2025

Evidencia 1 – Modelos de IA

María Laura Peralta

TSCDIA - ISPC

19-4-2025

Índice

Objetivo

Explorar distintas transformaciones aplicadas a imágenes digitales utilizando las librerías OpenCV y Pillow, con el fin de adquirir habilidades para trabajar con datos visuales en aplicaciones de inteligencia artificial.

Introducción

Las imágenes digitales representan visualmente objetos o escenas del mundo real mediante una matriz de puntos de color llamados **píxeles**. Cada píxel contiene información sobre la intensidad y el color, siendo estos los elementos fundamentales para la interpretación digital de una imagen.

Una imagen se caracteriza principalmente por:

- **Resolución**: número total de píxeles, expresado como ancho × alto.
- **Profundidad de color**: número de bits asignados por píxel para representar colores.

Los formatos de imagen definen cómo se almacena esta información. Algunos de los más utilizados son:

- **JPEG**: ampliamente usado para fotografías; permite compresión con pérdida.
- **PNG**: ideal para gráficos con transparencias; utiliza compresión sin pérdida.
- **GIF**: utilizado para animaciones simples; limitado a 256 colores.
- **BMP**: sin compresión, produce archivos grandes, pero de alta fidelidad.
- **TIFF**: empleado en fotografía profesional; admite compresión sin pérdida y capas.

Con el avance del aprendizaje profundo, las imágenes se transforman en **tensores**, permitiendo su análisis automático por redes neuronales.

Representación de Imágenes en Deep Learning

En el contexto del Deep Learning, una imagen se interpreta como un **tensor** tridimensional compuesto por alto, ancho y canales de color (normalmente RGB). Esta estructura permite que los modelos de inteligencia artificial realicen tareas como clasificación, detección de objetos, segmentación semántica, entre otras.

Herramientas Utilizadas

- **OpenCV** (Open Source Computer Vision Library) es una biblioteca poderosa y de código abierto orientada al procesamiento de imágenes y visión por computadora, ampliamente adoptada tanto en proyectos académicos como industriales. Su estructura optimizada permite realizar transformaciones geométricas complejas, operaciones en tiempo real con video o cámaras en vivo, y aplicar algoritmos avanzados de detección como Canny, detección de objetos o reconocimiento facial. Su compatibilidad con lenguajes como Python y C++, junto a su integración fluida con frameworks de aprendizaje profundo como TensorFlow y PyTorch, la convierte en una opción versátil para el desarrollo de soluciones basadas en inteligencia artificial.

- **Pillow** (PIL Fork) es una biblioteca de Python especializada en la manipulación sencilla de imágenes. Brinda herramientas intuitivas para realizar operaciones básicas como recorte, rotación, redimensionamiento, aplicación de filtros y conversión entre distintos formatos de imagen (JPEG, PNG, BMP, entre otros). Está diseñada para integrarse fácilmente en scripts, notebooks interactivos (como los de Google Colab) y entornos de prototipado rápido, lo cual la vuelve ideal para tareas cotidianas de procesamiento de imágenes, especialmente en las primeras etapas del desarrollo o en aplicaciones donde la simplicidad y rapidez son esenciales. Pillow también puede complementar a OpenCV cuando se necesita una manipulación más directa de los archivos de imagen.

Repositorio

El repositorio asociado a este proyecto contiene el código fuente implementado en Google Colab, así como una copia del presente informe. Se incluyen todos los scripts necesarios para ejecutar las transformaciones sobre imágenes, junto con instrucciones básicas para su uso. Puede accederse al repositorio a través del siguiente enlace: Link

Metodología

Se utilizó un entorno de trabajo en Google Colab, donde se implementó un script interactivo para cargar una imagen, aplicar distintas transformaciones, visualizar los resultados y descargar las imágenes modificadas.

Las operaciones disponibles fueron:

1. **Rotación**
2. **Recorte**
3. **Desenfoque gaussiano**
4. **Detección de bordes (Sobel/Scharr)**
5. **Binarización**
6. **Detección de bordes con Canny**

Cada operación se ejecuta a través de un menú interactivo, permitiendo elegir qué transformación aplicar y guardando automáticamente cada imagen resultante.

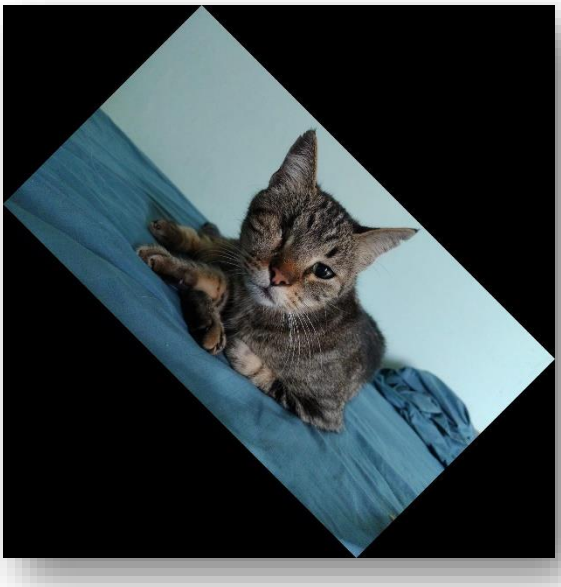
Se implementaron contadores para diferenciar múltiples versiones de imágenes modificadas, esto permite una fácil identificación de las imágenes que, además se pueden visualizar y descargar.

Resultados de la Ejecución

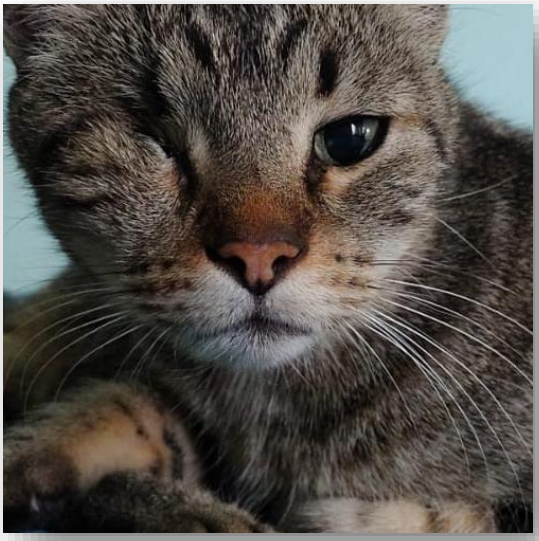
El script fue ejecutado y se pudo obtener en primera instancia, la visualización de la imagen seleccionada junto con su información técnica: resolución de 899x1599 píxeles y el tipo de dato. Luego, se muestra el menú para aplicar diversas transformaciones utilizando funciones implementadas con las bibliotecas solicitadas:

A lo largo de la ejecución, se aplicaron las siguientes transformaciones a la imagen original:

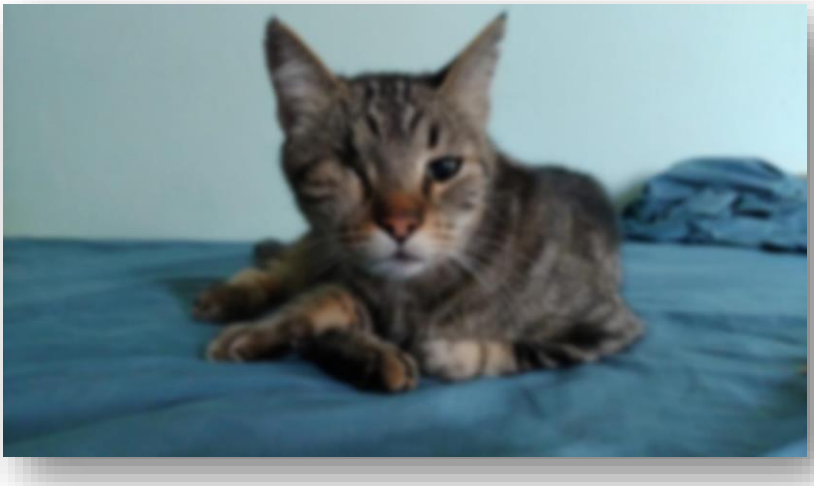
- Rotación: Se rotó la imagen para modificar su orientación, ajustando automáticamente el lienzo para conservar la integridad del contenido visual.



- Recorte: El menú brinda la posibilidad de establecer los puntos de corte de forma manual o recortar la imagen desde el centro; en este caso se utilizó la segunda opción, conservando únicamente la parte de interés. Esta operación es útil en tareas de enfoque o eliminación de ruido visual.



- Desenfoque Gaussiano: El menú de opciones permite establecer el radio de desenfoque que se desea aplicar, en este caso en particular se utilizó el radio 5. Este filtro de desenfoque sirve para suavizar detalles, es útil en el preprocesamiento para reducir ruido antes de aplicar técnicas como la detección de bordes.



- Binarización: La imagen fue convertida a una versión en blanco y negro basada en un umbral, permitiendo distinguir claramente las áreas de interés.

- Detección de Bordes y detección de borden con Canny: Se ejecutaron ambos algoritmos, que permiten resaltar los contornos más relevantes de la imagen original.

Finalmente, mediante otro menú de opciones se permite visualizar y descargar las imágenes en un archivo comprimido que se adjunta en el repositorio, y contiene todas las imágenes utilizadas en el proyecto, tanto las modificadas como la original.

Conclusiones

El uso combinado de OpenCV y Pillow permitió desarrollar un flujo completo de procesamiento de imágenes, desde la carga hasta la modificación y descarga de resultados. Ambas bibliotecas demostraron ser complementarias: mientras OpenCV destaca por su potencia en tareas complejas, Pillow facilita operaciones más sencillas con gran eficiencia.

Esta experiencia práctica refuerza la importancia del procesamiento de imágenes en aplicaciones actuales de inteligencia artificial, especialmente en áreas como visión por computadora, análisis biométrico, diagnóstico médico por imagen, vigilancia y más.

Además, trabajar en un entorno interactivo como Google Colab facilitó la implementación, prueba y ejecución del proyecto, evidenciando su utilidad como herramienta educativa y de desarrollo.