

TAG READER

Procesamiento de Imágenes con Python

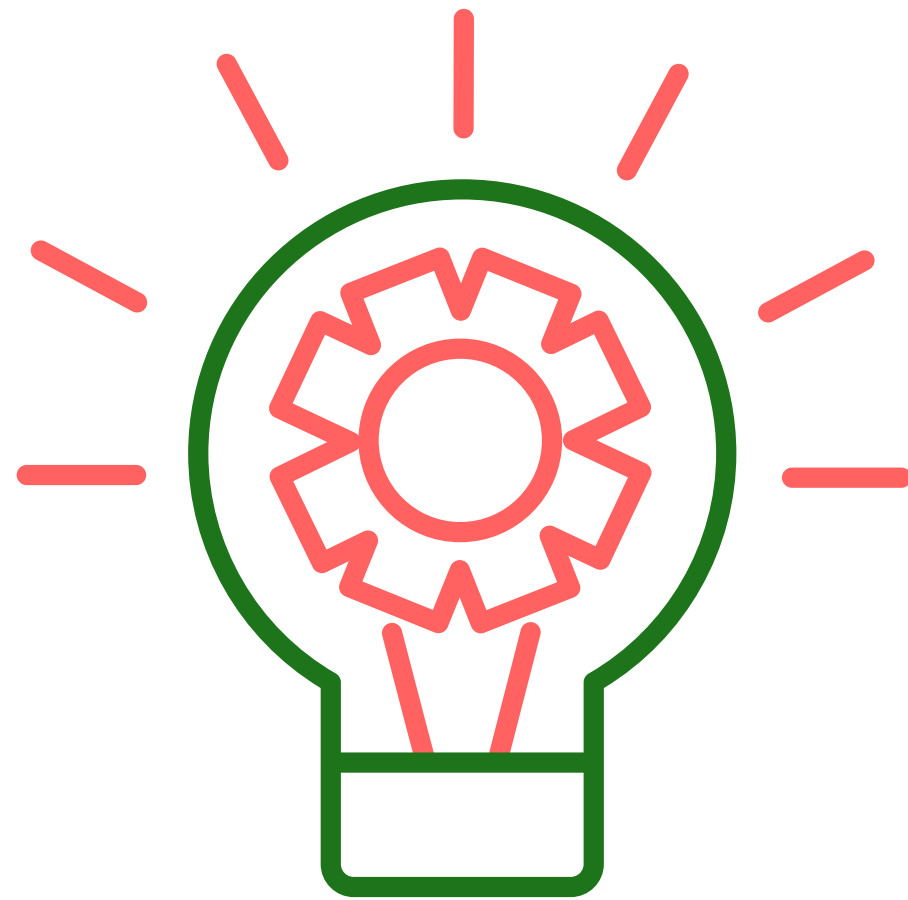
OMAR ÁVILA

PROBLEMÁTICA

Las empresas suelen manejar inventarios de productos electrónicos donde cada unidad posee etiquetas con información como marca, modelo y número de serie. La lectura manual de estos datos es lenta y propensa a errores.



EL PROYECTO



TagReader surge como una solución que ayuda a todas las empresas que dispongan de un manejo de inventarios. Este proyecto busca automatizar el proceso, extrayendo de manera rápida y confiable la información relevante directamente desde imágenes de las etiquetas.

Se trata de un script en Python que procesa imágenes de etiquetas mediante técnicas de procesamiento de imagen para resaltar el texto. Posteriormente, se utiliza OCR (reconocimiento óptico de caracteres) para extraer la información de interés. Los datos extraídos se organizan y almacenan en un archivo CSV, generando un registro único por producto.

Librerías



OpenCv

Para leer y procesar imágenes, aplicar máscaras y filtros de color.



NumPy

Para manipulación de matrices y arrays de imagen.



Pytesseract

Para reconocimiento de texto en las imágenes.



Re/RegEx

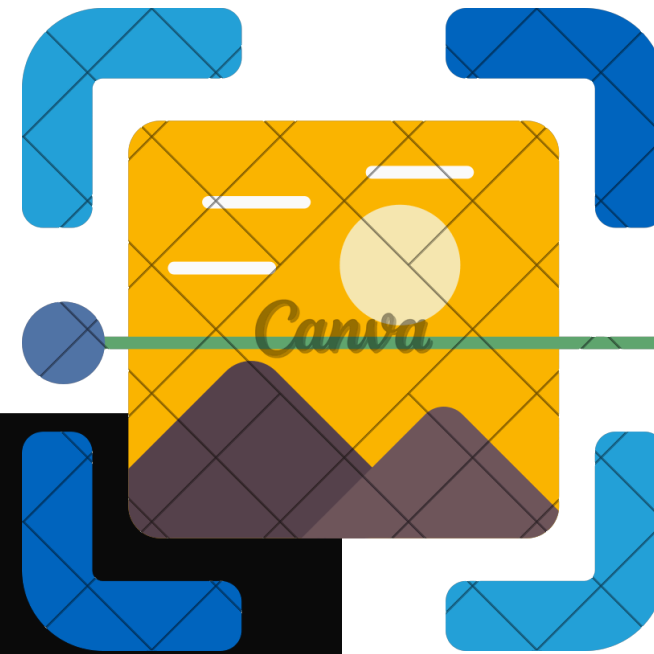
Para extraer patrones específicos de texto mediante expresiones regulares.



Csv

Para guardar y manejar los registros de productos en un archivo CSV.

Procesamiento de Imagen (filtros)



```
def read_image(image_path):  
    # Leer imagen y convertir a RGB  
    img = cv2.imread(image_path)  
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
  
    # Invertir colores de la imagen  
    img_inverted = cv2.bitwise_not(img_rgb)  
  
    # Definir rango de color blanco para crear máscara  
    blanco_bajo = np.array([190, 190, 190])  
    blanco_alto = np.array([255, 255, 255])  
    mask = cv2.inRange(img_inverted, blanco_bajo, blanco_alto)  
  
    # Aplicar máscara para resaltar la etiqueta  
    result_white = cv2.bitwise_and(img_inverted, img_inverted, mask=mask)  
  
    # Extraer texto usando pytesseract OCR  
    product_string = pytesseract.image_to_string(result_white)  
    return product_string
```

Se cambia el formato de la imagen RGB

Se invierte el color para tener el texto en blanco y poder utilizar máscaras

Se utiliza una máscara para que después pytesseract identifique el texto

Extracción de datos

- Detección de marca: Se compara el texto extraído con una lista de marcas conocidas para identificar la marca del producto.
- Detección de modelo: Se buscan patrones específicos en el texto que indiquen el modelo del producto.
- Detección de número de serie: Se identifican los patrones estándar de S/N o SN para extraer el número de serie.
- Almacenamiento estructurado: Cada dato (marca, modelo, número de serie) se organiza en un registro único listo para guardar en un inventario digital.



ENTRADAS



SALIDAS



ID	MARCA	MODELO	S/N
1	HUAWEI	KLVL-W76W	C5M8D196B9000134
2	HP	16-f0010ca	5CB1234FGH9012
3	SAMSUNG	NP960XFH	GSC1A2B3HGH5K67

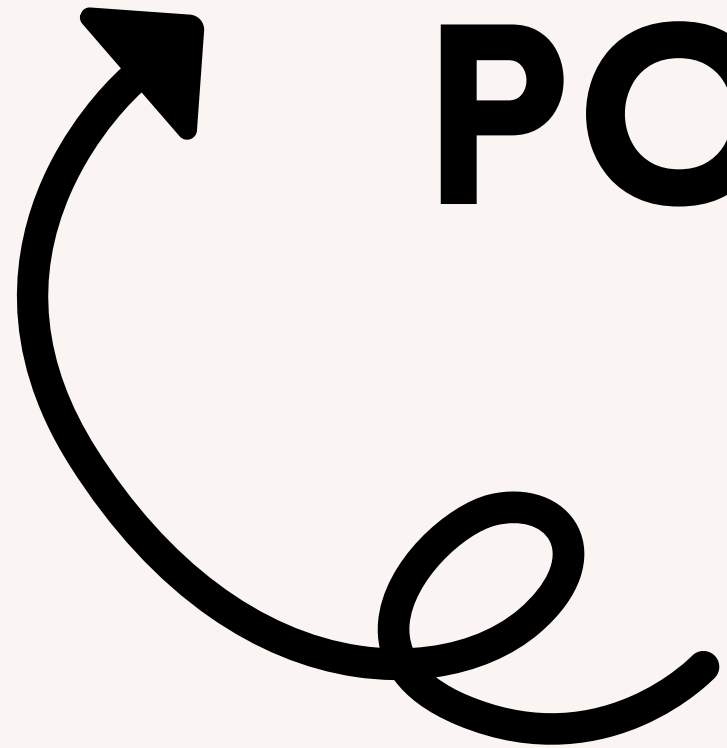
Conclusiones

Este proyecto demuestra cómo el procesamiento de imágenes puede automatizar tareas que antes eran manuales y propensas a errores, como la lectura de etiquetas de productos. La extracción de información mediante OCR y el filtrado de texto permiten registrar datos de manera rápida y precisa, mejorando la gestión de inventarios.

Más allá de este caso, las técnicas utilizadas son aplicables en diversos escenarios, desde control de calidad en producción hasta digitalización de documentos, mostrando el potencial del procesamiento de imágenes para optimizar procesos y ahorrar tiempo en múltiples industrias.



**¡MUCHAS GRACIAS
POR SU ATENCIÓN!**



TAGREADER