



FACULTAD DE
CIENCIAS EXACTAS,
INGENIERIA Y AGRIMENSURA



Tecnicatura Universitaria en Inteligencia Artificial

Procesamiento de Imágenes

“TRABAJO PRÁCTICO N° 2”

Balverdi, Valentina - (B-6588/9)

Caballero, Franco - (C-7328/8)

Grimaldi, Damián - (G-5977/3)

1º Semestre - Año 2025

Problema 1 - Detección y clasificación de componentes electrónicos

Introducción al problema

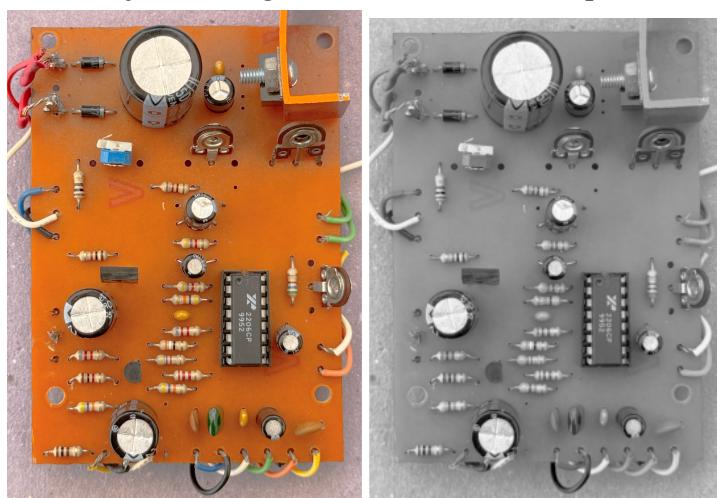
La imagen con la que se trabajó muestra una placa de circuito impreso (PCB) que contiene una variedad de componentes electrónicos soldados, formando un circuito eléctrico completo.

Para detectar los componentes destacados, se utilizaron las siguientes técnicas: filtro de mediana, detección de bordes con canny, operaciones morfológicas como dilatación y clausura, por último se utilizó, componentes conectados (`cv2.connectedComponentsWithStats()`).

Estas técnicas conforman la base del procedimiento realizado, permitiéndonos detectar correctamente los componentes, y clasificarlos de forma correcta.

Desarrollo y resolución

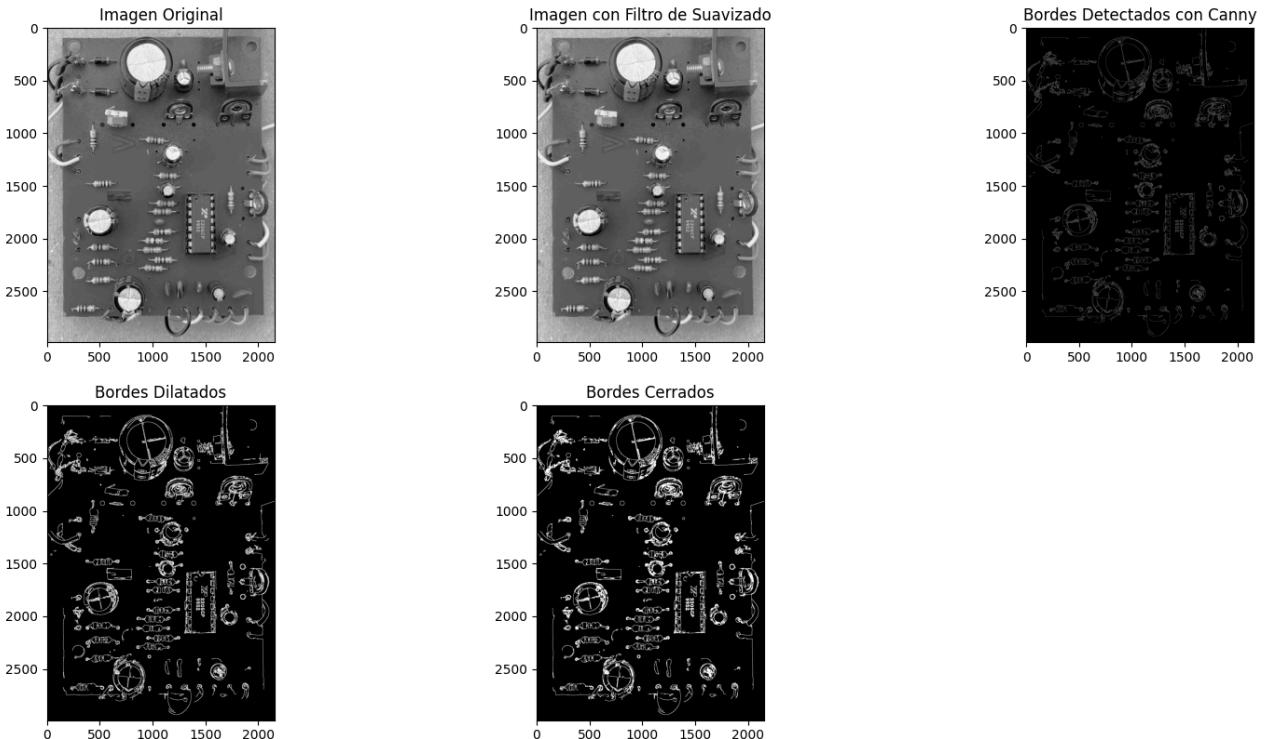
En primer lugar, pasamos a la escala de grises. Debemos detectar los objetos del enunciado y nos encontramos con que la imagen con niveles de grises, en el fondo de los objetos, los grises son similares lo que dificulta la segmentación.



Comparación imagen original e imagen en escala de grises

Una vez pasado a escala de grises proseguimos con la detección de bordes, a través de la función de `detectar_bordes()`, donde a la imagen en escala de grises, le

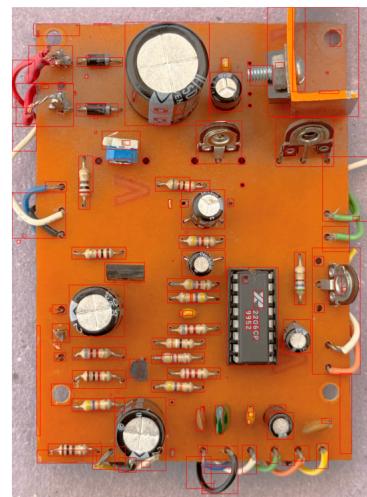
aplicamos un filtro para suavizar la imagen, que en este caso, utilizamos el filtro de mediana, un filtro pasa bajos no lineal. Este filtro reemplaza cada píxel por la mediana de los valores de intensidad de sus vecinos. Una vez aplicado, detectamos los bordes con canny, para posteriormente aplicarle dos operaciones morfológicas, dilatación y clausura. Con dilatación, lo que hicimos fue engrosar los bordes, para resaltar más las uniones, para luego aplicarle una clausura para afinar bordes y resaltar mejor las formas.



Procedimiento de detección de bordes.

Una vez realizado esto, pasamos a visualizar cómo detecta todos los componentes, para poder guiarnos de cómo segmentarlo.

Sabiendo cómo se detectan todos los componentes, procedimos con 3 funciones (`detectar_resistencia()`, `detectar_chips()` y `detectar_capacitadores()`), que se encarga de detectar



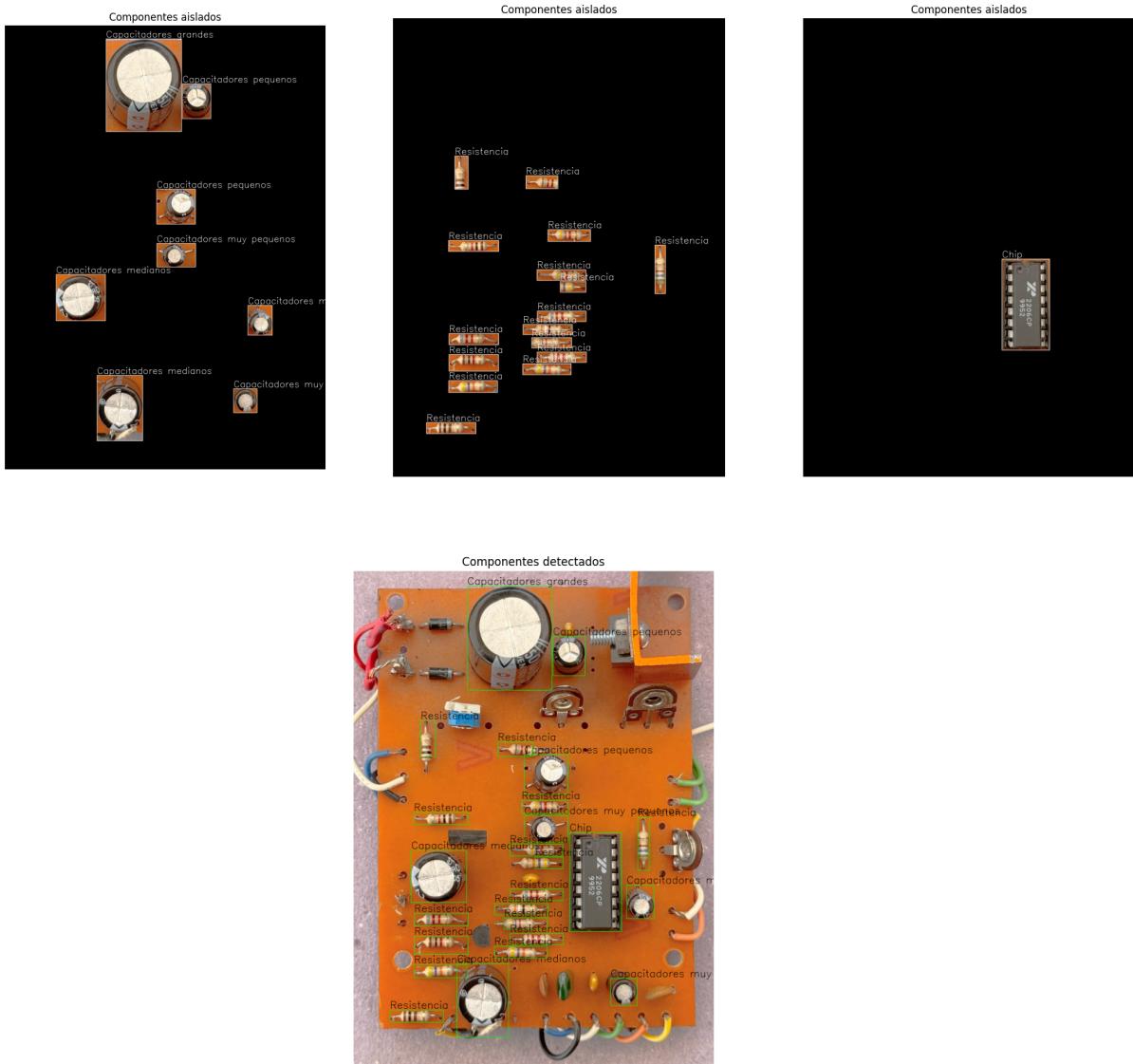
cada componentes. Las 3 funciones utilizadas, utilizan el mismo principio, utilizar de cv2, la función de connectedComponentsWithStats.

Con la función de detectar_chips(), a partir de los stats, obtenidos detecta, filtramos por un ancho y un alto para almacenar en una lista, con las coordenadas del mismo, y posteriormente aislarlo.

Después con la función de detectar_capacitadores(), se realizo el mismo procedimiento con chips, pero en su lugar, lo fuimos clasificando en: grandes, medianos, pequeños, y muy pequeños, y los almanecamos en una lista de cada uno, ademas de que tipo de capacitor es.

Con la función de visualizacion_componentes_aislados(), visualizamos cada uno de los componentes detectados de forma individual.

Por último generamos una imagen, con todas los componentes detectados, junto que componente es cada uno.



Conclusión

El algoritmo implementado demostró ser eficaz para la detección y clasificación de los distintos componentes electrónicos sobre la PCB. Al convertir la imagen a escala de grises y aplicar un filtro de mediana seguido de Canny, logramos resaltar los bordes; luego, con operaciones de dilatación y clausura, reforzamos las formas para facilitar la segmentación.

Gracias a la función de componentes conectados (`cv2.connectedComponentsWithStats()`), pudimos aislar cada elemento y, mediante criterios de tamaño y forma, distinguir resistencias, chips y capacitores (clasificándolos además según su tamaño). Finalmente, la visualización individual y la imagen compuesta con todas las detecciones confirman que el procedimiento cumple con los requisitos planteados.

Problema 2 - Identificación de resistencias eléctricas

Introducción al problema

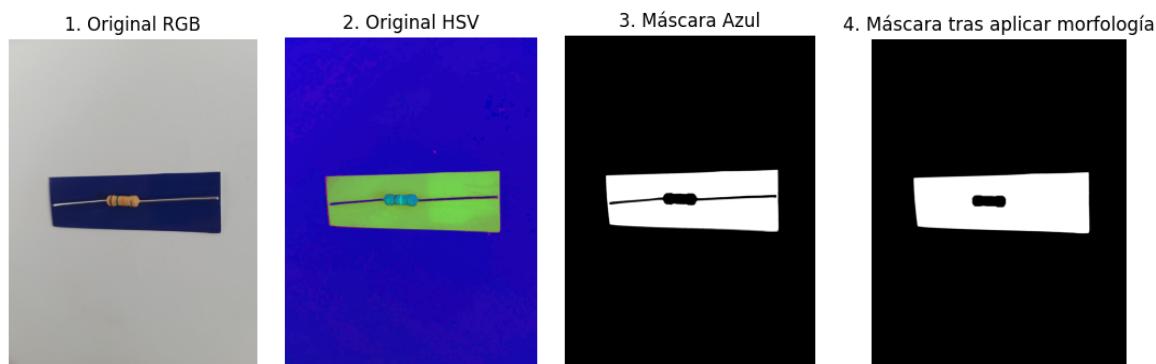
En este problema se aborda la elaboración de algoritmos en Python para la segmentación y clasificación de resistencias eléctricas a partir de fotos tomadas desde diferentes perspectivas. El conjunto de datos está compuesto por 40 fotografías de 10 resistencias distintas (cada una fotografiada en cuatro perspectivas diferentes). Cada imagen muestra una resistencia apoyada sobre un rectángulo azul con fondo blanco.

El objetivo final es, primero, transformar cada imagen para obtener una vista superior del rectángulo azul donde aparece la resistencia; luego, a partir de esas vistas superiores, localizar las tres bandas de color que definen el valor de la resistencia (ignorando la banda dorada que siempre se encuentra ligeramente separada); y, finalmente, traducir esos colores a dígitos y calcular el valor numérico en ohmios.

Desarrollo y resolución

En primera instancia, nos enfocamos en obtener la vista superior del rectángulo azul sobre el que se encuentran las resistencias.

Cada imagen original se convirtió a HSV para filtrar los píxeles azules del rectángulo. Tras aplicar cierre y apertura morfológica, se detectó el contorno de mayor área (el rectángulo), se aproximaron sus vértices a un polígono de cuatro esquinas y, por si no se encontraban únicamente cuatro vértices, se empleó como último recurso el bounding-box. Los cuatro puntos obtenidos se reordenaron (top-left, top-right, bottom-right, bottom-left) para luego calcular la homografía que endereza la vista y recorta la imagen al área del rectángulo. El resultado son 40 archivos “R{i}_j_out.jpg” con la resistencia en vista superior.



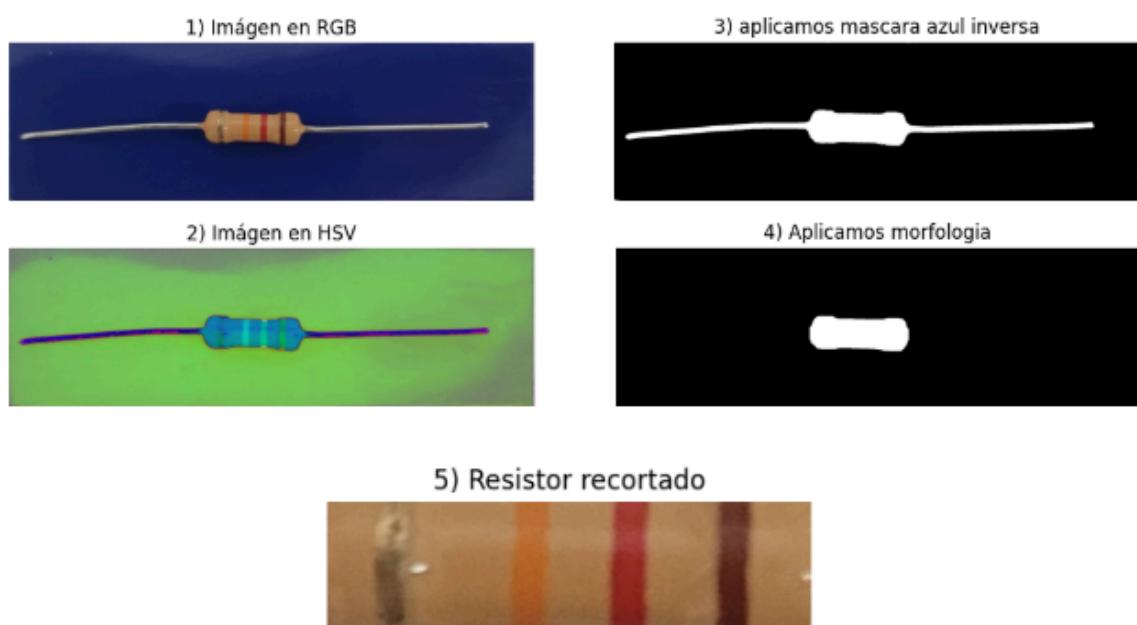
7. Área azul corregida (rectángulo)



La segunda parte consistió en detectar el valor de la resistencia mediante sus colores. Empezamos desde el uso de las imágenes llamadas “R{i}_a_out.jpg” y las sometimos a un procedimiento inicial muy similar al que empleamos con el rectángulo azul, la diferencia es que en este caso nos interesaba lo que no era azul y aplicamos una máscara para identificar esa zona. Luego utilizamos la misma morfología anteriormente aplicada pero con diferentes tamaños de kernels y luego de detectar contornos y un rectángulo que encuadre la parte importante de la resistencia hicimos un recorte.

La idea siguiente era emplear funciones para detectar las bandas pero tuvimos muchos intentos fallidos. Probamos detectando colores con HSV y tratamos segmentar con CIELAB, pero el dorado se confundía con el naranja y el amarillo. Intentamos descartar la franja dorada por su lejanía a las demás pero tampoco tuvimos éxito. Nuestra idea pendiente es trabajar con las imágenes en escala de grises, detectar líneas verticales para segmentar las bandas, quedarnos con las más cercanas entre sí (descartando la dorada) y ordenandolas según su lejanía a la banda restante. Luego usar otra función para detectar el color de las franjas segmentadas.

Finalmente, la función para detectar el valor de la resistencia a partir de los colores en formato string está hecha.



Conclusión

Se cumplió exitosamente la primera etapa: convertir cada imagen a espacio HSV, filtrar el rectángulo azul y aplicar operaciones morfológicas para extraer sus cuatro vértices; luego, mediante reordenamiento de puntos y cálculo de homografía, se obtuvo una vista superior uniforme de las 40 fotos. En la segunda etapa, se emplearon técnicas similares de enmascaramiento y morfología para aislar la resistencia dentro de ese rectángulo. Sin embargo, la segmentación final de las bandas de color presentó dificultades: los intentos con HSV y CIELAB no discriminaron adecuadamente el dorado del amarillo o naranja, y la lógica de descartar la banda dorada por su separación espacial quedó pendiente de refinamiento.

Conclusión final

En conjunto, en este trabajo se muestra cómo un mismo enfoque de procesamiento de imágenes (basado en filtrado, morfología y detección de contornos) puede adaptarse a dos casos prácticos distintos: la identificación de componentes en una PCB y la extracción del código de color de resistencias. En el primer caso, se logró aislar y clasificar cada elemento (resistencias, chips y capacitores) con alta confiabilidad, validando que las etapas de preprocesamiento generan mapas de contorno suficientemente nítidos para la agrupación mediante componentes conectados. En el segundo caso, se consiguió recortar y corregir la perspectiva de las resistencias para uniformizar las tomas desde cuatro ángulos distintos, quedando pendiente únicamente perfeccionar la segmentación de las franjas coloreadas para distinguir de manera robusta la banda dorada y ordenar las demás.