

OCR con Red Neuronal Convolutiva (CNN)

1. Introducción y objetivo del proyecto

El objetivo de este proyecto es el desarrollo de un sistema de **Reconocimiento Óptico de Caracteres (OCR)** implementado **íntegramente desde cero**, como parte de la evaluación de la asignatura de Inteligencia Artificial.

A diferencia de soluciones comerciales o librerías OCR ya existentes, este proyecto impone la restricción explícita de **no utilizar sistemas de reconocimiento de texto de caja negra** (como Tesseract OCR, EasyOCR o APIs externas).

Esto obliga a abordar de forma explícita **todos los subproblemas clásicos del OCR**, desde la segmentación hasta la clasificación, y no únicamente el entrenamiento de un modelo de Deep Learning.

El objetivo final no es únicamente obtener un sistema funcional, sino **entender y justificar la complejidad real de un OCR**, así como las decisiones técnicas necesarias para resolverla.

2. Alcance funcional del sistema

El sistema desarrollado es capaz de:

- Procesar imágenes externas en formatos estándar (PNG, JPG, BMP)
- Trabajar con texto sobre fondo claro
- Segmentar el texto en caracteres individuales
- Reconocer:
 - Dígitos (0–9)
 - Letras mayúsculas (A–Z)
 - Letras minúsculas (a–z)
- Reconstruir el texto final carácter a carácter

El sistema funciona tanto con:

- Texto **impreso**
- Texto **manuscrito**

3. Complejidad del problema OCR

El reconocimiento óptico de caracteres no es un problema único, sino la **composición de varios problemas encadenados**, cada uno con sus propias dificultades:

1. **Preprocesamiento de imagen**
2. **Segmentación correcta de caracteres**
3. **Normalización geométrica**
4. **Clasificación robusta**

5. Corrección de errores sistemáticos

Un fallo en cualquiera de estas etapas degrada el sistema completo, lo que obliga a un diseño modular y a un proceso iterativo de mejora.

Este proyecto aborda explícitamente cada una de estas etapas, justificando la evolución de las técnicas utilizadas.

4. Tecnologías utilizadas y justificación

Python 3

Lenguaje principal del proyecto por su ecosistema científico, facilidad de prototipado y compatibilidad con bibliotecas de visión artificial y Deep Learning.

OpenCV

Utilizado para:

- Conversión a escala de grises
- Binarización
- Operaciones morfológicas
- Segmentación de caracteres

Se descartó el uso de herramientas OCR integradas de OpenCV para cumplir la restricción del proyecto.

TensorFlow / Keras

Framework utilizado para:

- Definición de la Red Neuronal Convolutiva
- Entrenamiento desde cero
- Persistencia del modelo

Se optó por Keras por su claridad conceptual y facilidad para experimentar con arquitecturas.

EMNIST

Dataset externo utilizado para:

- Aumentar la variabilidad de ejemplos manuscritos
- Evitar sobreajuste al dataset local
- Simular un escenario realista de OCR

5. Evolución del diseño del sistema (enfoque iterativo)

5.1 Primer enfoque: clasificación directa de caracteres aislados

El desarrollo comenzó con un enfoque reducido:

- Entrenar una CNN para reconocer **caracteres individuales**
- Imágenes ya segmentadas y normalizadas a 28×28

Este enfoque permitió:

- Validar la arquitectura del modelo
- Confirmar que la CNN aprendía correctamente las clases

Limitación detectada:

- No resolvía el problema real del OCR, ya que asumía segmentación perfecta.

5.2 Segundo enfoque: segmentación básica por contornos

El siguiente paso fue integrar la CNN dentro de un pipeline OCR completo.

Se implementó una segmentación inicial basada en:

- Detección de contornos (findContours)
- Cajas delimitadoras (bounding boxes)

Problemas encontrados:

- Separación incorrecta del punto de la i
- Recortes excesivamente ajustados
- Confusión entre caracteres similares (o/0, l/i)

Este enfoque demostró que **la segmentación es tan crítica como la red neuronal**.

5.3 Mejora de segmentación: componentes conexas

Para solucionar los problemas anteriores, se sustituyó la segmentación por contornos por un enfoque basado en:

- **Connected Components Analysis**
- Operaciones morfológicas previas (dilatación ligera)
- Unión de componentes cercanos (ej. punto + cuerpo de la i)

Justificación técnica:

- Las componentes conexas son más estables frente a ruido
- Permiten trabajar directamente con regiones binarias
- Facilitan el filtrado por área y proporciones

Este cambio mejoró significativamente la calidad de los recortes.

6. Normalización y estandarización de caracteres

Cada carácter segmentado se normaliza a:

- Tamaño fijo: **28×28 píxeles**
- Escala de grises
- Fondo blanco / tinta negra
- Centrado geométrico
- Padding para evitar distorsión

Esta normalización es crítica porque:

- La CNN solo aprende correctamente si las entradas siguen una distribución homogénea
- Pequeños errores de recorte generan grandes errores de clasificación

7. Dataset: problemas reales y decisiones de diseño

7.1 Dataset manuscrito

El dataset manuscrito local se organizó explícitamente en:

numeros/

mayusculas/

minusculas/

Cada subcarpeta representa una **clase semántica clara**, lo que evita ambigüedades durante el entrenamiento.

7.2 Dataset impreso: problema mayúsculas/minúsculas

Inicialmente, el dataset impreso no diferenciaba correctamente entre mayúsculas y minúsculas, lo que provocó errores sistemáticos como:

Hola → HOLA

Causa:

- En OCR, el modelo **no interpreta el significado visual**, solo aprende etiquetas.
- Si a se etiqueta como A, el modelo aprende una asociación incorrecta.

Solución:

- Separación explícita en carpetas:
 - numeros/
 - mayusculas/
 - minusculas/

Esta decisión evita además problemas del sistema de archivos en Windows (case-insensitive).

8. Arquitectura de la CNN

La red neuronal implementada sigue una arquitectura CNN clásica:

- Entrada: $28 \times 28 \times 1$
- Capas convolucionales con activación ReLU
- Capas de MaxPooling
- Dropout para regularización
- Capas densas finales
- Salida Softmax con todas las clases

La CNN se entrena **desde cero**, sin pesos preentrenados, para cumplir la restricción académica.

9. Errores típicos y corrección por heurísticas

Incluso con un modelo bien entrenado, existen confusiones inevitables entre caracteres visualmente similares:

- $o \leftrightarrow 0$
- $l \leftrightarrow i \leftrightarrow 1$

Para mitigar estos errores se introdujo:

- Selección Top-2 de predicciones
- Reglas heurísticas basadas en contexto alfabético
- Correcciones locales en la reconstrucción del texto

Este enfoque refleja prácticas reales en sistemas OCR industriales.

10. Estado final del sistema

El sistema final:

- Implementa un OCR completo y funcional
- Distingue mayúsculas, minúsculas y números
- Integra visión artificial, Deep Learning y heurísticas
- Es modular, extensible y explicable

El proyecto demuestra que el OCR es un problema **sistémico**, no únicamente un problema de clasificación

11. Conclusión

11.1 Aprendizajes clave sobre la complejidad del OCR

A lo largo del desarrollo se han identificado varios aspectos fundamentales:

- **El modelo no es el único elemento crítico**
Aunque la CNN alcanza altas tasas de precisión sobre caracteres bien segmentados, el rendimiento global del OCR depende en gran medida de la calidad de la segmentación y del preprocesamiento.
- **La segmentación es un cuello de botella fundamental**
La transición desde una segmentación basada en contornos hacia un enfoque con componentes conexas y operaciones morfológicas fue decisiva para mejorar la estabilidad del sistema, evidenciando que técnicas aparentemente simples pueden tener un impacto mayor que aumentar la complejidad del modelo.
- **El etiquetado del dataset es determinante**
El problema de distinguir mayúsculas y minúsculas puso de manifiesto que la red neuronal aprende exclusivamente a partir de las etiquetas proporcionadas. Una organización incorrecta del dataset conduce a errores sistemáticos imposibles de corregir únicamente mediante entrenamiento adicional.
- **Existen ambigüedades visuales inevitables**
Caracteres como o/0 o l/i presentan similitudes estructurales que incluso un modelo bien entrenado puede confundir. Esto justifica la introducción de reglas heurísticas y decisiones basadas en contexto, reflejando soluciones utilizadas en sistemas OCR reales.

11.2 Valor del enfoque iterativo

Uno de los principales valores del proyecto ha sido el **proceso iterativo de mejora**:

1. Implementación de una solución inicial funcional
2. Identificación de errores reales en escenarios prácticos
3. Análisis de las causas técnicas de dichos errores
4. Sustitución o refinamiento de métodos (segmentación, dataset, post-procesado)

Este ciclo ha permitido evolucionar el sistema desde un prototipo básico hasta una solución mucho más robusta, demostrando que en Inteligencia Artificial el progreso suele venir más de **mejoras estructurales bien razonadas** que de cambios aislados.

11.3 Conclusión final

En conclusión, este proyecto ha permitido adquirir una comprensión profunda del Reconocimiento Óptico de Caracteres como un problema complejo y multifacético, en el que la Inteligencia Artificial debe integrarse cuidadosamente con técnicas de procesamiento de imagen y conocimiento del dominio. La experiencia obtenida demuestra que el éxito de un sistema OCR no reside únicamente en la potencia del modelo de aprendizaje automático, sino en el diseño coherente y justificado de todo el pipeline.

El trabajo realizado constituye una base sólida para futuras ampliaciones, como la incorporación de modelos de lenguaje o el reconocimiento de texto continuo, y representa una experiencia formativa alineada con los desafíos reales del desarrollo de sistemas de visión artificial.