# Document Digitization Pipeline: Comparative Analysis of OCR Approaches

## 1. Introduction

The task is to develop robust and efficient OCR solutions capable of converting various document types—including handwritten forms, handwritten receipts, printed receipts, and printed forms—into structured, machine-readable data. The objective is to research, compare, and implement OCR approaches that balance accuracy and computational efficiency for effective document digitization.

## 2. Datasets

Research focused on two broad dataset categories: Printed and Handwritten documents.

- **Printed Dataset:** Includes two categories—Forms and Receipts—sourced from publicly available datasets. Printed data was used to complement our study due to the scarcity of annotated handwritten datasets.
- **Handwritten Dataset:** Since publicly available annotated handwritten data is limited, we manually created and annotated our own handwritten dataset to ensure accurate evaluation on this challenging document type.

| Dataset Type | Description | Number of Images | Source |
|---|---|---|---|
| Handwritten Forms | Hand-annotated handwritten forms | 5 | Manually created & labelled |
| Handwritten Receipts | Hand-annotated handwritten receipts | 5 | Manually created & labelled |
| Printed Receipts | Printed receipts | 20 | https://www.kaggle.com/datasets/trainingdatapro/ocr-receipts-text-detection |
| Printed Forms | Printed structured forms | 50 | - |

## 3. Models Used

For this study, we evaluated a range of OCR and vision-language models across three main categories to cover diverse approaches in document digitization.

**1st Category: Large Language Models (LLMs) with Vision Capabilities**

- Meta Llama 3.2 Vision: A large-scale vision-language model that requires substantial GPU resources, typically 15GB or more.
- Qwen-2 VL: Comparable to Llama 3.2, this vision-language model also demands high GPU memory (≥15GB) for effective inference.
- Microsoft Florence-2: A vision-language model with a more moderate GPU memory requirement of approximately 5GB, balancing performance and resource usage.

**2nd Category: Traditional OCR Systems**

- PaddleOCR: An open-source OCR framework optimized for CPU execution, eliminating the need for GPU acceleration.

**3rd Category: Cloud-Based OCR APIs**

- Google Gemini Vision API: A cloud-hosted OCR service that offloads computation to remote servers, requiring no local hardware resources.

## 4. Methodology

### 4.1 Experimental Setup

- Each model evaluated on all datasets.
- Metrics: ROUGE-1, ROUGE-2, ROUGE-L F1 scores for accuracy; inference time per image; GPU/CPU memory usage.

### 4.2 Evaluation Metrics

- **ROUGE scores:** Measure textual overlap accuracy.
- **Inference Time:** Efficiency per image processed.
- **Computational Resources:** GPU memory footprint and CPU/GPU utilization.

## 5. Results and Analysis

### 5.1 Handwritten Forms

| Model | ROUGE-1 F1 | ROUGE-2 F1 | ROUGE-L F1 | Inference Time (s) | GPU Memory (MB) | GPU Required? |
|---|---|---|---|---|---|---|
| Gemini | 0.9333 | 0.8951 | 0.9319 | 4.21 | N/A | No |
| PaddleOCR | 0.7795 | 0.6550 | 0.7705 | 2.14 | N/A | No |
| Llama 3.2 | 0.6962 | 0.5330 | 0.6471 | 33.83 | ~14,084 | Yes |
| Florence-2 | 0.4698 | 0.3696 | 0.4698 | 2.21 | ~4,926 | Yes |
| Qwen-2 | 0.7634 | 0.7269 | 0.7634 | 46.49 | ~7,547 | Yes |

### 5.2 Handwritten Receipts

| Model | ROUGE-1 F1 | ROUGE-2 F1 | ROUGE-L F1 | Inference Time (s) | GPU Memory (MB) | GPU Required? |
|---|---|---|---|---|---|---|
| Gemini | 0.8782 | 0.7502 | 0.7770 | 3.95 | N/A | No |
| PaddleOCR | 0.5962 | 0.3689 | 0.5300 | 1.43 | N/A | No |
| Llama 3.2 | 0.4054 | 0.2140 | 0.3079 | 14.15 | ~14,084 | Yes |
| Florence-2 | 0.1890 | 0.0989 | 0.1827 | 0.90 | ~4,926 | Yes |
| Qwen-2 | 0.5951 | 0.4614 | 0.5407 | 38.67 | ~7,547 | Yes |

### 5.3 Printed Receipts

| Model | ROUGE-1 F1 | ROUGE-2 F1 | ROUGE-L F1 | Inference Time (s) | GPU Memory (MB) | GPU Required? |
|---|---|---|---|---|---|---|
| Gemini | 0.4867 | 0.3748 | 0.2456 | 5.60 | N/A | No |
| PaddleOCR | 0.3517 | 0.1837 | 0.1758 | 2.10 | N/A | No |
| Llama 3.2 | 0.3427 | 0.1877 | 0.1767 | 32.77 | ~14,084 | Yes |

| Model | ROUGE-1 F1 | ROUGE-2 F1 | ROUGE-L F1 | Inference Time (s) | GPU Memory (MB) | GPU Required? |
|---|---|---|---|---|---|---|
| Florence-2 | 0.1936 | 0.0532 | 0.1137 | 2.23 | ~4,926 | Yes |
| Qwen-2 | 0.3071 | 0.2179 | 0.1639 | 45.22 | ~7,547 | Yes |

### 5.4 Printed Forms

| Model | ROUGE-1 F1 | ROUGE-2 F1 | ROUGE-L F1 | Inference Time (s) | GPU Memory (MB) | GPU Required? |
|---|---|---|---|---|---|---|
| Gemini | 0.9701 | 0.7361 | 0.6852 | 4.08 | N/A | No |
| PaddleOCR | 0.8821 | 0.6221 | 0.6271 | 3.65 | N/A | No |
| Llama 3.2 | 0.0159 | 0.0000 | 0.0144 | 23.11 | ~14,084 | Yes |
| Florence-2 | 0.6014 | 0.3480 | 0.4872 | 8.13 | ~4,926 | Yes |
| Qwen-2 | 0.8256 | 0.6285 | 0.6079 | 34.31 | ~7,547 | Yes |

---

## 6. Computational Resource Analysis

- **Model Size vs. GPU Memory Usage**:
  - Florence-2: ~5GB model size, uses ~4.9GB GPU memory during inference.
  - Llama 3.2 Vision: ~15GB model size, ~14GB GPU memory usage.
  - Qwen-2 VL: ~15GB model size, uses around ~7.5GB GPU memory (indicating possible memory optimization).
- **Inference Speed**:
  - PaddleOCR (CPU-only) fastest (1.4 - 3 seconds).
  - Gemini API (cloud) fast and does not require local hardware.
  - Larger LLMs (Llama 3.2, Qwen-2) significantly slower, 14-46 seconds per image.
  - Florence-2 moderate performance, 2-8 seconds inference time.

## 7. Consolidated Graphs

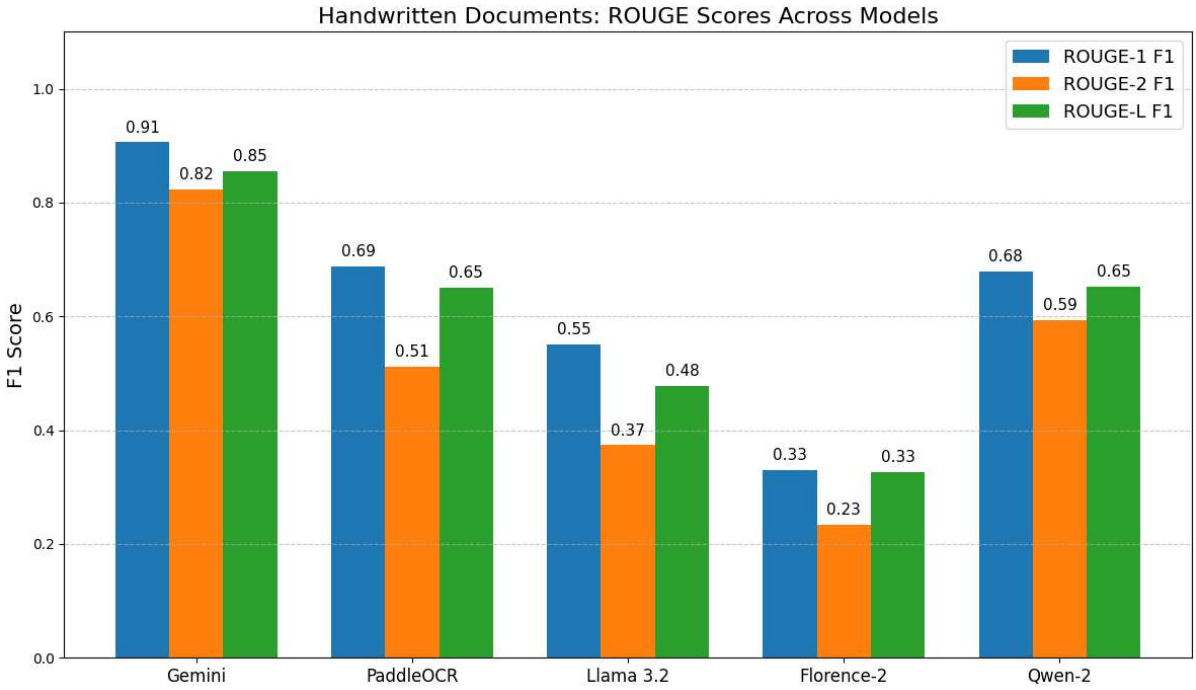


*Fig 1 : Handwritten Documents Metrics*

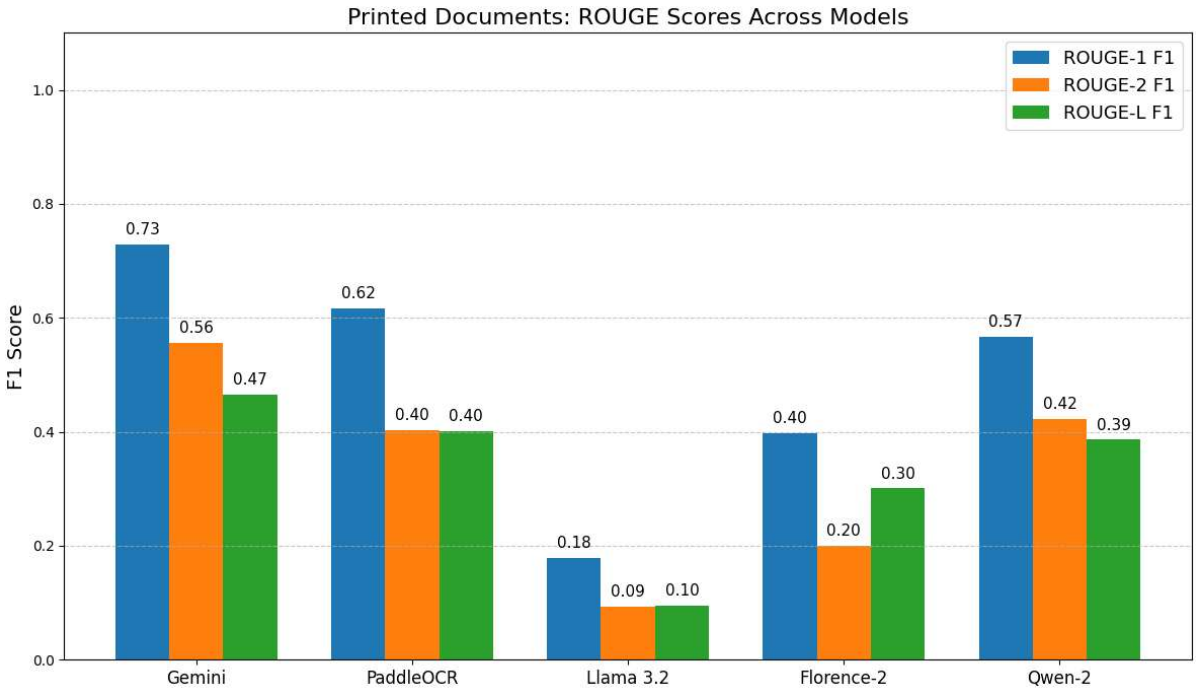


*Fig 2 : Printed Documents Metrics*

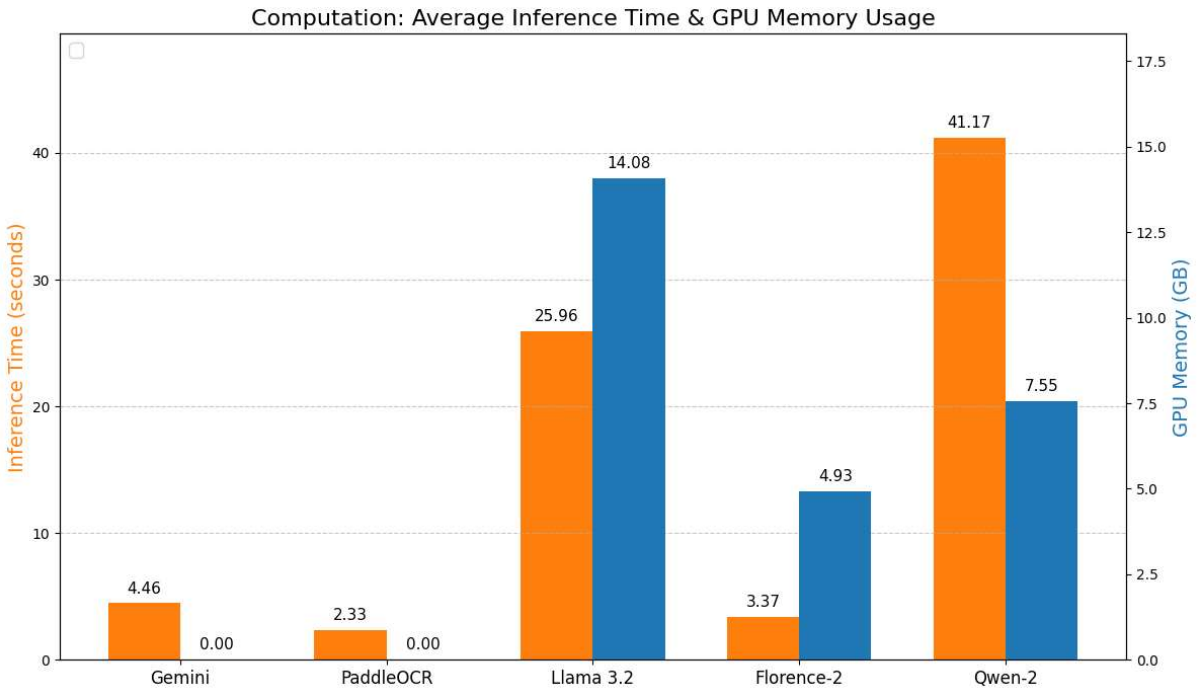


*Fig 3 : Computation Comparison*

## 8. Discussion and Recommendations

- **Best Accuracy**: Gemini Vision API consistently highest ROUGE scores across datasets without requiring GPU or heavy local compute.
- **Lightweight Option**: PaddleOCR is excellent for offline use, fast, and requires no GPU, albeit with slightly reduced accuracy.
- **Large Model Limitations**: Llama 3.2 and Qwen-2 require expensive hardware, with slower inference and only marginally better or worse accuracy depending on dataset.
- **Florence-2**: Moderate GPU usage and speed, performs well on printed forms but struggles with receipts.
- **Hybrid Deployment**: Use PaddleOCR as primary local OCR; fallback to Gemini API for complex cases requiring higher accuracy.

---

## 9. Future Work

- Although this research directly utilized pre-existing OCR and vision-language models, future work will explore fine-tuning these models on domain-specific handwritten and printed document layouts for enhanced accuracy.
- Investigate transformer-based OCR architectures such as TrOCR that specialize in handwritten text recognition.

- Explore integration of multimodal large language models like GPT-4 Vision for richer semantic understanding beyond text extraction.
- Focus on optimizing computational efficiency, including GPU memory usage and inference speed, to improve deployment feasibility.

## 10. Conclusion

- Google Gemini Vision API consistently achieved the highest accuracy across all document types, especially on handwritten forms and receipts.
- PaddleOCR demonstrated strong performance for printed documents and offered fast inference without GPU requirements, making it suitable for offline use.
- Large vision-language models like Llama 3.2 and Qwen-2 VL required significant GPU resources and had slower inference times, limiting practical deployment.
- Microsoft Florence-2 showed moderate accuracy and computational demands, performing better on printed forms than receipts.
- In terms of computational efficiency, PaddleOCR and Google Gemini API were the most practical choices, balancing speed and resource usage.