

20 de junio de 2025

Informe de Trabajo Final Visión por Computadora III - CEIA

Traducción de Imágenes a Texto

Integrantes:
Florentino Arias
Juan Cruz Piñero
Agustina Quiros
Agustín de la Vega

1. Introducción

Este proyecto tuvo como objetivo aplicar y comparar modelos de Vision Transformers (ViT) para tareas de OCR (Reconocimiento Óptico de Caracteres) en escenas naturales, utilizando el dataset COCO-Text. Se realizaron procesos de preprocesamiento, visualización, fine-tuning de modelos preentrenados y visualizaciones de atención para interpretar el comportamiento de cada arquitectura.

2. Datos

2.1. Elección del dataset

COCO-Text [3] es uno de los datasets más grandes dedicados al reconocimiento de texto en escenas en contexto natural. Este conjunto de datos permite abordar diversas tareas como la detección de texto, la transcripción y el reconocimiento óptico de caracteres. Incluye anotaciones detalladas, tales como las cajas delimitadoras (bbox), cadenas de texto en formato UTF-8, legibilidad, idioma, entre otras. Además, está construido sobre el dataset COCO 2014 [4], lo que proporciona acceso a un conjunto amplio y diverso de imágenes.

2.2. Análisis Exploratorio Inicial

2.2.1. Dataset original

- Total de imágenes en train2014: 82783.
- Total de imágenes etiquetadas (con texto): 23485.

2.2.2. Legibilidad

- Cantidad de textos legibles: 80844. - Cantidad de textos ilegibles: 120282.

2.2.3. Distribución de idiomas

- English: 96.36 % - Not english: 3.64 %

2.2.4. Ejemplos de imágenes del dataset

A continuación se muestran ejemplos de imágenes de los conjuntos de entrenamiento y validación.

Images and annotations from the set train2014 examples.

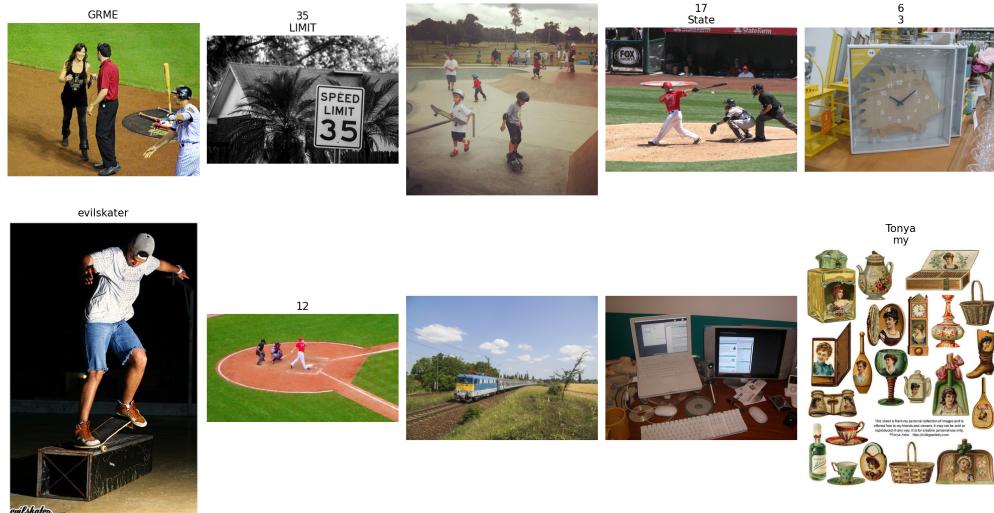


Figura 1: Subconjunto de entrenamiento.

Images and annotations from the set val2014 examples.



Figura 2: Subconjunto de validación.

Se detallan ejemplos de imágenes de las clases *handwritten* y *machine printed*.



Figura 3: Imagen con texto de clase handwritten.



Figura 4: Imagen con texto de clase machine printed.

2.3. Preprocesamiento

- En el archivo train2014 se dispone de 23.485 imágenes de las cuales se tomaron:
 - Imágenes en subset train: 15656, representa aproximadamente 70 %.
 - Imágenes en subset val: 7829, representa aproximadamente 30 %.
- Se filtraron solo las anotaciones legibles con utf8 string no vacío.
- Recorte y guardado de regiones de texto: Se recorren imágenes anotadas, se validan las bounding boxes y se recortan las regiones que contienen texto. Cada recorte se guarda como una imagen PNG en una carpeta de salida.
- Generación de archivo de etiquetas: Por cada recorte válido, se registra su nombre de archivo y el texto correspondiente en un archivo labels.csv, listo para ser usado en entrenamiento de modelos OCR.

3. Modelos Encoder-Decoder

3.1. Elección de modelos

Se seleccionaron dos modelos preentrenados basados en ViT que forman parte de la librería *transformers* de **Hugging Face**:

El modelo **TrOCR** [1] es un modelo codificador-decodificador, que consiste en un Transformador de imagen como codificador y un Transformador de texto como decodificador. El codificador de imagen se inicializó a partir de los pesos de BEiT, mientras que el decodificador de texto se inicializó a partir de los pesos de RoBERTa.

- Identificador: *microsoft/trocr-base-handwritten*.
- Arquitectura: ViT + GPT2.
- Diseñado para transcripción de texto manuscrito.
- Entrenado sobre IAM y similares.
- Requiere recortes individuales de texto como entrada.

Donut [2] consiste en un codificador de visión (Swin Transformer) y un decodificador de texto (BART).

- Identificador: *naver-clova-ix/donut-base*.
- Arquitectura: ViT + Decoder.
- Diseñado para tareas documentales estructuradas.
- OCR-free: no requiere segmentación previa.
- Entrada: imagen completa. Salida: JSON estructurado con todos los textos.

3.2. Fine-tuning

Los modelos preentrenados fueron entrenados en dominios diferentes (documentos, manuscritos, formularios), COCO-Text representa escenas en contextos naturales (carteles, letreros, callejeros). El objetivo de aplicar fine-tuning es que permite adaptar el conocimiento del modelo a este nuevo dominio visual y textual. Este proceso es fundamental para mejorar la capacidad del modelo de generalizar sobre este nuevo tipo de datos.

3.3. Cuestiones de diseño

- TrOCR: Se usaron recortes individuales (regiones de texto) para aprovechar el diseño del modelo orientado a transcripción. Cada imagen se asocia a un string objetivo (utf8-string).
- Donut: Se usaron imágenes completas. Para cada imagen, se generó un target-json que concatena o estructura los textos anotados como respuesta. Se aprovechó su arquitectura OCR-free para evaluar su desempeño en tareas de document understanding.

4. Visualizaciones

Para analizar el comportamiento del modelo TrOCR, se extrajeron los mapas de atención de cada token generado:

- Se generó el mapa de atención del primer token.
- Se generaron secuencias paso a paso mostrando atención por token.
- Se superpusieron los mapas sobre la imagen original recortada.

Esto permitió observar cómo el modelo enfoca su atención en distintas partes del texto a medida que genera cada carácter.

A continuación se detallan imágenes que ilustran como el modelo realiza el proceso de atención para distintos textos.

Atención para letra: '767'



📄 Texto completo generado: 767

Figura 5: Ejemplo 1 TrOCR attention.

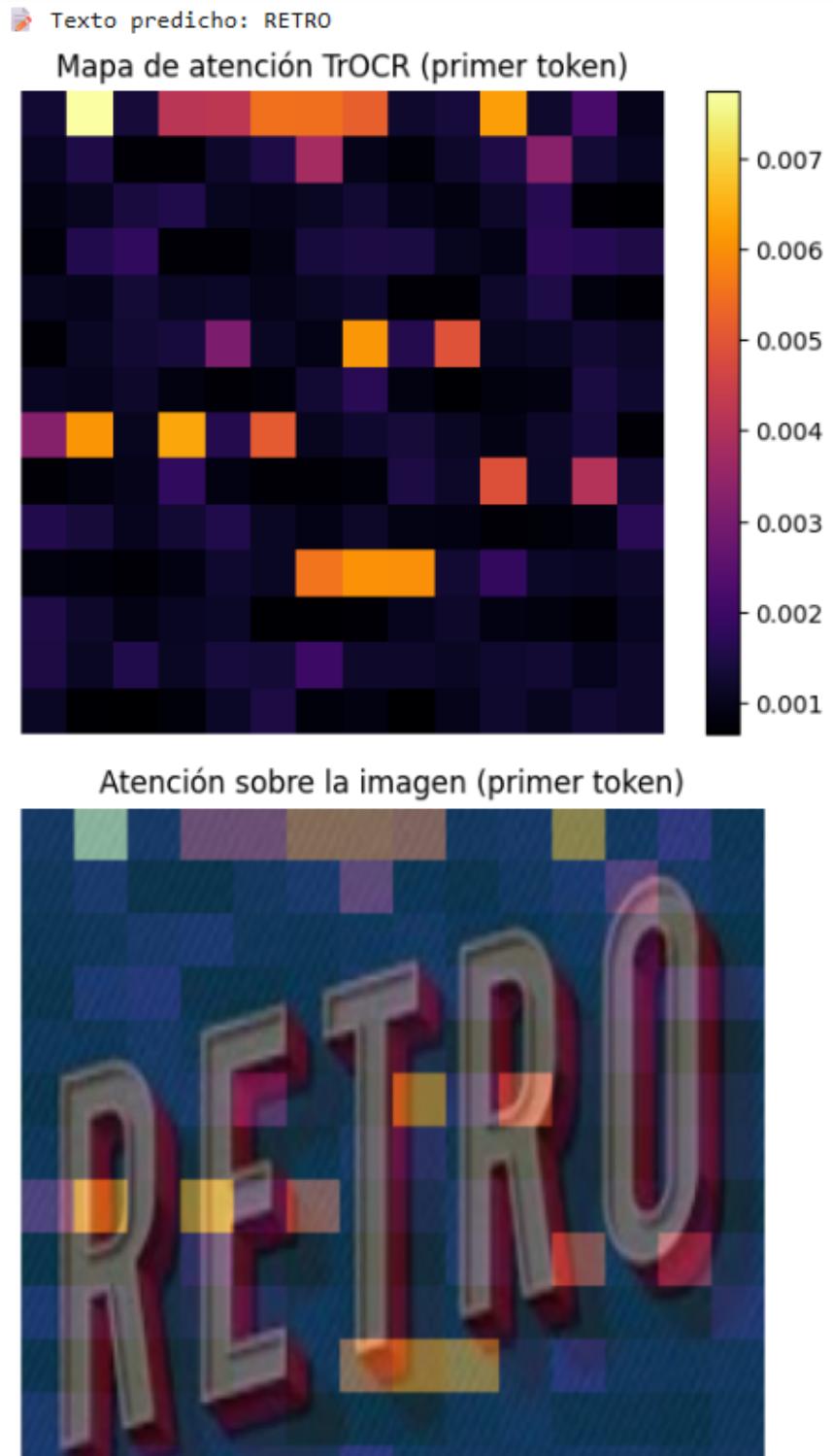


Figura 6: Ejemplo 2 TrOCR attention.

5. Resultados

5.1. Métricas

Se evaluaron los modelos Donut y TrOCR sobre el conjunto de datos de test, a continuación se detallan los resultados obtenidos:

Model	Mean Character Accuracy	Mean Character Error Rate
Donut	-	-
TrOCR	0.6717	0.3283

Cuadro 1: Resultados de evaluación de los modelos Donut y TrOCR.

5.2. Análisis de los resultados

TrOCR	Donut
<ul style="list-style-type: none">■ CER promedio: 3.4 %■ Predicciones precisas en recortes bien centrados y legibles■ Sensible a ruido o recortes mal alineados	<ul style="list-style-type: none">■ Predicciones razonables en contexto global■ Más robusto frente a textos múltiples o estructurados■ Requiere cuidado en el formato del JSON objetivo

Cuadro 2: Comparación entre TrOCR y Donut.

6. Conclusiones

Se pudo demostrar que ambos modelos resultan aplicables de manera efectiva a tareas de reconocimiento óptico de caracteres (OCR) sobre el conjunto de datos COCO-Text. En particular, TrOCR se mostró más adecuado para tareas de transcripción directa en regiones de texto aisladas, mientras que Donut evidenció ventajas en escenarios donde es necesario interpretar la imagen en su totalidad o generar salidas estructuradas. Asimismo, el proceso de fine-tuning resultó fundamental para alcanzar desempeños satisfactorios, dada la divergencia existente entre el dominio de los datos originales y las características específicas del dataset utilizado.

7. Futuras mejoras

Consideramos que se podrían aplicar distintas tareas para la futura mejora de los modelos, a continuación se detallan algunas de ellas:

- Aumentar el tamaño del dataset de entrenamiento.
- Aplicar data augmentation sobre los recortes.
- Usar métricas más precisas como WER (Word Error Rate) [5].
- Explorar adaptaciones de Donut a OCR en escena natural (prompting o instrucción).

Referencias

- [1] Li, M., Yin, F., Zhang, C., & Liu, C. (2021). TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. *arXiv preprint arXiv:2109.10282*. Disponible en: <https://huggingface.co/microsoft/trocr-base-handwritten>
- [2] Kim, G., Kim, Y., Cho, S., & Park, S. (2022). Donut: Document Understanding Transformer without OCR. *arXiv preprint arXiv:2111.15664*. Disponible en: <https://huggingface.co/naver-clova-ix/donut-base>
- [3] Veit, A., Matera, J., Neumann, L., Matas, J., & Belongie, S. (2016). COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images. Disponible en: <https://bgshih.github.io/cocotext/>
- [4] Lin, T.-Y., Maire, M., Belongie, S., et al. (2014). Microsoft COCO: Common Objects in Context. Disponible en: <https://cocodataset.org/#download>
- [5] Wikipedia contributors. (s.f.). Word error rate. Wikipedia. Disponible en: https://en.wikipedia.org/wiki/Word_error_rate