

TECNICAS DE PROCESAMIENTO DIGITAL DE IMAGENES

PROYECTO INTEGRADOR

Docente:
Barreto, Matias

Integrantes:
Allievi, Andress
Serafini, Martin

1° Cuatrimestre 2025



DESARROLLO DE UN MOTOR DE OCR A TRAVÉS DE MODELOS NEURONALES



OBJETIVO Y PROCESO

Objetivo: Crear un sistema de OCR para letras mayúsculas y números utilizando una CNN (Convolutional Neural Network)

Proceso:

1. Preparación de Datos
2. Aumento de Datos (Data Augmentation)
3. Definición de Modelo
4. Entrenamiento
5. Persistencia
6. Inferencia

ARQUITECTURA Y ENTRENAMIENTO

Arquitectura: Red Neuronal Convolutiva (CNN) adaptada para clasificación de caracteres.

- Capas Convolutivas
- Capas de Max-Pooling
- Capa de Aplanamiento
- Capas Densas
- Capa de Salida con Softmax

Entrenamiento:

- Función de Pérdida: `sparse_categorical_crossentropy`
- Optimizador: Adam
- Métrica: `accuracy`
- Proceso por Época

INFERENCIA Y INTERFAZ

- Carga del Modelo
- Procesamiento de Imagen de Entrada
- Detección de Caracteres
- Procesamiento de Cajas (Bounding Boxes)
- Reconocimiento con el Modelo
- Construcción del Resultado
- Visualización
- Interfaz Web

ANALISIS DE RESULTADOS

- El modelo entrenado reconoce los caracteres que se le son presentados (teniendo en cuenta solo los caracteres por los cuales se realizo el entrenamiento -Letras mayusculas y numeros del 0 al 9-)con eficiencia.





DESARROLLO DE DETECTOR DE TEXTO EN IMAGENES



OBJETIVO

Objetivo: Crear un dataset sintético de imágenes de caracteres con sus bounding boxes y luego entrenar un modelo para detectar estos caracteres. Este dataset y modelo están diseñados para servir como base para un detector de texto y sistema OCR.

PROCESO

Proceso:

- Se utilizan fuentes TrueType para generar imágenes base de caracteres (blanco sobre fondo negro).
- Se aplican transformaciones a las imágenes base para aumentar la diversidad.
- Se calculan y almacenan las bounding boxes para cada carácter en cada imagen, en caracteres compuestos se unifica el bounding box para que lo identifique como uno.
- Se generan metadatos (carácter, fuente, tipo de fondo, color).

EXPERIMENTACIÓN Y OPTIMIZACIÓN DE MODELOS DE DETECCIÓN

- Reentrenamiento de modelos disponibles:
 - Desarrollo e implementación de una arquitectura de detección personalizada utilizando TensorFlow.
 - TensorFlow Object Detection API: Pruebas con modelos pre-entrenados y configuraciones disponibles dentro de esta API.
 - YOLOv8: Evaluación de su balance entre velocidad y precisión.
- Se utilizó Transfer Learning (incluyendo pesos de COCO) en las pruebas para acelerar el entrenamiento y mejorar la robustez.

ANALISIS DE RESULTADOS

- El modelo entrenado no llega a identificar los caracteres completos generando bounding boxes parciales sobre detecciones.
 - Se intento modificando el tamaño de las imágenes de entrenamiento, originalmente 32x32 llevándolos hasta 640x640 sin una mejora valida en el resultado y con un costo computacional muy elevado.
- Se probó generando datos de entrenamiento con caracteres combinados (en horizontal y vertical), esto aumento el volumen de los datos de entrenamiento sin beneficios verificables.



GRACIAS!