Consolidación de Facturas Contables

José Emanuel Figueroa, Juan Esteban Correa, Esteban Alexander Arias

josee.figueroas; juane.correap; estebana.ariasg@utadeo.edu.co

Universidad Jorge Tadeo Lozano — Programa: Ciencias de Datos, Modelado y Simulación

Curso: Inteligencia Artificial — Bogotá D.C., Colombia

05 de octubre de 2025

1. Resumen

Proponemos una aplicación de tipo móvil (Android) o implementada en Web, con interfaz para equipos móviles, consistente en una IA de tipo **Intelligent Document Processing pipeline**, con la capacidad de analizar facturas individuales de tipo talonario y clasificar los datos contenidos en estas: nombre de la empresa emisora, NIT, dirección, fecha de emisión, cliente, productos vendidos o comprados, cantidad, valor en COP, con la capacidad de exportar estos datos a una tabla de Excel, con el fin de que los equipos contables puedan usar este formato (`.xlsx`) en sus labores contables. Se busca un error menor al 1/20, siendo esta la versión inicial o *pitch* de la aplicación.

2. Problema local y motivación

En aras de mantener un sistema fiscal consolidado, la DIAN necesitan recaudar información contable de todos los negocios inscritos como personas jurídicas. La mayoría de estos negocios están compuestos por menos de cinco personas como empleados calificados en cualquier actividad menos la contable.

Estas personas dependen de contratar un contador de profesión, quien realiza la Declaración de Renta anual y visita la empresa mensualmente para consolidar las cuentas. Como estos negocios no poseen una persona dedicada a tiempo completo para registrar las facturas, tanto de ventas como de compras, el contador visitante debe pasar al menos un día completo registrándoles manualmente en una hoja de cálculo antes de iniciar el análisis contable.

La idea de desarrollar una aplicación con IA capaz de reconocer óptica-mente el texto en las facturas y completar automáticamente una hoja de cálculo permitirá que tanto el contador como el negocio accedan fácilmente a la información contable consolidada.

3. Dataset

Varios de los miembros del equipo trabajamos en el área contable, y con ciertos permisos hemos logrado acceder a una fuente de facturas tipo talonario, este dataset sigue

siendo propiedad de las empresas que lo registran y se usa únicamente con fines académicos. La cantidad de documentos es variable debido al tipo de talonario, la diversidad de negocios y la caligrafía particular de cada persona que lo diligencia.

El dataset principal estará compuesto por 300 imágenes de facturas reales proporcionadas por la compañia contable M&M Consultorias SAS. Estas facturas estarán en formato de imagen (.jpg o .png) y contendrán variables como el nombre del emisor, NIT, fecha, descripción de productos, valores, IVA y totales. Su licencia será de uso interno académico con consentimiento, y la diversidad de tipos de factura, caligrafías y diseños garantizará una adecuada representatividad. Como fuentes públicas complementarias, se emplearán conjuntos de OCR abiertos como el FUNSD Dataset y el SROIE 2019 Dataset.

4. Tarea de IA y algoritmo(s)

La aplicación se compondrá de dos módulos: uno de **detección de caracteres** y otro de **clasificación semántica**, con una consolidación programática posterior en una hoja de cálculo.

El reconocimiento de caracteres se basará en **Deep Learning**, específicamente en redes convolucionales (CNN), utilizando modelos de OCR (*Optical Character Recognition*) como **Tesseract 5**, **Google Cloud Vision** y **Amazon Textract**, que usan arquitecturas RNN, LSTM o Transformer con mecanismos de *Self-Attention*, *Differentiable Binarization* o YOLO.

El segundo módulo, de clasificación, aplicará técnicas de **NLP** mediante modelos como **NER** o **BERT**, para clasificar los campos extraídos en categorías comprensibles (NIT, fecha, valor, productos, etc.). Se prestará atención a modelos *open source* como **PaddleOCR**, **EasyOCR**, **HuggingFace**, **LayoutLM** y **SpaCy**.

Será necesario realizar **Fine-Tuning** o **Transfer Learning** para asegurar una correcta interpretación. Los resultados serán almacenados en formato `JSON` y exportados a Excel mediante librerías como `pandas`, `openpyxl` o `xlsxwriter`. Finalmente, la interfaz se desarrollará con Dart y Flutter, o con JavaScript.

5. Metodología y evaluación

Las facturas serán escaneadas con smartphones —el medio previsto para su uso real—, aplicando conversión a alto contraste para mejorar la lectura del OCR. Se usará una división 70/30 para entrenamiento y prueba, seleccionando ejemplos variados. Los resultados se compararán con una consolidación humana basada en los valores reales.

Fases del proceso:

- **Preprocesamiento:** conversión a escala de grises, binarización adaptativa, aumento de contraste y eliminación de ruido.
- **Entrenamiento y validación:** división del dataset (70/15/15) y *transfer learning* para adaptar modelos preentrenados.

- **Evaluación:** métricas como *Character Accuracy Rate (CAR)*, *Word Error Rate (WER)* y *F1-score*.
- **Líneas base:** comparación con OCR comerciales sin ajuste, como Tesseract sin fine-tuning.

6. Resultados esperados, ética y cronograma

Se espera que el sistema alcance una precisión superior al 95 % en reconocimiento y clasificación, reduciendo en al menos 60 % el tiempo promedio que los contadores dedican al registro manual. Esto permitirá a las microempresas mejorar su trazabilidad contable y reducir errores.

Consideraciones éticas: los datos personales serán anonimizados y cifrados; las copias locales serán eliminadas. Todo el material se utilizará con consentimiento y fines académicos.

Cronograma tentativo (4 semanas):

- Semana 1: recopilación, limpieza y anonimización de datos.
- Semana 2: entrenamiento del modelo OCR y pruebas de preprocesamiento.
- Semana 3: integración del módulo NLP y validación cruzada.
- Semana 4: despliegue del prototipo web/móvil, evaluación y documentación final.

Roles del equipo: El equipo de trabajo está conformado por José Emanuel Figueroa, quien asumirá el liderazgo técnico general, el diseño del pipeline de OCR y el ajuste del modelo base; Juan Esteban Correa, encargado de la integración del modelo NLP, la evaluación de resultados y la documentación técnica; y Esteban Alexander Arias, responsable del desarrollo de la interfaz gráfica en Flutter o JavaScript, las pruebas de usuario y la conexión con el módulo contable. [*]

Referencias

[1] Amazon AWS. *Textract Developer Guide*. AWS Documentation. 2023.

[2] J. Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers". En: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. 2019.

[3] Explosion AI. *SpaCy NLP Library*. Software Library. 2023.

[4] Google Cloud. *Cloud Vision OCR Documentation*. Google Cloud Documentation. 2023.

[5] ICDAR. *SROIE Dataset, ICDAR 2019 Competition on Scanned Receipts OCR and Information Extraction*. 2019.

[6] G. Jaume. "FUNSD: Form Understanding in Noisy Scanned Documents". En: (2019).

[7] PaddleOCR. *PaddleOCR, Open Source OCR System*. GitHub Repository. 2023.

[8] Python Software Foundation. *OpenPyXL Documentation*. Python Library Documentation. 2023.

[9] R. Smith. "An Overview of the Tesseract OCR Engine". En: *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*. 2007.

[10] Y. Xu et al. "LayoutLMv3: Pre-training for Document AI". En: *arXiv preprint arXiv:2204.08387* (2022).