Ingenería en Informática

Trabajo Final OCR

Alumno: Luis Pedro Solares Serrano

Profesor: Jose Manuel Breñosa

Asignatura: Inteligencia Artificial

Santander, 12 de Enero del 2026

1. INTRODUCCIÓN

1.1 Planteamiento del problema

El reconocimiento óptico de caracteres (OCR, Optical Character Recognition) es una de las áreas clásicas de la Inteligencia Artificial y la Visión por Computador. Consiste en la conversión automática de texto contenido en imágenes a formato digital editable.

El reto planteado en este trabajo consiste en desarrollar un sistema OCR desde cero, sin utilizar librerías externas especializadas que ya resuelvan directamente el problema (como Tesseract), abordando de forma manual cada una de las etapas del proceso: preprocesamiento, segmentación y reconocimiento.

El sistema debe ser capaz de reconocer tanto caracteres impresos como manuscritos, lo que introduce dificultades adicionales relacionadas con la variabilidad en la forma de los caracteres, el grosor del trazo, la inclinación y el ruido presente en las imágenes.

1.2 Objetivos del proyecto

El objetivo principal de este proyecto es diseñar e implementar un sistema OCR funcional utilizando Python, capaz de:

- Leer imágenes externas en formato PNG.
- Detectar y segmentar caracteres individuales dentro de una imagen con texto.
- Reconocer caracteres manuscritos y digitales.
- Convertir el texto reconocido a formato digital.

Como objetivos secundarios se incluyen:

- Implementar una arquitectura modular y fácilmente ampliable.
- Analizar el rendimiento del sistema.
- Documentar el proceso de desarrollo y las decisiones técnicas tomadas.

1.3 Restricciones del proyecto

Una de las principales restricciones del trabajo es la prohibición del uso de librerías externas que ya resuelvan el problema del OCR de forma directa. Por ello, no se han utilizado herramientas como Tesseract, EasyOCR o similares.

Sí se permite el uso de librerías de propósito general como:

- OpenCV (procesamiento básico de imágenes)
- NumPy (operaciones matriciales)
- Matplotlib (visualización)

Estas librerías no realizan OCR por sí mismas, sino que permiten implementar las etapas necesarias desde un enfoque algorítmico.

1.4 Enfoque general de la solución

La solución propuesta se divide en las siguientes fases:

1. **Preprocesamiento de imágenes**
2. **Segmentación de caracteres**
3. **Reconocimiento de caracteres**
4. **Reconstrucción del texto**

Cada fase se implementa en un módulo independiente, facilitando la comprensión, el mantenimiento y la mejora del sistema.

2. SOLUCIÓN APORTADA

2.1 Arquitectura del sistema

El proyecto se ha estructurado de forma modular, separando las distintas responsabilidades en archivos independientes:

```
OCR/
├── Cuaderno.ipynb
├── Src/
│   ├── preprocesamiento.py
│   ├── segmentation.py
│   └── recognition.py
├── Data/
└── test_images/
```

Esta estructura permite aislar cada fase del OCR y facilita la depuración y evolución del sistema.

2.2 Preprocesamiento de imágenes

El preprocesamiento tiene como objetivo mejorar la calidad de la imagen y facilitar la segmentación posterior.

Las operaciones realizadas son:

- Conversión a escala de grises.
- Redimensionado a 28×28 píxeles.
- Binarización mediante umbral fijo.
- Normalización de valores al rango [0,1].

Estas operaciones permiten reducir la variabilidad de las imágenes y trabajar con un formato homogéneo.

2.3 Segmentación de caracteres

La segmentación se realiza detectando contornos en la imagen binarizada. Para cada contorno detectado:

- Se calcula su bounding box.
- Se filtran regiones demasiado pequeñas (ruido).
- Se centra el carácter dentro de una imagen cuadrada.
- Se redimensiona a 28×28 píxeles.

Los caracteres segmentados se ordenan de izquierda a derecha para reconstruir el texto original.

2.4 Reconocimiento de caracteres

El reconocimiento se realiza mediante comparación con un conjunto de plantillas obtenidas a partir del dataset de entrenamiento.

Para cada carácter:

- Se compara con las plantillas promedio de cada clase.
- Se calcula una medida de similitud basada en el producto escalar.
- Se selecciona la clase con mayor puntuación.

Este método, aunque simple, ofrece resultados razonables para caracteres bien segmentados.

2.5 Análisis de rendimiento

El rendimiento del sistema depende en gran medida de:

- La calidad del preprocesamiento.
- La correcta segmentación de los caracteres.
- La similitud entre el manuscrito de entrada y los ejemplos del dataset.

En pruebas realizadas con textos cortos, el sistema presenta un rendimiento aceptable para un enfoque basado en plantillas, especialmente en caracteres impresos.

3. REGISTRO DE RESULTADOS

Fecha	Prueba	Resultado	Anotaciones
26/12	Carga de imágenes	Correcto	Se corrigieron rutas
27/12	Segmentación	Parcial	Ajuste de centrado
08/01	Reconocimiento	Mejorado	Uso de plantillas promedio

4. CONCLUSIONES

El sistema desarrollado demuestra que es posible construir un OCR funcional sin recurrir a herramientas especializadas, abordando manualmente cada una de las fases del proceso.

Aunque el rendimiento no alcanza el de soluciones comerciales, el proyecto cumple con los objetivos propuestos y permite comprender en profundidad los desafíos asociados al reconocimiento de caracteres manuscritos.

Como mejoras futuras se plantea:

- Uso de clasificadores KNN o redes neuronales.
- Segmentación adaptativa para textos cursivos.
- Reconocimiento de palabras completas.