

# CASO 1: Facturas a JSON

## 1. Preprocesamiento e Incremento de Contraste

Para mejorar la calidad de las imágenes de las facturas y facilitar la detección y reconocimiento de texto, se proponen las siguientes técnicas:

- **CLAHE (Contrast Limited Adaptive Histogram Equalization):**  
Este método incrementa el contraste local en áreas con iluminación deficiente o no uniforme, haciendo el texto más visible y robusto ante variaciones de calidad.  
Referencia: Implementación de CLAHE en OpenCV.
- **Binarización Adaptativa:**  
Se aplican métodos como Otsu o binarización adaptativa para separar texto del fondo, eliminando ruido y mejorando la precisión en el reconocimiento.  
Referencia: Implementación de binarización adaptativa en OpenCV.

## 2. Detección de Textos en la Factura

Se propone realizar un fine-tuning de YOLOv8-seg para identificar bloques de texto relevantes en las facturas, utilizando bounding boxes o máscaras para aislar las áreas de interés. Para el dataset se incluirán imágenes en distintos formatos como escritas a mano impresas y con distinta estructura. Para aumentar la precisión del modelo, se realizará un aumento del dataset, modificando las imágenes existentes mediante cambios de orientación, cambios en la iluminación y adición de ruido gaussiano. Utilizando YOLOv8-seg aprovechamos el conocimiento del modelo preentrenado, además de que cuenta con una amplia documentación y es fácil de utilizar. Para garantizar la calidad del entrenamiento y la validez del modelo, se emplearán las siguientes métricas de evaluación:

- **Precisión:** Mide qué porcentaje de las detecciones realizadas por el modelo son correctas (True Positives) en relación con todas las detecciones realizadas. Una alta precisión indica que el modelo comete pocos falsos positivos.
- **IoU (Intersection over Union):** Evalúa la superposición entre los bounding boxes predichos y los reales. Se calcula como:

## 3. Detección de Textos (OCR)

Se utilizará PaddleOCR para extraer el texto de las facturas de cada una de las bounding boxes generadas por YOLOv8-seg, para posteriormente pasárselo como input al modelo que lo transformara a formato JSON.

## 4. Conversión a JSON

Una vez completado el entrenamiento del modelo YOLOv8-seg, se propone emplear el modelo de Hugging Face `mychen76/mistral7b_ocr_to_json_v1`, basado en la arquitectura Mistral-7B-v0.1, para estructurar la información extraída de las facturas en formato JSON.

## Funcionamiento del Modelo

El modelo mychen76/mistral7b\_ocr\_to\_json\_v1 ha sido diseñado específicamente para convertir resultados de OCR en objetos JSON bien formados. Este modelo recibe como entrada las bounding boxes detectadas por YOLOv8-seg, acompañadas del texto extraído de estas regiones mediante OCR. A partir de esta información, genera un JSON estructurado con los campos clave de las facturas.

## **5. Evaluación**

Para la evaluación del sistema al completo, se contará con un conjunto reducido de imágenes de facturas y el json que se espera de cada una de ellas. Se medirá el porcentaje de acierto comparando los números de clave-valor de los JSON predecidos por el sistema que sean iguales a los esperados.

## **6. Justificación de éxito**

Se espera que el rendimiento de este sistema sea exitoso ya que se fundamenta en el uso de modelos, los cuales han demostrado un alto rendimiento en multitud de tareas anteriormente. Además, se incluye un procesamiento previo de las imágenes de las facturas para reducir el riesgo de errores en etapas posteriores.

## **Fuentes**

Documentación YOLOv8: <https://docs.ultralytics.com/es/models/yolov8/#usage-examples>

Modelo Hugging face: [https://huggingface.co/mychen76/mistral7b\\_ocr\\_to\\_json\\_v1](https://huggingface.co/mychen76/mistral7b_ocr_to_json_v1)