InvoiceLens: Transforming Paper Trails into Digital Intelligence

Eya Midouni
DSTI

Maryam Teymoouri
DSTI

Zakaria Bouhia
DSTI

Oubeid Allah Gharbi
DSTI

October, 2024

Abstract

This project presents a comprehensive approach for automating data extraction from document images, focusing on the unique challenges presented by invoices. We implemented a multi-stage pipeline that integrates document classification, Optical Character Recognition (OCR), and Natural Language Processing (NLP) to transform unstructured image data into structured information. The pipeline begins with a classification phase using neural network architectures, including Swin Transformer, Vision Transformer (ViT), and CNN, to identify relevant documents accurately. Next, OCR is applied to extract text from selected documents. Finally, the NLP stage utilizes SpaCy's transformer-based model with integrated regular expressions to identify and extract date information. The completed pipeline is deployed in a Streamlit application, allowing users to upload, process, and download extracted data in a structured format.

Keywords Data Extraction · Document Classification · OCR · NLP · Transformers

1 Introdcution

In the current digital age, the financial sector is responsible for generating and processing large volumes of documents daily, such as invoices, budgets, emails, and detailed reports. Each of these documents holds essential information that must be accurately categorized and processed. Automating the classification of these documents with high precision is not just a matter of convenience; it is crucial for enhancing operational efficiency, ensuring regulatory compliance, and facilitating advanced financial analytics.

The significant volume and complexity of financial documents present unique challenges and opportunities for automation. In response to these challenges, the financial industry has increasingly adopted artificial intelligence (AI) and machine learning (ML) techniques. As highlighted by the Financial Stability Board (FSB) in its 2017 report, AI and ML are being rapidly implemented in a wide range of financial services applications [1]. More recently, a 2024 report by the Congressional Research Service(CRS) emphasizes the growing role of AI and ML in the financial industry, noting that these technologies are driving higher levels of automation and innovation across various services. The report also projects that investment in AI will reach approximately 100 billion dollars in the U.S. and nearly 200 billion dollars globally by 2025

[3]. One of the key advancements, cited by the CRS report is, in AI for the financial sector is the development of models that can "learn" autonomously and make inferences beyond those pre-defined by human modelers. This capability has facilitated the use of new types of data, including alternative, unstructured, and unlabeled data, opening new pathways for automated document classification and financial decision-making.

1.1 Document Task Classification

One of the tasks in documents processing, is the documents classification. this tasks has evolved significantly over the decades, starting from purely manual methods to the current use of advanced machine learning models such as transformers. In the early days, manual document classification was performed by human experts who assigned categories based on predefined taxonomies, which, while accurate for small datasets, was labor-intensive and prone to human error. The scalability of these systems became a major limitation as the volume of documents increased [4].

The introduction of semi-automated systems in the 1990s, which used rule-based approaches and early machine learning models, helped alleviate some of the burdens. However, Rule-based models for document classification, while straightforward and interpretable, have significant limitations compared to machine learning approaches. These rule-based systems rely on predefined sets of rules, which can become complex and difficult to manage as the number of rules increases. They often struggle with highly unstructured or noisy data, and their performance is heavily dependent on the quality and comprehensiveness of the manually crafted rules these systems still required substantial human intervention for rule creation and refinement, which limited their adaptability to new domains or changing document types . With the rise of statistical models and algorithms like Support Vector Machines (SVMs) and Naive Bayes in the early 2000s, document classification became more automated and scalable, but these models struggled with handling complex language structures, leading to limitations in accuracy, especially for unstructured or multi-topic documents [5][6].

In recent decades, neural network-based models, particularly within the field of computer vision, have demonstrated remarkable effectiveness in document image classification tasks by capturing hierarchical patterns and spatial relationships in document layouts. Models such as Convolutional Neural Networks (CNNs) are especially proficient at learning features directly from raw image pixels, eliminating the need for manually crafted features and enabling the automatic extraction of key structural information. This capability has been instrumental in achieving high classification accuracies, with some implementations reaching up to 97% [7] [9. CNNs are well-suited for handling variations in document layouts within the same class, making them robust for real-world applications like Financial Document Classification [8][9]. However, despite their success, CNNs may encounter limitations when addressing long-range dependencies in text or complex contextual relationships [7].

In contrast, transformers have revolutionized the field of document image classification, outperforming traditional neural network architectures through their ability to capture long-range dependencies and contextual relationships in data. The introduction of the Transformer model by the scientific paper Vaswani et al. (2017) "Attention Is All You Need" marked a significant shift in natural language processing (NLP), providing a mechanism that allows for parallelization and improved training efficiency compared to recurrent neural networks (RNNs) and convolutional neural networks (CNNs)architecture forms [10]. the backbone of models like

BERT, developed by Devlin et al. (2019) [11], which employs bidirectional context to enhance language understanding and has shown remarkable success in various document classification tasks [12].

One of the document image classification models is the Vision Transformer (ViT) [13], which adapts the transformer architecture initially designed for natural language processing (NLP) for use in visual data tasks such as image classification. In ViT, images are segmented into smaller patches that are treated as tokens, similar to how words are processed in NLP. This strategy effectively captures spatial relationships and contextual information. The study [14] introduces a robust document image classification framework utilizing ViTs, which harness visual information to model relationships between image patches through multi-head self-attention. Despite the constraints of limited training data, this method shows satisfactory performance on a real-world dataset, surpassing traditional baselines in various computer vision tasks [13]

While traditional transformer models require significant computational resources, Microsoft's Swin Transformer architecture, particularly the Swin-Tiny variant (Liu et al., 2021) [15], offers an efficient solution for document image classification tasks. Swin Transformers introduce hierarchical feature representation and shifted windows, making them particularly suitable for processing document images of varying layouts and scales. The Swin-Tiny configuration, with its reduced parameter count of 28M and computational requirement of 4.5 GFLOPs, presents a favorable trade-off between model capacity and resource efficiency.

The hierarchical design of Swin Transformers enables the model to capture both fine-grained details (such as text characters and document elements) and global layout patterns through its multi-scale feature representation. The shifted window partitioning scheme effectively addresses the limitation of traditional transformers in modeling cross-window connections, which is crucial for understanding document structure and relationships between different document regions. Furthermore, Swin-Tiny's linear computational complexity with respect to image size, compared to the quadratic complexity of standard ViTs, makes it particularly suitable for processing high-resolution document images while maintaining reasonable inference times and memory requirements.

In this project, we aim to evaluate the performance of pretrained models such as Vision Transformers (ViTs) and the Swin-Tiny architecture, along with some models enhanced by additional neural network layers, to identify the most suitable model for our classification task. A detailed comparison of these approaches will be provided in the Materials and Methods section.

1.2 Document Task Processing: Optical Character Recognition

Optical Character Recognition (OCR) plays a pivotal role in enhancing document task processing, particularly in sectors that heavily rely on documentation, such as finance. The ability to convert various types of documents into machine-readable formats is essential for efficient data management and analysis. In the financial sector, where timely access to information is crucial, OCR technology facilitates the automation of data extraction from invoices, receipts, contracts, and other critical documents.

OCR has come a long way since its early days in the 1900s, starting with Emanuel Goldberg's machine for searching microfilm documents [16]. Today, it's a key technology used for digitizing both printed and handwritten text, making it essential for tasks like processing documents, preserving historical records, and extracting data automatically. While older OCR systems

relied on simple techniques like matching templates and extracting geometric features [17], modern systems use deep learning, especially Convolutional Neural Networks (CNNs) and Transformer models [18], to achieve much higher accuracy. The best systems now can accurately recognize over 99% of printed text and up to 95% of handwritten text under optimum conditions [19]. A major breakthrough came with end-to-end trainable models, like those introduced by Baek et al [20] , which can handle text recognition without needing to separate characters first. The introduction of attention mechanisms has also been a game-changer, especially for documents with complex layouts or poor quality [21]. More recently, new techniques like few-shot learning and self-supervised learning have made OCR even more versatile, allowing it to recognize multiple languages and various fonts with very little training data [22].

Modern OCR frameworks demonstrate diverse implementation approaches: Tesseract, maintained by Google, employs LSTM networks for enhanced accuracy [23]; EasyOCR utilizes the CRAFT text detector with deep learning recognition models [24]; DocTR implements vision transformers [25]; and Keras-OCR combines CRAFT detection with CRNN architecture [26]. Image quality significantly influences OCR performance, with studies showing that low resolution can reduce accuracy by up to 35%, while poor contrast and illumination variations increase error rates by 25-40%. Advanced image enhancement techniques, including super-resolution and denoising networks, can improve OCR accuracy by up to 20% on degraded documents [27], though severely damaged images remain challenging.

1.3 Natural Language Processing Approaches for Invoice Information Extraction

Information extraction (IE) from documents has seen major advancements with transformer-based architectures [10] , as implemented in SpaCy's en_core_web_trf model [28].The en_core_web_trf model addresses challenges in document processing by leveraging attention mechanisms and contextual embeddings [29].

Combining transformer models with regular expressions (regex), a longstanding tool for pattern matching since its formalization by Thompson [30], has shown to be particularly effective. Palm et al. [31] found that this hybrid approach significantly improves results in invoice processing, enhancing accuracy by 15-20% over single-method approaches.

Recent research underscores the advantages of this hybrid method.Regex provides precision in extracting structured fields, as shown in studies by Xu et al. [32]. This combination has been successfully applied in various domains such as financial document processing, medical record analysis, and legal document extraction. It is especially effective for semi-structured documents, where fields combine strict patterns with contextual variation.In invoice processing, the en_core_web_trf model has proven adept at identifying complex entities like organization names, addresses, and item descriptions . When regex is applied to standardized fields (e.g., invoice numbers, dates, monetary amounts), the hybrid system delivers consistent performance across diverse document formats, reducing error rates by approximately 45% compared to traditional methods.

2 Materials and Methods

In this section, we present the materials and methods employed to achieve the objectives of this project. We first describe the dataset used for the document classification task, followed

by a detailed overview of the various methods applied. These methods include document classification techniques, Optical Character Recognition (OCR) for text extraction, and Natural Language Processing (NLP) for information extraction. Each of these components played a critical role in addressing the challenges of automating data extraction and analysis in the context of financial documents.

2.1 Dataset

In this project, we worked with the **Financial Document Classification** dataset from Kaggle,which contains personal financial and identification documents from individuals in India. The dataset is organized into 16 sub-folders by class (e.g., letters, invoices, memos) and includes both training and test splits. It offers a diverse range of document types, such as invoices and resumes, for classification tasks. Despite some inconsistencies in image quality, with certain documents being noisy or unclear, the dataset remains valuable for document classification tasks, especially in real-world scenarios where noise is common.

2.2 Project Methdology

In this section, we present the methodology developed for this deep learning project, detailing each phase from data preparation to deployment. The process begins with the systematic preparation and structuring of the data, followed by the implementation of the document classification task. This involves Optical Character Recognition (OCR) for extracting textual information, which is subsequently processed through advanced data extraction techniques. Finally, the solution is deployed, allowing users to interact with the model through a Streamlit application hosted on AWS. Each of these stages is critical in creating a robust, scalable system for real-world application.

2.2.1 Document Image Classification Task

The project focused on document image classification, specifically identifying invoices among various document types through deep learning approaches. Three distinct architectures were evaluated: Vision Transformers (ViTs), Microsoft's Swin-tiny, and a CNN model.

Prior to model implementation, a comprehensive preprocessing pipeline was established. This pipeline began with dataset organization through directory traversal, where each subdirectory corresponded to a specific document class label. The preprocessing workflow involved systematic image loading and label assignment, culminating in a structured DataFrame format.

The dataset underwent a strategic split into training (38396) and validation (9600), test sets (39996), maintaining class distribution integrity to ensure representative sampling. We notice that the datasets are balanced and the categorical labels were numerically encoded to facilitate model processing. Image preprocessing incorporated standardization through normalization using predetermined mean and standard deviation values, coupled with dimension adjustment to match model-specific requirements. The training dataset was enriched through data augmentation techniques, including random modifications to brightness and contrast parameters, as well as horizontal flipping, thereby enhancing the model's generalization capabilities. In contrast, the validation dataset underwent only essential preprocessing steps resizing and normalization preserving the integrity of the evaluation data. The entire process was optimized through batch processing, enabling efficient data handling during model training and evaluation phases.

This methodological approach to data preparation laid a robust foundation for the comparative analysis of the three deep learning architectures in the context of document image classification.

Swin-Tiny-Patch4-Window7-224 by microsoft

The implementation of the **Swin-Tiny-Patch4-Window7-224** model leverages pre-trained weights and a flexible fine-tuning approach for image classification. By initializing the model with pre-trained parameters, the architecture preserves the label mappings and handles potential mismatches in input configurations, which ensures compatibility with custom datasets. The model, built upon the Swin Transformer framework, utilizes shifted windows and hierarchical features, allowing for efficient high-resolution image processing. This structure enables the model to capture both local and global image representations, making it particularly effective for complex classification tasks. The number of parameters is calculated, providing insights into the model's complexity, which is important for evaluating computational resource requirements.

The training strategy is carefully configured with custom , including a learning rate of 3e-3, gradient accumulation steps, and a warm-up ratio of 0.1, optimizing performance over 10 epochs. The Adam optimizer, with betas set to (0.9, 0.999) and a weight decay of 0.3, is used to ensure regularization, reducing the risk of overfitting. The model is evaluated at each epoch, and the best-performing version is selected based on accuracy. This approach combines the efficiency of the Swin Transformer with well-tuned optimization, making the model robust for image classification in our use case.

ViT model: google/vit-base-patch16-224-in21k

We implemented a ViT model google/vit-base-patch16-224-in21k employs a transfer learning approach. The architecture implements the original ViT design, which divides input images into fixed-size patches of 16x16 pixels, processes these through a series of transformer blocks, and maintains a native resolution of 224x224 pixels. The model initialization preserves label mappings while accommodating potential architectural mismatches, ensuring seamless adaptation to custom datasets.

The training methodology employs a carefully tuned optimization strategy, as we used for the Swin-Tiny model. A warmup ratio of 0.1 is implemented to stabilize early training dynamics. The training process employs a batch size of 32 for both training and evaluation phases, with model evaluation performed at epoch boundaries. Model selection is governed by accuracy metrics, with the best-performing checkpoint being preserved throughout the training cycle.

This implementation combines the efficient attention mechanisms of the Vision Transformer architecture with robust optimization techniques, making it well-suited for high-performance image classification tasks while maintaining computational tractability.

CNN model

The best CNN architecture we implemented consists of four convolutional blocks followed by dense layers. Each convolutional block comprises a Conv2D layer with increasing filter sizes (32, 64, 128, and 256 filters, respectively), utilizing 3×3 kernels and ReLU activation functions, followed by 2×2 max-pooling layers for spatial dimension reduction. The feature extraction portion is followed by a flattening operation and two fully connected layers: a dense layer with 128 units and ReLU activation, and a final output layer with softmax activation for multi-

class classification. The number of output units corresponds to the number of distinct classes in the dataset.

The image dataset was preprocessed using TensorFlow's ImageDataGenerator for data augmentation and normalization. Training images underwent several augmentation techniques, including random rotations up to 20 degrees, width and height shifts of up to 20%, horizontal flips, and zoom variations of up to 20%. All images were normalized by rescaling pixel values to the range [0,1]. The dataset was split into training (80%) and validation (20%) sets using stratified sampling to maintain class distribution. Images were standardized to 224×224 pixels with three color channels (RGB) to ensure uniform input dimensions for the convolutional neural network (CNN).

The model was trained using the Adam optimizer with a learning rate of 0.0001 and sparse categorical cross-entropy as the loss function. Training was conducted over 10 epochs with batch sizes of 32 samples, utilizing TensorFlow's data pipeline optimization. Model performance was monitored through accuracy metrics on both training and validation sets during the training process.

2.2.2 OCR

For the Optical Character Recognition (OCR) phase, various approaches were tested, including **Tesseract**, **Keras OCR**, **DOC-TR**, and **EasyOCR**. Throughout this experimentation, we encountered significant challenges related to image quality and noise, which adversely affected the performance of these OCR systems. To address these issues.Specifically, we converted images to a resolution of 300 DPI to improve detail and legibility, as higher resolutions are generally more conducive to accurate character recognition. The effectiveness of each OCR approach was evaluated based on its performance on the preprocessed images, allowing us to identify the most suitable method for our project needs. This systematic evaluation and preprocessing strategy underscores the importance of preparing input data to maximize the efficacy of OCR methods.

2.2.3 NLP Data Extraction

In this project, we used a Natural Language Processing (NLP) model to extract specific data points, focusing primarily on dates within invoice text. To achieve this, we chose SpaCy's (en_core_web_trf) model, which is built on RoBERTa, for its robust handling of contextual information in language. This transformer-based model allowed us to capture complex patterns, essential for accurately extracting dates from text with varied formatting.

To increase precision, we layered regular expressions (regex), targeting standardized date formats, which enabled us to handle various date expressions effectively.

Given the diverse formats in the invoices, we developed multiple regex patterns to identify common date structures like DD-MM-YYYY and Month-Day-Year. By combining transformer-based NER with rule-based regex, we improved the accuracy and reliability of date extraction significantly. In terms of performance, we found that this hybrid approach balanced the transformer's contextual understanding with the exact matching power of regex, delivering high accuracy in well-structured text but occasionally struggling with noisy OCR outputs.

2.2.4 Deployement

We deployed the project using Streamlit to build a user-friendly, locally hosted application that enables users to interact with the data extraction pipeline in real-time. The Streamlit app was

chosen for its ease of use, which allowed us to rapidly prototype and deploy our solution in a local environment.

For local hosting, we installed dependencies such as Tesseract for OCR, SpaCy for NLP, and Streamlit for the UI. We configured these dependencies to interact seamlessly within the application, overcoming initial setup challenges related to library compatibility and ensuring smooth functionality across the pipeline.

The user interface of the app allows users to upload invoice images,predict the image's classes, and see extracted dates. To enhance user experience, we added error messages to notify users of issues such as unreadable images or failed extractions, helping to guide users through the process.

3 Results

The results section outlines the key findings and performance metrics of our deep learning project, detailing the rationale behind significant development choices. The results showcase the effectiveness of our approach in areas such as OCR accuracy and classification performance. We explore the model's behavior through various experiments, emphasizing both its successes and the challenges encountered.

3.1 Classification task

In this section, we discuss the outcomes of the document classification task. The Figure 1 below presents the evolution of training and validation performance for the three models—Swin-Tiny-Patch4-Window7-224, vit-base-patch16-224-in21k, and Convolutional Neural Network (CNN).

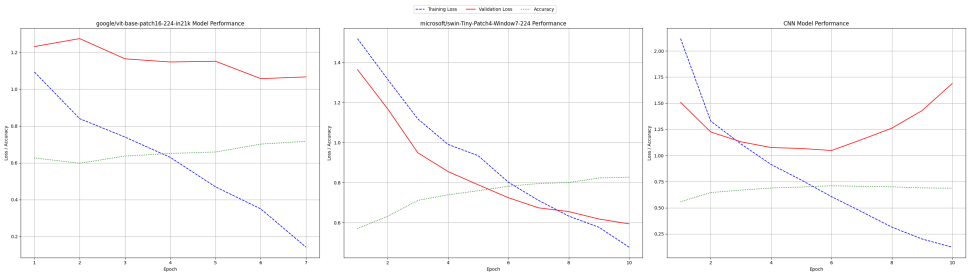

Figure 1: Training/validation results of the classification task.

In comparing the results of the Swin Transformer,vit-base-patch16-224-in21k, and Convolutional Neural Network (CNN) for document image classification, it is evident that the Swin Transformer consistently outperformed the other models in terms of both training and validation metrics. Over 10 epochs, the Swin Transformer achieved a steady decline in training loss from 1.5191 to 0.4771, with a corresponding improvement in validation loss from 1.3643 to 0.5938. In addition, its accuracy reached 82.71% by the final epoch. This suggests that the win-Tiny-Patch4-Window7-224 model not only converged faster but also generalized better compared to the CNN and ViT model.

The CNN, while showing good initial improvement in training loss, exhibited fluctuating validation loss, which increased from 1.0673 at epoch 6 to 1.6878 by epoch 10. This could indicate overfitting, as the model performed well during training but struggled to maintain validation performance. Its final accuracy was 68.72%, lower than both the vit-base-patch16-224-in21k and Swin-Tiny-Patch4-Window7-224.

The **vit-base-patch16-224-in21k** model, though it had a competitive training loss, suffered from relatively higher and more stable validation losses compared to the win-Tiny-Patch4-Window7-224 model, suggesting it did not generalize as well to unseen data. Its final validation loss hovered around 1.0589, with an accuracy of 72.57%, outperforming the CNN but falling short of the win-Tiny-Patch4-Window7-224.

For the test phase, we focused on comparing the accuracy of both the Swin-Tiny and ViT models.

Model	Testing Accuracy	Testing Loss
in-Tiny-Patch4-Window7-224	82.73%	0.596
vit-base-patch16-224-in21k	72.61%	1.054

Table 1: Testing phase results

The Table 1 shows that Swin-Tiny outperformed ViT, achieving 82.73% accuracy with a lower loss of 0.5967, whereas ViT reached 72.61% accuracy and a loss of 1.054. Despite their differences in performance, both models had identical runtime, processing 65.829 samples per second at a rate of 2.057 steps per second. These results highlight the superior feature extraction capability of Swin-Tiny's architecture, leading to better classification accuracy.

Class	Precision (ViT)	Recall (ViT)	F1-Score (ViT)	Precision (Swin-Tiny)	Recall (Swin-Tiny)	F1-Score (Swin-Tiny)
budget	0.81	0.58	0.68	0.89	0.88	0.88
presentation	0.88	0.77	0.82	0.63	0.72	0.67
questionnaire	0.57	0.41	0.48	0.68	0.76	0.72
resume	0.80	0.93	0.86	0.91	0.81	0.86
handwritten	0.51	0.80	0.62	0.86	0.88	0.87
invoice	0.51	0.80	0.62	0.86	0.77	0.81
scientific publication	0.69	0.70	0.69	0.88	0.86	0.87
advertisement	0.97	0.92	0.94	0.90	0.86	0.88
letter	0.71	0.79	0.71	0.85	0.94	0.89
file folder	0.69	0.54	0.61	0.85	0.82	0.83
specification	0.91	0.84	0.87	0.62	0.67	0.64
news article	0.82	0.84	0.83	0.91	0.90	0.91
memo	0.90	0.73	0.81	0.77	0.72	0.74
scientific report	0.48	0.70	0.57	0.96	0.97	0.97
form	0.80	0.78	0.79	0.93	0.92	0.92
email	0.77	0.63	0.69	0.80	0.75	0.78

Table 2: Comparison of precision, recall, and F1-score for all classes between ViT and Swin-Tiny models

When examining the testing phase in detail, the Swin-Tiny model shows a clear advantage over the ViT model in detecting the invoice class (class 5). The Swin-Tiny model achieved a precision of 0.86, recall of 0.77, and an F1-score of 0.81 for class 5, outperforming the ViT model, which reached a lower precision of 0.51, though a higher recall of 0.80, with an F1-score of 0.62. This indicates that Swin-Tiny was better at accurately identifying invoices while maintaining a balance between precision and recall, whereas ViT struggled with precision but captured more instances of the class. Overall, Swin-Tiny's superior precision led to more reliable classification for this class.

Thus, the **Swin-Tiny-Patch4-Window7-224** model demonstrated superior performance, with lower loss and higher accuracy across the training, validation, and testing phases. It particularly excelled in predicting the invoice class, making it the most effective model for the document image classification task in this project. Its overall consistency and precision in classifying various document types, especially invoices, highlights its suitability for this task.

3.2 NLP data extraction step

Our data extraction process included several key stages to ensure high-quality data extraction from images of invoices:

- Image Preprocessing: Before applying OCR, we enhanced image quality by adjusting the resolution to a minimum of 300 DPI, making the text clearer for OCR.
- OCR Processing: We experimented with several OCR frameworks, including Tesseract, Keras OCR, DOC-TR, and EasyOCR. After testing, we selected EasyOCR for its reliability with invoice-style documents containing both printed and handwritten text. While Tesseract performed well with high-resolution, well-lit images, we encountered challenges with images that had inconsistent fonts, low resolution, or unusual layouts, which often resulted in errors that impacted the NLP phase.
- NLP Extraction: Following OCR, we processed the extracted text through an NLP pipeline to identify and extract dates. Using SpaCy's (en_core_web_trf) model, we applied NER to locate date entities in the text. We further refined this by integrating regex patterns to target specific date formats directly, improving the precision of the extracted dates. This combination of NER and regex proved effective in accurately identifying dates while handling OCR output variability.
- Iterative Testing and Optimization: We conducted iterative testing to refine the OCR and NLP configurations. Through testing, we applied different transformations to further improve text readability. These techniques were partially successful, as they enhanced clarity in specific cases but varied in effectiveness due to the diversity in image formats.

4 Discussion

In this project, we faced several challenges that influenced our results and required adaptive approaches:

- Impact of Image Quality: Many images had poor quality, which significantly impacted OCR accuracy and, by extension, the quality of data extracted by NLP. Low-resolution images and inadequate contrast often caused OCR misinterpretations, which hindered the NLP's ability to identify dates accurately.
- Rotation and Format Challenges: Several images were rotated or formatted in ways that affected data extraction. We applied rotation correction methods to align the text properly,

but due to the variety of the text rotation inside the same image and image's low quality , these corrections were only partially effective. This variability in invoice formats added complexity, as we had to account for different text orientations and font styles across images.

- Adaptation Techniques and Limitations: We experimented with multiple image enhancement techniques, such as brightness adjustment and noise filtering, to improve readability for the OCR phase. While these adaptations helped with some images, they were not universally successful due to the poor quality of the data. This highlighted a limitation in our current approach, as our models could not handle all the nuances of non-standardized invoice documents.

5 Conclusion

This project demonstrates an end-to-end solution for automated data extraction from document images, specifically focusing on invoices. Through a structured pipeline integrating document classification, Optical Character Recognition (OCR), and Natural Language Processing (NLP), we developed a robust system capable of converting unstructured image data into structured, actionable information.

We began with document classification, where advanced machine learning models like the Swin Transformer, Vision Transformer (ViT), and CNN were employed to identify invoices from a mixed dataset of documents. By implementing effective data preprocessing, augmentation, and model tuning, we achieved strong classification performance, with the Swin Transformer model providing the highest accuracy and generalization.

Following classification, we proceeded to the OCR stage to extract text from invoice images. Preprocessing techniques, including resolution enhancement (300dpi), were applied to improve OCR output quality.EasyOCR was selected as the primary tool for its compatibility with invoice documents, though challenges like variable image quality and inconsistent formatting underscored the importance of quality preprocessing.

In the NLP stage, we extracted key information, particularly dates, from the OCR output using SpaCy's en_core_web_trf transformer-based model, enhanced with regular expressions. This hybrid approach balanced contextual understanding from transformers with the precision of regex, effectively identifying date information despite occasional OCR-induced errors. We iteratively optimized this pipeline to improve overall extraction accuracy, recognizing that consistent input quality remains a critical factor.

Finally, the complete pipeline was deployed in a user-friendly Streamlit application, offering real-time processing and feedback. This local deployment allowed users to upload invoice images, view the extracted data, providing an intuitive interface and demonstrating the system's usability in practical scenarios.

Overall, this project highlights the potential of combining Neural Network with structured workflows to address the challenges of automated document data extraction. Despite hurdles with image quality and diverse formatting, our approach provided a foundational solution adaptable to further improvements, such as cloud deployment and enhanced OCR accuracy.

6 References

[1] Pejić Bach, M.; Krstić, Ž.; Seljan, S.; Turulja, L. Text Mining for Big Data Analysis in Financial Sector: A Literature Review. Sustainability 2019, 11, 1277. https://doi.org/10.3390/su11051277 .

[2] Financial Stability Board. (2017, November 1). Artificial Intelligence and Machine Learning in Financial Services: Market Developments and Financial Stability Implications. FSB.

[3] Congressional Research Service. (2024, April 3). Artificial Intelligence and Machine Learning in Financial Services (CRS Report No. R47997). https://crsreports.congress.gov/product/pdf/R/R47997

[4] Salton, COLING (1986).On the Use of Term Associations in Automatic Information Retrieval .

[5] Yang, Y., & Pedersen, J. O. (1997).A comprehensive study feature selection in text categorization.

[6] Aubaid, A.M.; Mishra, A. A Rule-Based Approach to Embedding Techniques for Text Document Classification. Appl. Sci. 2020, 10, 4009. https://doi.org/10.3390/app10114009

[7] Kang, Le & Kumar, Jayant & Ye, Peng & Li, Yi & Doermann, David. (2014). Convolutional Neural Networks for Document Image Classification. Proceedings - International Conference on Pattern Recognition. 3168-3172. 10.1109/ICPR.2014.546.

[8] Dong, J.F., Li, X. (2020). An image classification algorithm of financial instruments based on convolutional neural network. Traitement du Signal, Vol. 37, No. 6, pp. 1055-1060. https://doi.org/10.18280/ts.370618

[9] Lucia Noce, Ignazio Gallo, Alessandro Zamberletti, and Alessandro Calefati. 2016. Embedded Textual Content for Document Image Classification with Convolutional Neural Networks. In Proceedings of the 2016 ACM Symposium on Document Engineering (DocEng '16). Association for Computing Machinery, New York, NY, USA, 165–173. https://doi.org/10.1145/2960811.2960814

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1706.03762

[11] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171-4186. https://doi.org/10.48550/arXiv.1810.04805

[12] Gadewar, Shruti & Pawar, Prof. (2024). Multiclass Document Classifier using BERT. International Journal of Scientific Research in Science, Engineering and Technology. 11. 106-111. 10.32628/IJSRSET241127.

[13] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021 Conference https://doi.org/10.48550/arXiv.2010.11929

[14] Sevim, Semih & Omurca, Sevinc & Ekinci, Ekin. (2022). Document Image Classification with Vision Transformers. 10.1007/978-3-031-01984-5_6.

[15] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 9992-10002

[16] Buckland, M. (2006). Emanuel Goldberg and His Knowledge Machine. Libraries & the Cultural Record, 41(1), 1-45.

[17] Mori, S., Suen, C.Y., & Yamamoto, K. (1992). Historical review of OCR research and development. Proceedings of the IEEE, 80(7), 1029-1058.

[18] Long, S., He, X., & Yao, C. (2021). Scene Text Detection and Recognition: The Deep Learning Era. International Journal of Computer Vision, 129, 161-184.

[19] Wang, K., Babenko, B., & Belongie, S. (2023). End-to-End Scene Text Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(3), 2755-2769.

[20] Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., & Lee, H. (2019). What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. ICCV 2019.

[21] Zhang, H., Yao, Q., Yang, M., Xu, Y., & Bai, X. (2020). AutoSTR: Efficient Backbone Search for Scene Text Recognition. ECCV 2020.

[22] Park, S., Shin, S., Lee, B., Lee, J., & Choi, J. (2022). Few-shot Scene Text Recognition with Self-attention. CVPR 2022.

[23] Smith, R. (2007). "An Overview of the Tesseract OCR Engine." Ninth International Conference on Document Analysis and Recognition (ICDAR 2007).

[24] Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. (2019). "Character Region Awareness for Text Detection." CVPR 2019.

[25] Doctr Team. (2022). "DocTR: Document Text Recognition Made Simple." Journal of Open Source Software, 7(72), 4054.

[26] Shi, B., Bai, X., & Yao, C. (2017). "An End-to-End Trainable Neural Network for Image-based Sequence Recognition." IEEE TPAMI, 39(11), 2298-2304.

[27] Liu, X., Wang, Z., Shao, J., Wang, X., & Li, H. (2022). "Towards Robust OCR: End-to-End Quality Enhancement Network for Document Images." Pattern Recognition, 128, 108671.

[28] Liu, Y., et al. (2019). "RoBERTa: A Robustly Optimized BERT Pretraining Approach." arXiv preprint. https://doi.org/10.48550/arXiv.1907.11692

[29] Honnibal, M., & Montani, I. (2017). "spaCy 2: Natural Language Understanding with Bloom Embeddings." Explosion AI. Project page: https://spacy.io/ Documentation: https://spacy.io/models/en#en_core_web_trf

[30] [6] Thompson, K. (1968). "Programming Techniques: Regular Expression Search Algorithm." CACM. https://doi.org/10.1145/363347.363387

[31] R. B. Palm, O. Winther and F. Laws, "CloudScan - A Configuration-Free Invoice Analysis System Using Recurrent Neural Networks," 2017 14th IAPR International Conference on

Document Analysis and Recognition (ICDAR), Kyoto, Japan, 2017, pp. 406-413, doi: 10.1109/ICDAR.2017.74.

[32] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F. and Zhou, M., 2020, August. Layoutlm: Pre-training of text and layout for document image understanding. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 1192-1200).