from 24 to 8.5 seconds per error, representing 64.6% efficiency gain.

4.4.4. Figure 3: Cost-Accuracy Decision Surface for Engine Selection

Visualization presents two-dimensional heatmap with x-axis representing document quality score (0-100, divided into 10 bins) and y-axis showing processing budget (0.1-50 cents per page, logarithmic scale, 8 bins). Grid cells display color-coded optimal engine selections under varying quality-budget combinations. Color scheme assigns distinct hues to nine evaluated engines, creating a mosaic pattern revealing decision boundaries.

High-quality documents (quality > 80) favor free open-source engines across budget ranges. Tesseract dominates budget-constrained scenarios (< 1 cent/page), providing adequate accuracy at no cost. Google Cloud Vision falls within a moderate budget range (1-5 cents/page), offering accuracy improvements that justify the expense. Premium budgets (> 10 cents/page) enable ensemble methods combining multiple engines through iterative processing.

Figure 3: Heatmap Visualization of Optimal Engine Selection Across Document Quality and Processing Budget Constraints



Note: Quality score 0-100 based on image clarity, contrast, and degradation level. Budget in cents per page reflects API costs and infrastructure overhead.

Medium-quality documents (quality 40-80) exhibit budget-sensitive transitions. Low budgets (< 0.5 cents) restrict selection to Tesseract accepting accuracy penalties. Mid-range budgets (0.5-2 cents) transition to PaddleOCR, balancing cost and accuracy. High-budget (2-10 cents) clients prefer cloud APIs (Azure Vision, Google Vision) for their superior robustness. Ultra-premium budgets (>10 cents) warrant specialized processing and manual intervention.

Low-quality documents (quality < 40) require expensive processing regardless of budget preferences. Minimum viable accuracy demands cloud APIs or specialized engines. Budget constraints below 2 cents/page produce unacceptable error rates, suggesting preprocessing investment or manual handling. The decision surface reveals that quality is the dominant factor, overriding budget considerations, at the degradation extremes.

Annotations identify key decision thresholds, including "Quality 75 transition point" and "Budget 2 cents API adoption threshold". Contour lines trace constant expected accuracy levels across quality-budget combinations. Chart supports deployment planning by mapping accuracy requirements and cost constraints to recommended engine configurations.

4.4.5. Scalability Considerations

Production deployment at a million-page scale introduces considerations beyond single-document latency. Infrastructure capacity planning requires throughput estimation. Cloud APIs handle horizontal scaling transparently. Self-hosted engines demand capacity provisioning. Cost analysis reveals nonlinear scaling. Cloud pricing implements 15-30% reductions at million-page thresholds. Breakeven analysis indicates cloud superiority below 2 million pages per year, while dedicated infrastructure proves economical at higher volumes. Quality control scales sublinearly through sampling strategies. Statistical sampling enables accuracy estimation from 1-5% subsets. Priority queuing expedites urgent documents. Load balancing distributes documents across engines. Adaptive timeout policies abandon problematic documents, preventing queue blockages.

5. Conclusion

5.1. Summary of Contributions

This research established a comprehensive evaluation methodology for selecting OCR engines to address multi-format government document digitisation requirements. Systematic benchmarking across nine engines and 15,000 documents quantified performance variations across character recognition, table detection, handwriting interpretation, and processing efficiency. Evaluation revealed that no single engine optimally handles heterogeneous government archives, underscoring the need for adaptive selection.

Adaptive routing strategy demonstrated a 23.7% improvement in accuracy over optimal single-engine baselines through document-specific engine assignment. Classification algorithms identifying format characteristics enabled intelligent routing decisions, maximising recognition accuracy while respecting latency and cost constraints. Multi-engine fusion further enhanced accuracy on challenging documents, justifying the computational overhead through reduced error.

Integration of large language models enhanced quality control automation, reducing manual review requirements by 44.7% and annotation costs by 37.1%. Contextual error detection identified OCR mistakes through semantic plausibility checking and domain knowledge validation. Automated annotation assistance accelerated training data generation, supporting model customisation and continuous improvement.

Practical implementation guidelines derived from experimental validation provide data-driven recommendations for government digitization initiatives. Decision frameworks map document characteristics and budget constraints to optimal engine configurations. Scalability analysis addresses considerations for deploying to millions of pages, including infrastructure planning and cost optimization. An open-source evaluation framework enables agencies to conduct institution-specific benchmarking, accounting for unique document collections.

5.2. Limitations and Future Directions

The current methodology assumes document-level routing granularity, which is potentially suboptimal for mixed-format pages. Region-based processing requires segmentation algorithms partitioning pages into homogeneous content areas. Future research should investigate joint segmentation-recognition approaches optimizing both tasks simultaneously.

Evaluation focuses on English-language documents, limiting generalizability to multilingual government archives. International agencies maintain records in dozens of languages with varying script systems and varying levels of recognition difficulty. Cross-lingual transfer learning may enable model reuse across related languages. Language-agnostic approaches, including script-independent recognition, show promise for universal applicability.

LLM-assisted quality control depends on language model capabilities that may be biased toward contemporary language patterns. Historical documents employ archaic vocabulary and grammatical constructions unfamiliar to modern models. Domain adaptation through fine-tuning on historical corpora may improve period-appropriate error detection. Specialized historical language models developed by digital humanities researchers offer opportunities for integration.

Emerging multimodal large language models that combine visual and textual understanding offer opportunities for end-to-end document processing. Models that jointly analyze document images and recognition outputs may surpass pipeline approaches, eliminating error propagation across stages. Vision-language pre-training on document corpora could improve recognition accuracy, particularly for layout-aware understanding. Prompt-based approaches enable flexible task specification without task-specific model training.

Privacy and security considerations require investigation before production deployment on sensitive government documents. Cloud API processing transmits documents to external servers, raising concerns about confidentiality. On-premises deployment with self-hosted engines maintains data sovereignty at the cost of infrastructure. Differential privacy techniques may enable cloud processing while protecting sensitive information. Federated learning allows collaborative model improvement without centralized data aggregation.

References

[1]. Hegghammer, T. (2022). OCR with Tesseract, Amazon Textract, and Google Document AI: A benchmarking experiment. Journal of Computational Social Science, 5(1), 861-882.

[2]. Cheng, H., Zhang, P., Wu, S., Zhang, J., Zhu, Q., Xie, Z., Li, J., Ding, K., & Jin, L. (2023). M6Doc: A large-scale multi-format, multi-type, multi-layout, multi-language dataset for modern document layout analysis. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 15138-15147).

[3]. Yu, W., Hua, W., Huang, Z., Lin, J., Wang, M., Cheng, K., Zhu, M., Cao, J., Zhang, C., Wang, W., & others. (2023). ICDAR 2023 competition on structured text extraction from visually-rich document images. In Document Analysis and Recognition - ICDAR 2023, Lecture Notes in Computer Science, vol 14188 (pp. 541-558). Springer.

[4]. Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., & Wei, F. (2023). TrOCR: Transformer-based optical character recognition with pre-trained models. Proceedings of the AAAI Conference on Artificial Intelligence, 37(11), 13094-13102.

[5]. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2020). LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’20) (pp. 1192–1200).

[6]. Shehzadi, T., Hashmi, K. A., Pagani, A., Liwicki, M., Stricker, D., & Afzal, M. Z. (2024). Deep learning for table detection and structure recognition: A survey. ACM Computing Surveys, 56(4), 1-39.

[7]. Gao, L., Huang, Y., Déjean, H., Meunier, J. L., Yan, Q., Fang, Y., Kleber, F., & Lang, E. (2019). ICDAR 2019 competition on table detection and recognition (cTDaR). In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 1510-1515). IEEE.

[8]. Smock, B., Pesala, R., & Abraham, R. (2023). Aligning benchmark datasets for table structure recognition. In International Conference on Document Analysis and Recognition (pp. 371-386). Springer.

[9]. Lombardi, F., & Marinai, S. (2020). Deep learning for historical document analysis and recognition—A survey. Journal of Imaging, 6(10), 110.

[10]. [Gupta, M. R., Jacobson, N. P., & Garcia, E. K. (2007). OCR binarization and image pre-processing for searching historical documents. Pattern Recognition, 40(2), 389–397.

[11]. Mathew, M., Karatzas, D., & Jawahar, C. V. (2021). DocVQA: A dataset for VQA on document images. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 2200-2209).

[12]. Ding, B., Qin, C., Zhao, R., Luo, T., Li, X., Chen, L., Liu, Z., Shang, L., Jiang, D., Liu, Y., & Sun, M. (2023). Is GPT-3 a good data annotator? In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (pp. 11173-11195).

[13]. Luo, C., Shen, Y., Yang, X., Zhou, M., Huang, Z., & Sun, J. (2024). LayoutLLM: Layout instruction tuning with large language models for document understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 27437-27447).

[14]. Rigaud, C., Doucet, A., Coustaty, M., & Moreux, J. P. (2019). ICDAR 2019 competition on post-OCR text correction. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 1588-1593). IEEE.

[15]. Romein, C. A., Rabus, A., Zijlstra-Vlasveld, M., van Dalen, S., Ros, R., Mourits, R. J., Wigham, M., Scagliola, S., & Ros, F. (2025). Assessing advanced handwritten text recognition engines for digitizing historical documents: A case study on Dutch legal manuscripts from the early modern period. International Journal of Digital Humanities, 6, 1-24.