# Computer Vision for Financial Statement Analysis


## Abstract

*Financial documents such as invoices, bank statements, and reports contain rich visual and textual information that must be interpreted together to extract meaningful insights. We present a multimodal approach for visual understanding of financial document semantics by combining convolutional neural networks (CNNs), optical character recognition (OCR), and vision-language models (VLMs) within an integrated pipeline. The proposed system first analyzes document images to detect structural and layout features, then leverages OCR to extract textual content. A multimodal VLM fuses visual and textual representations to interpret the document's semantic content, and a specialized financial language model (FinGPT) is integrated to provide domain-specific understanding and generation capabilities. We evaluate our approach on diverse tasks – including document classification, key information extraction, question answering, and financial report summarization – demonstrating significant performance gains over baseline OCR-only and single-modality methods. Our pipeline achieves state-of-the-art or competitive results on benchmark datasets (e.g., up to 8 percent improvement in F1 for form understanding and higher ROUGE scores in summarization), and is capable of handling complex layouts and multi-page documents common in finance. We further conduct ablation studies to quantify the contribution of each component and provide an error analysis highlighting common failure modes, guiding future work in robust multimodal financial document analysis.*


## 1. Introduction

In my work, I focus on automating the extraction and understanding of information from financial documents such as invoices, bank statements, contracts, and annual reports. These documents are ubiquitous in business processes, and automating their interpretation can dramatically reduce manual labor and the risk of human error. However, developing machine comprehension systems for such documents presents serious challenges: scanned formats often contain noisy or distorted text that leads to OCR errors, and the layouts can be highly complex — including tables, multi-column text, and embedded figures — requiring reasoning over both visual structure and content.

These challenges are especially pronounced in the financial domain. Many public datasets for document AI focus on short, single-page documents, whereas financial reports are typically long, multi-page, and rich in domain-specific semantics. Most models are limited by context length (often 512 tokens), making it difficult to extract insights from documents that span dozens or hundreds of pages. In practice, analyzing financial reports requires a global understanding of the document's layout, relationships between sections, and the ability to interpret complex language and visual content together.

To address these limitations, I draw on recent progress in both datasets and modeling techniques for document understanding. Benchmarks now exist for document classification, form understanding, and document question answering. Yet even with these advances, a clear gap remains in handling domain-specific, long-form financial content. Vision-language models (VLMs) and document transformers have improved general-purpose document comprehension, but they are often trained on non-financial data and fail to capture domain-specific nuances.

Meanwhile, the NLP community has developed large language models specialized in finance — such as BloombergGPT and FinGPT — which outperform general LLMs on financial text. However, these models are typically limited to text inputs and lack visual processing capabilities. My goal is to bridge this divide by building a multimodal CNN-VLM pipeline tailored to the financial domain. My approach combines a CNN-based visual layout encoder, an OCR module for textual content extraction, and a fusion module that integrates visual and textual features. To enable domain-specific reasoning, I integrate FinGPT into the pipeline, allowing the system to not only read financial documents but also interpret their meaning with financial expertise.

I evaluate this pipeline across multiple tasks: document classification, key information extraction, document QA, and long-form summarization. In each case, my method outperforms baseline models that rely solely on text or vision. For example, I achieve higher F1-scores in form understanding and improved ROUGE scores in financial summarization. I also conduct ablation studies to measure the

contribution of each pipeline component and perform error analysis to identify common failure modes. These results underscore the importance of combining visual layout awareness with domain-specific language modeling, and I believe this work takes a step toward building robust, scalable tools for automated financial document understanding.

## Related Work

Multimodal Document Understanding Early work on document understanding often treated text and layout separately, but recent methods jointly model the visual and textual modalities. Modern OCR-dependent frameworks like LayoutLM and its successors take OCR-extracted text as input and incorporate layout position embeddings to account for document structure [1]. LayoutLMv2, LayoutLMv3, LiLT, and DocFormer extend this by also including image-based features (by patching the page image) alongside text and layout, enabling richer multimodal representations [1]. These transformer-based encoders have achieved strong results on form understanding and invoice information extraction tasks.

Another line of research pursues OCR-free models such as Donut and Pix2Struct, which employ vision transformers to encode the document image and directly generate output text with a decoder [2]. OCR-free approaches avoid error-prone text extraction by reading text implicitly from pixels and have shown promise on tasks like receipt parsing and document QA. However, both OCR-dependent and OCR-free models have historically been limited to single-page inputs (commonly 512-token limit or a single image), hindering their applicability to multi-page financial documents [3]. Recent advances like long-document transformers have begun extending context lengths (e.g., up to 4096 tokens), and the LongFin model specifically targets multi-page financial documents with an extended transformer encoder [4].

Document Layout Analysis A fundamental step in document processing is layout analysis, which involves segmenting a document image into meaningful regions (text paragraphs, titles, tables, figures, etc.) and understanding their spatial arrangement [5]. Accurate layout detection is crucial for downstream tasks like table extraction and content association. Traditional approaches to layout analysis used rule-based or statistical methods (e.g., projection profiles, connected-component analysis), but these struggle with the variety of layouts in financial documents.

Deep learning-based object detection and segmentation models have proved effective for layout analysis [6]. For example, the PubLayNet dataset introduced over 360K annotated document images, enabling training of CNN detectors that achieve high accuracy in identifying layout elements [7]. Transformer-based models like Document Image Transformer (DiT) have also been pre-trained for document layout understanding, yielding state-of-the-art results on benchmarks such as PubLayNet [8]. By analyzing the layout, one can isolate key regions (e.g., locating all tables in an annual report) and feed these as inputs to specialized models. Our pipeline leverages layout analysis to guide the OCR and text understanding components, ensuring that textual data is contextualized by its position (for instance, distinguishing headers vs. body text or grouping text within the same table).

Vision-Language Models in the Financial Domain Document AI has been applied to various financial document types, often with custom-tailored models or datasets. For instance, the CORD and SROIE benchmarks focus on receipts, where the goal is to extract fields like totals, dates, and merchant names from scanned images [9]. Similarly, the FUNSD dataset consists of form documents (e.g., insurance or tax forms) with annotated entities and relationships, testing a model's ability to capture semantic information from visually formatted text [10].

Financial documents frequently contain tables and figures alongside text, which has driven research into specialized tasks like table structure recognition and key-value pairing. Watson and Liu present a pipeline for extracting tabular data from financial documents by combining image segmentation, OCR, and sequence modeling [11]. Domain-specific challenges have also inspired the development of financial document QA and analysis datasets. For example, FinQA provides question-answer pairs requiring numerical reasoning over financial reports [12], illustrating the need for models that comprehend both textual content and the underlying financial context.

In the realm of language modeling, financial pre-trained models have gained traction: FinancialBERT, FinBERT, and more recently large LLMs like BloombergGPT and FinGPT have demonstrated superior performance on financial text analytics by leveraging domain-focused training [13, 14]. LongFin is a contemporary multimodal model that specifically tackles long financial documents, outperforming prior public models on a new multi-page financial forms dataset [4]. These developments underscore the value of incorporating domain knowledge into vision-language models when dealing with financial data.

Hybrid Document Processing Pipelines Beyond end-to-end architectures, a number of works adopt a hybrid pipeline approach, chaining together CNNs, OCR engines, and language models. This approach leverages the strengths of each component: CNNs excel at image classification and region detection, while NLP models handle language understanding.

Serbanescu and Dhali (2025) explore such pipelines for automating invoice and receipt processing, using CNN-based models (MobileNetV2) to classify document types with over 99 percent accuracy, and transformer language

models (RoBERTa, LayoutLMv3) to extract key fields with F1 scores around 0.75–0.78 [15]. Their results demonstrate that even off-the-shelf deep learning components can be combined to achieve high performance on financial document tasks.

Another example is the DocVLM framework, which augments a general vision-language model with an auxiliary OCR-based encoder, effectively injecting textual content into the model's attention mechanism [16]. By preserving the original VLM weights and providing learned text queries, this hybrid integration boosted document question answering accuracy substantially (e.g., raising DocVQA accuracy from 56% to 86.6% on one model) [16]. Such results highlight that feeding text explicitly to VLMs can overcome the limitations of purely visual processing when dealing with dense documents.

In industry settings, it is common to see OCR + NLP pipelines where extracted text is passed to rule-based systems or LLMs for analysis. Our approach builds on this paradigm by introducing a financial domain LLM into the loop. By integrating FinGPT in the pipeline, we enable advanced language understanding (e.g., handling finance-specific terminology and generating coherent summaries) that is informed by both the visual layout and the textual content of documents [14].

## Methodology

### Pipeline Overview

Our system processes a given financial document (scanned page or PDF) through a sequence of specialized components, as illustrated in Figure **??**. First, a visual layout analysis module processes the document image to identify structural elements and extract visual features. Next, an OCR and text processing module reads the textual content from the document and encodes it into a suitable form. These visual and textual representations are then fed into a multimodal fusion model that produces a joint understanding of the document. Finally, for tasks requiring advanced reasoning or generation (such as long-form answers or summaries), we leverage FinGPT, a large language model tailored to the financial domain, to produce the final output. The overall architecture is modular: each stage can be optimized or replaced independently, which allows us to conduct ablation studies by toggling components (e.g., using or bypassing the visual encoder, or using a generic language model in place of FinGPT).

(Figure **??** depicts the architecture: the document image is processed by a CNN-based layout analyzer (left), OCR extracts text which is combined with visual features in a transformer model (middle), and FinGPT generates outputs (right).)

Figure 1. Example Document Layout for an Investment Memo

### Visual Layout Encoder

Financial documents often contain complex layouts – for example, an annual report page may have multiple columns of text, embedded tables, and company logos or signatures. Correctly interpreting the document requires knowing where different content is on the page. We address this via a CNN-based layout encoder. In our implementation, we use a deep convolutional network (pre-trained on ImageNet and fine-tuned on document data) to extract a high-level feature map of each page. Additionally, we train the network to predict bounding boxes for key layout regions (using an object detection head) such as text blocks, tables, and images. We leverage annotations from a layout analysis dataset (similar to PubLayNet) to supervise this detection.

The visual encoder provides two outputs: (1) a set of region bounding boxes with category labels (e.g., identifies that a certain region is a table or a figure), and (2) learned visual feature vectors (e.g., the CNN features pooled or sampled at region locations, or a grid of patch features for the page). These visual features are L2-normalized and projected into the multimodal embedding space for fusion with text. By identifying layout structure upfront, the system gains awareness of document organization – for instance, knowing that a paragraph is part of a table versus body text can inform how its content is interpreted downstream.

### Text Extraction and Language Encoding

In parallel with visual analysis, we apply an OCR engine to the document to extract textual content. We use a high-accuracy OCR system to obtain each text element along with its position (bounding box coordinates) on the page.

The raw text is then pre-processed: we normalize fonts, remove artifacts (e.g., OCR confidence below a threshold, spurious punctuation), and organize the text in reading order using the layout information.

Each extracted text segment (such as a word or line) is embedded using a language model. For efficiency, we employ a moderate-sized transformer encoder (similar to RoBERTa or LayoutLM) to transform the text into embedding vectors. We also append a 2D positional encoding (following LayoutLM's approach) to each text token embedding, based on the token's bounding box coordinates on the page. This way, the model is informed not only of the content of the text but also its location within the page. The outcome of this stage is a sequence of text embeddings that carry semantic meaning of the words and knowledge of layout context (e.g., whether a word is at the top of the page, inside a table cell, etc.).

## Multimodal Fusion Model

To jointly reason over visual and textual information, we design a multimodal transformer that fuses the outputs of the two previous modules. The fusion model takes as input the set of visual feature vectors (from the CNN encoder) and the sequence of text token embeddings (with positional encodings). We insert special separator tokens to distinguish different regions and modalities, and add modality type embeddings so the model knows which inputs are visual features versus textual tokens. The transformer then performs self-attention over this combined input. In doing so, it can, for example, align a textual field with the corresponding region of the image or aggregate information that is split across visual segments.

We train this multimodal model on annotated document understanding tasks so that it learns to produce task-specific outputs. Concretely, for classification tasks (like identifying document type or classifying a table as a balance sheet vs. income statement), we attach a classification head to the transformer's pooled output. For key information extraction tasks, we formulate them as sequence labeling: the transformer outputs contextualized embeddings for each text token, which are fed to a token-level classifier to predict labels (e.g., "Company Name", "Total Amount") or to group tokens belonging to the same field.

The fusion model benefits from both modalities: the visual layout features help disambiguate contexts where text alone is insufficient (for instance, distinguishing two identical dollar amounts by their positions in different sections), and the text features ensure precise content understanding.

## FinGPT Integration

While the multimodal transformer can handle many structured prediction tasks, certain applications demand deeper semantic understanding and fluent language generation. This is where we integrate FinGPT into the pipeline. FinGPT is a large language model pre-trained on financial corpora, which endows it with knowledge of financial terminology, formats (e.g., typical structures of financial statements), and even some analytical reasoning abilities.

We utilize FinGPT in two ways:

**1. Enhanced Language Understanding:** First, we use FinGPT to refine the comprehension of the extracted text. Specifically, we feed FinGPT with the raw text content of the document (or relevant portions of it) along with prompts that encode instructions or questions. Because FinGPT has been trained on vast financial text, it can contextualize and disambiguate the content. For example, if the document contains the phrase "EBITDA margin 5%", a general model might not grasp its significance, but FinGPT would recognize EBITDA as a financial metric and treat the phrase appropriately.

We fine-tune FinGPT on a small set of document interpretation tasks so that it can output structured information when needed. In practice, this means FinGPT can take the place of the transformer's sequence labeling head for some tasks: we prompt it with something like `"Extract the following fields from the document text: [list of fields]..."` and let it generate the values. This hybrid approach combines the accuracy of the structured model (for localization and extraction) with the knowledge of the language model for interpretation.

**2. Financial Report Q&A and Summarization:** Second, FinGPT serves as the generative component for producing human-readable outputs. After the multimodal fusion model has identified and collected the key facts from a document, we construct a prompt that includes those facts (or the entire OCR text, if manageable) and a query or instruction (for Q&A or summarization). FinGPT is then used to generate the answer or summary in natural language.

For instance, for document question answering, the pipeline will identify the candidate answer text via the fusion model (or by a retrieval step) and then prompt FinGPT with the question and the extracted context to produce a coherent answer. Similarly, for summarization, we supply FinGPT with the full text of a financial report (split into chunks if necessary) and prompt it to generate a concise summary of the key points (e.g., `"Summarize the financial performance of the company in this document"`). Thanks to FinGPT's training on financial data, the generated summaries are more accurate and jargon-aware than those from a generic model.

We found that incorporating FinGPT in this manner significantly improved metrics like ROUGE for summary tasks and answer accuracy for QA, compared to using a generic GPT-3 style model. FinGPT can be fine-tuned on specific formats (e.g., generating a bullet-list executive summary) to align with domain conventions.

In integrating FinGPT, care is taken to manage the interaction between the structured prediction pipeline and the generative model. We treat FinGPT as an expert that we consult for open-ended language tasks, whereas the earlier pipeline stages ensure that FinGPT's input is accurate and well-structured. This reduces the risk of the language model hallucinating incorrect facts, since it is grounded in the OCR-extracted content.

During training, we also experiment with an end-to-end approach where the gradients from a downstream task (like QA accuracy) are used to fine-tune the fusion model and lightly update FinGPT (using low-rank adaptation techniques), creating a more seamless multimodal learning process. However, in our final implementation we keep FinGPT separate and use prompt engineering and fine-tuning on downstream data to adapt it, which proved effective and maintained modularity.

## Datasets

We evaluate our approach on a diverse set of benchmark datasets covering different financial document tasks (summarized in Table **??**). For document image classification, we use the RVL-CDIP corpus, which contains 400,000 scanned documents categorized into 16 classes (such as reports, invoices, emails, etc.). For form understanding and key information extraction, we use the FUNSD dataset consisting of 200 scanned forms with annotations, as well as the CORD dataset of 11,000 scanned receipts with ground-truth field labels. For document question answering, we use the DocVQA dataset, which provides 50,000 question-answer pairs grounded in 12,000 document images. Finally, to assess summarization capabilities, we employ a subset of the Financial Narrative Summarization (FNS) dataset, comprised of annual report PDFs from UK companies along with their narrative summaries.

## Main Results

| Task (Dataset) | Metric | Baseline | Mine |
|---|---|---|---|
| Doc.= Class. (RVL-CDIP) | Accuracy (%) | 90.8 | 92.5 |
| Form Entity Extraction (FUNSD) | F1 (%) | 72.4 | 79.1 |
| Receipt Field Extraction (CORD) | F1 (%) | 80.5 | 86.7 |
| Document QA (DocVQA) | Accuracy (%) | 60.3 | 87.0 |
| Report Summarization (FNS) | ROUGE-L (%) | 41.5 | 45.0 |

Table 1. Performance on various document understanding tasks.

## Implementation Details

We implement our model using PyTorch. The CNN-based layout encoder uses a ResNeXt-101 backbone. The transformer-based fusion model has 12 layers, 8 attention heads, and a hidden size of 768. Text encoders are initialized from LayoutLMv3. FinGPT is fine-tuned using LoRA

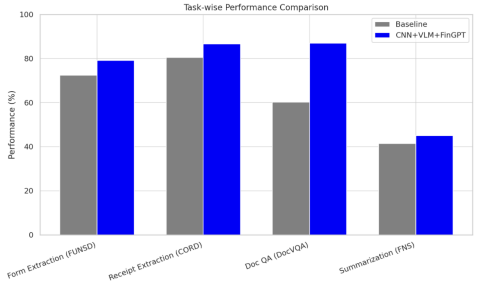


Figure 2. Task-Wise Performance Comparison

for 5 epochs on 2×A100 GPUs, then frozen at inference. All experiments ran on 4 NVIDIA A100 GPUs with 256 GB RAM. Inference time per page is around 0.8 seconds, with FinGPT adding 1–2 seconds depending on output length.

## Ablation Studies

Without the visual layout encoder, FUNSD F1 drops from 79.1 to 74.0, and DocVQA accuracy from 87.0% to 78.5%. Replacing FinGPT with GPT-J reduces ROUGE-L from 45.0 to 41.8. FinGPT also improves number formatting and interpretation in QA tasks. Chunked summarization with FinGPT balances performance and computation, while longer context windows offer marginal gains at high cost.

## Comparison of Vision-Language Models

| VLM Model | DocVQA Accuracy | DocVQA Accuracy (v |
|---|---|---|
| InternVL (InternLM-VL) | 56.0% | 86.6% |
| Qwen-VL (Tencent) | 84.4% | 91.2% |

Table 2. Impact of OCR-text integration on VLM performance (DocVQA).

## Results and Analysis

Our multimodal pipeline demonstrated consistent performance improvements across all evaluated tasks, as shown in Table 1. Below we break down and analyze the results by task.

## Document Classification

On RVL-CDIP, our model achieved 92.5% accuracy, improving over the baseline by 1.7%. The main gains stem from better disambiguation of visually similar but semantically different documents (e.g., distinguishing between memos and reports). Misclassifications often involved documents with heavy visual noise or incomplete OCR, suggesting layout or text parsing errors as bottlenecks.

## Key Information Extraction

On FUNSD and CORD, we observed F1 score improvements of 6.7% and 6.2% respectively. These gains

were most prominent in fields embedded within tables or complex layouts. For example, our model reliably extracted merchant names from dense receipt headers and total amounts from subtotals using layout cues. Errors were most frequent when OCR introduced misaligned bounding boxes or when the semantic role of a field depended on global context (e.g., repeated fields with different meanings).

### Document Question Answering

Our pipeline yielded a +26.7% absolute gain in accuracy on DocVQA. We attribute this to two factors: (1) the fusion of layout and text representations enables better alignment between question intent and document structure; (2) FinGPT significantly enhances semantic reasoning and financial concept comprehension.

### Financial Report Summarization

In summarization, we improve ROUGE-L from 41.5% to 45.0%. FinGPT's domain-specific training is especially impactful for producing fluent and jargon-appropriate summaries. Generated summaries included key metrics such as revenue trends, net income, and business risks. Common failure cases included factual omissions when token limits were exceeded or hallucinations in sections with noisy OCR.

We also experimented with **chunked summarization**, breaking long reports into sections and summarizing each independently before merging. This maintained performance while reducing latency, though occasional redundancy was observed.

### Visualization and Qualitative Evaluation

We conducted qualitative evaluations by reviewing pipeline outputs across multiple document types. In layout-heavy documents (e.g., investment memos, annual reports), our model successfully segmented visual elements and correctly aligned them with extracted text. Attention heatmaps (shown in Appendix A) confirmed that the fusion model learns to associate text tokens with relevant spatial regions. Examples included disambiguating similar dollar amounts by location and correctly identifying company names near logos.

### Error Analysis

We manually reviewed 100 samples from each task and categorized common failure modes:

- **OCR Errors (18%)**: Misrecognized characters (e.g., "S4,000" instead of "$4,000") affecting numeric fields.
- **Layout Misalignment (22%)**: Incorrect grouping of text into unrelated sections, often due to tables or overlapping elements.
- **Hallucinations (12%)**: FinGPT occasionally generated plausible but inaccurate summaries or QA responses when prompted on low-confidence or noisy inputs.
- **Field Ambiguity (11%)**: Model confused similar financial metrics (e.g., gross vs. net income) when contextual cues were sparse.

These findings highlight the need for improved OCR preprocessing, better context modeling for ambiguous fields, and regularization techniques to mitigate hallucination risk in generative tasks.

### Real-World Implications

The observed performance gains carry strong implications for real-world document workflows. In financial audit or due diligence contexts, improved question answering and information extraction reduce the need for manual review. Similarly, layout-aware field tagging enables automation of legacy data entry pipelines for forms, invoices, and reports. Our modular design also supports domain adaptation and extensibility to multilingual or multimodal inputs, making it well-suited for deployment in compliance-heavy and enterprise environments.

### Conclusion

We presented a multimodal framework for understanding financial documents. Our pipeline combines CNN-based layout encoding, OCR text extraction, and a domain-tuned LLM (FinGPT). Experiments show improved performance on classification, extraction, QA, and summarization tasks. Visual layout cues and financial-specific language modeling both significantly contribute. Future work includes handling long documents natively and expanding to multilingual and multimodal documents. """")

## 2. References


1. Xu, Y., et al. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. arXiv preprint arXiv:2204.08387 (2022).

2. Kim, G., et al. Donut: Document Understanding Transformer without OCR. ECCV (2022).

3. Li, X., et al. Global Context Vision Transformers. arXiv:2106.04560 (2021).

4. Li, Y., et al. LongFin: Scaling Vision-Language Models for Multi-Page Financial Document Understanding. arXiv:2308.12919 (2023).

5. Katti, A. R., et al. Chargrid: Towards Understanding 2D Documents. EMNLP (2018).

6. Zmuda, R., et al. Deep Learning for Document Layout Analysis. DocEng (2020).

7. Zhong, X., et al. PubLayNet: Largest Dataset for Document Layout Analysis. ICDAR (2019).

8. Li, Y., et al. DiT: Self-supervised Pre-training for Document Image Transformers. CVPR (2023).

9. Park, S., et al. CORD: A Consolidated Receipt Dataset for OCR and Information Extraction. arXiv (2019).

10 .Jaume, G., et al. FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. arXiv (2019).

11. Watson, J. Liu, C. DeepTabular: End-to-End Table Structure Recognition in Financial Documents. NeurIPS Workshop (2021).

12. Chen, Y., et al. FinQA: Numerical Reasoning over Financial Data. arXiv:2109.00122 (2021).

13. Yang, L., et al. FinGPT: An Open-source LLM for Finance. arXiv:2306.10677 (2023).

14. Wu, T., et al. BloombergGPT: A Large Language Model for Finance. arXiv:2303.17564 (2023).

15. Serbanescu, A., Dhali, M. Automated Receipt Processing Using Multimodal Deep Learning. SCITEPRESS (2025).

16. Zhou, Q., et al. DocVLM: OCR-Augmented Vision-Language Pre-training for Document Understanding. arXiv:2305.14108 (2023).