

Detección de pastillas faltantes

Modelado y procesamiento de imágenes

5to Semestre

Universidad La Salle Bajío

Greys Alexa Almanza Vega

Saray Alexandra Martínez

Dennise Chagoya Veloz

Contenido

Introducción	2
Desarrollo.....	2
Flujo de Trabajo.....	2
Objetivo.....	2
Flujo de procesamiento.....	2
Funciones Clave.....	3
Visualización con Streamlit.....	3
Aplicación	4
Conclusión.....	5
Bibliografía	6

Introducción

El procesamiento digital de imágenes permite analizar visualmente datos que pueden ser difíciles de detectar a simple vista. En este proyecto, desarrollamos una aplicación para detectar automáticamente la presencia o ausencia de pastillas en un blíster, utilizando operaciones morfológicas básicas programadas manualmente en Python. Esto es especialmente útil para evitar errores humanos en el conteo y asegurar la correcta administración de medicamentos.

Las imágenes fueron capturadas con un Xiaomi Poco C65, con un tamaño de 1600 x 897 píxeles, en formato JPEG, una distancia focal de 28mm y tomadas a una altura fija de 18.5 cm, siempre centrando el blíster en la pantalla para mantener uniformidad.

Desarrollo

Flujo de Trabajo

1. Carga de imagen.
2. Conversión a escala de grises.
3. Binarización invertida.
4. Operaciones morfológicas (Cierre).
5. Análisis de parches para detectar pastillas.
6. Visualización de resultados en una interfaz con Streamlit.

Objetivo

Detectar de manera automática la presencia o ausencia de pastillas en posiciones circulares en el blíster, mediante operaciones de procesamiento digital.

Flujo de procesamiento

El sistema se desarrolla en Python usando la librería Streamlit para la interfaz, y todas las operaciones morfológicas están implementadas de forma manual usando NumPy, para un mayor control y comprensión del funcionamiento.

Funciones Clave

1. binarizar(imagen_gris, umbral=127)

Convierte una imagen en escala de grises a binaria invertida: Esto hace que los objetos oscuros (como pastillas ausentes) aparezcan en blanco, facilitando su detección.

2. dilatar(imagen_binaria, kernel)

Esto agranda las regiones blancas (presuntas ausencias), cerrando huecos pequeños.

3. erosionar(imagen_binaria, kernel)

Reduce regiones blancas pequeñas: Se usa para eliminar falsos positivos generados por la dilatación.

4. cerrar(imagen_binaria, kernel_size=5, iteraciones=2)

Combina dilatación y luego erosión para realizar un cierre morfológico: Esta función suaviza los bordes y rellena huecos dentro de las regiones blancas.

5. tiene_pastilla(parche_gray, ...)

Esta función recibe un parche de imagen centrado en una región circular detectada, y a partir de la binarización y el cierre, evalúa la proporción de píxeles blancos (hueco). Si supera cierto umbral (por ejemplo, 30%), se considera que no hay pastilla.

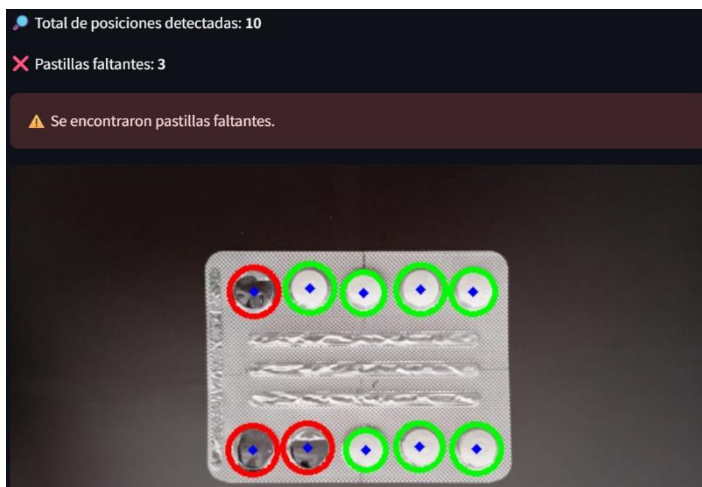
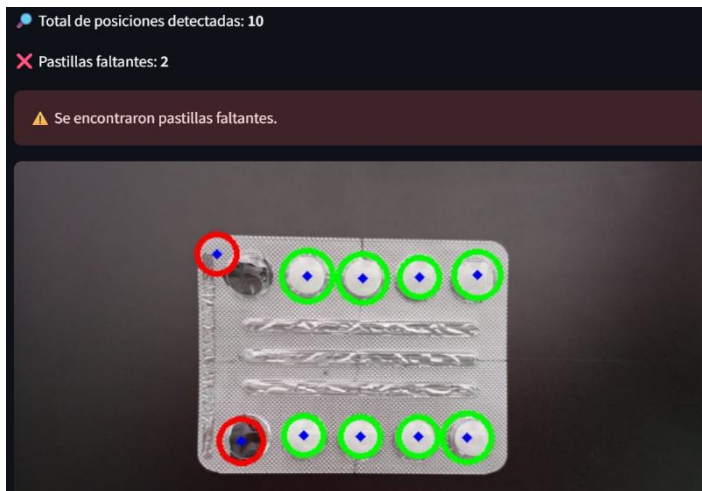
Visualización con Streamlit

La interfaz creada en Streamlit permite:

- Subir una imagen.
- Visualizar la imagen original.
- Detectar los círculos (posiciones de pastillas).
- Dibujar los resultados: verde (presente), rojo (faltante).
- Mostrar el conteo total y de faltantes.

Aplicación

<https://deteccion-pastillas-ny2bfvx3g8uqt9ta66ryut.streamlit.app/>



Conclusión

Este proyecto permitió comprender y aplicar técnicas de procesamiento de imágenes binarias como binarización, dilatación, erosión y cierre morfológico, de forma manual con NumPy. Estas funciones son esenciales en aplicaciones donde se requiere detección de formas simples, como blísteres con cavidades circulares.

La elección de una buena iluminación, el centrado de la imagen y el enfoque fueron factores críticos para la precisión. Las funciones manuales ayudaron a afinar los parámetros y comprender mejor el comportamiento morfológico.

En conjunto, esta aplicación puede ser la base para sistemas de control de calidad automatizados en la industria farmacéutica o médica, y constituye una demostración concreta del poder de las herramientas de procesamiento digital aplicadas a problemas reales.

Bibliografía

- mhadhbi, N. (2024, April 29). *Tutorial de Python: Streamlit*. Datacamp.com; DataCamp. <https://www.datacamp.com/es/tutorial/streamlit>
- <https://chatgpt.com/share/684100a9-6a08-8003-a50b-7a9596920c68>