



Universidad Nacional de Rosario

Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Trabajo práctico N° 02

Tecnicatura Universitaria en Inteligencia Artificial

Procesamiento de imágenes (IA4.4)

Integrantes	Legajo	Mail
Porcelli, Fabricio	P-5340/6	fabricioporcelli@gmail.com

Fecha de entrega: 09/06/2025

Docentes: Gonzalo, Sad - Julián, Álvarez - Juan Manuel, Calle



ÍNDICE

1. Introducción	Página 3
2. Primer problema - Detección de componentes.....	Página 3-9
3. Segundo problema - Valores de resistencias.....	Página 9-14
4. Conclusión.....	Página 15

1. Introducción

En este informe se explicará cómo se abordaron los problemas planteados utilizando diferentes estrategias vistas hasta el momento en la materia. Para cada solución, se comentarán los inconvenientes que surgieron durante el desarrollo, las modificaciones realizadas y la forma en que se resolvieron.

Además, se incluirán imágenes que faciliten la comprensión de los resultados obtenidos y en algunos casos, se presenta un ejemplo de su funcionamiento. El trabajo fue realizado en un repositorio de GitHub, que contenía un archivo README con instrucciones para poder ejecutar el programa paso a paso, dos archivos .py con las soluciones a los problemas dados y los archivos proporcionados por la cátedra.

2. Primer problema - Detección de componentes

Descripción del problema:

El problema consistía en identificar componentes electrónicos en una imagen .png de una placa: las resistencias, un chip de circuito integrado y los capacitores electrolíticos. Para lograr esto, se desarrollaron múltiples algoritmos de procesamiento de imágenes en Python utilizando la biblioteca OpenCV. Cada tipo de componente requirió técnicas específicas para su detección debido a sus características particulares (forma, tamaño y color).

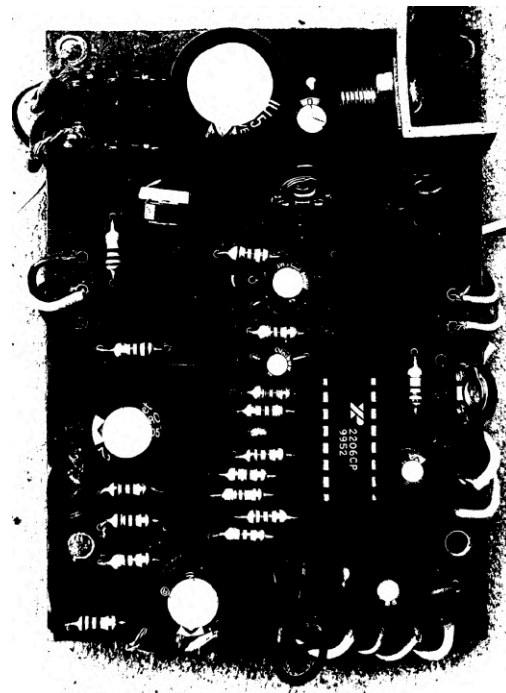
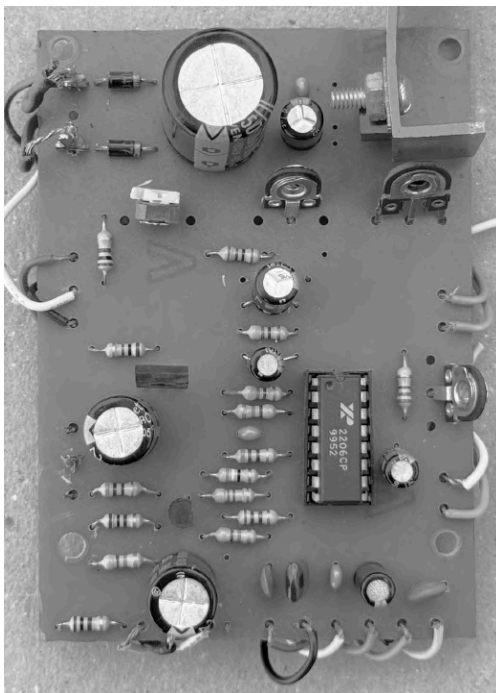
Resolución del problema:

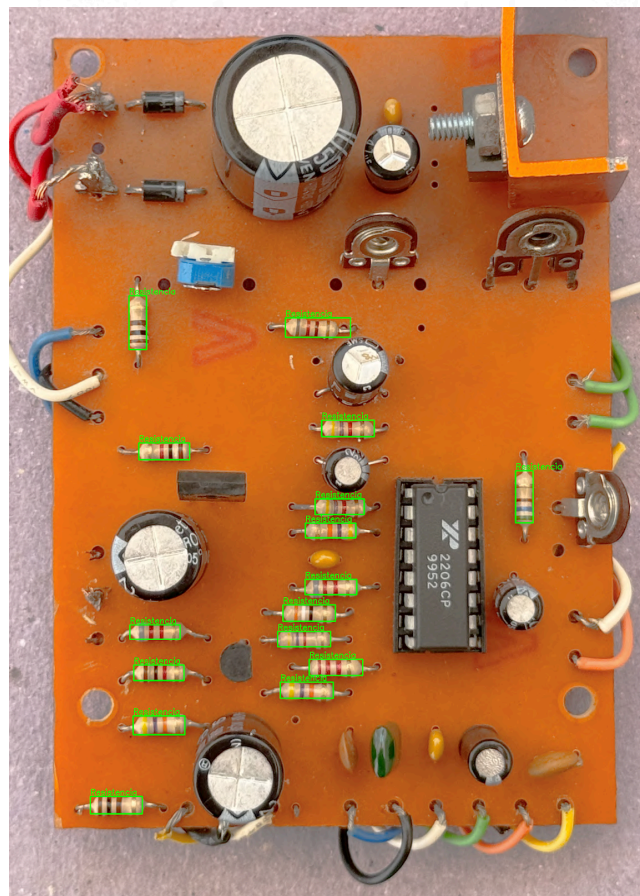
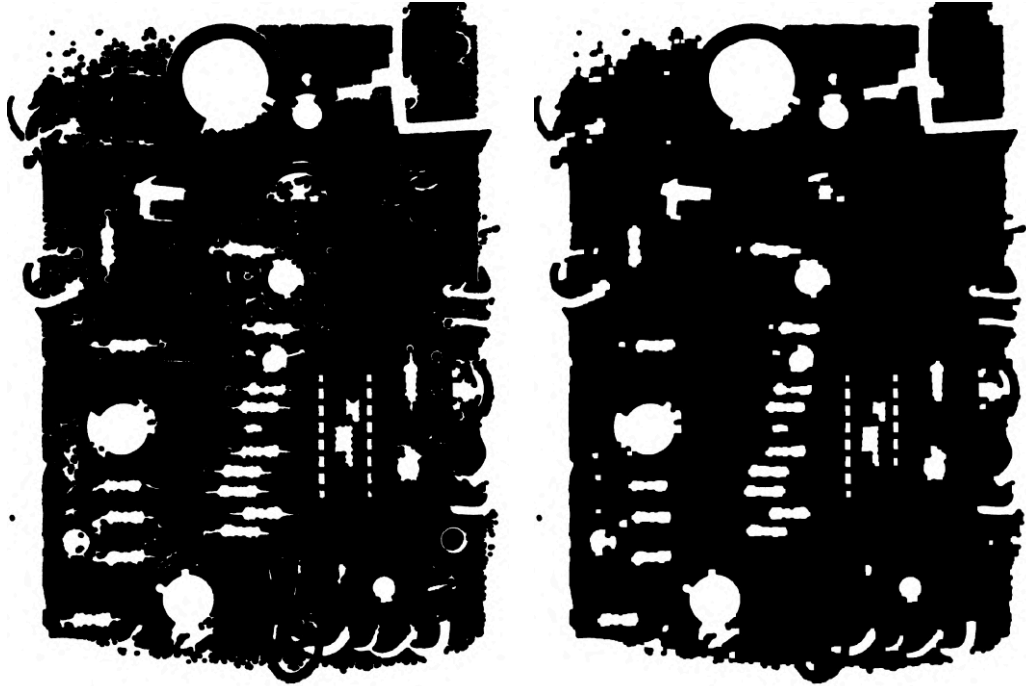
Para resolver este problema se utilizaron las librerías de OpenCV(cv2), Numpy(numpy), Matplotlib (matplotlib.pyplot) y el módulo KMeans de scikit-learn.

Proceso utilizado para la detección de las resistencias:

- **Conversión a escala de grises:** Se utilizó la función `cv2.cvtColor` para transformar la imagen a escala de grises, reduciendo la información de color y facilitando el procesamiento.
- **Umbralización adaptativa:** Con `cv2.threshold` y el método Otsu, se generó una máscara binaria para destacar áreas relevantes.
- **Operaciones morfológicas:** Se aplicaron cierres y aperturas con `cv2.morphologyEx` para eliminar ruido y mejorar la segmentación de las resistencias.
- **Detección de contornos:** Mediante `cv2.findContours`, se identificaron las áreas delimitadas por los componentes.
- **Filtrado por proporción y área:** Se consideraron sólo los contornos que cumplieran con criterios específicos de tamaño y relación de aspecto.

Imágenes del proceso en orden.

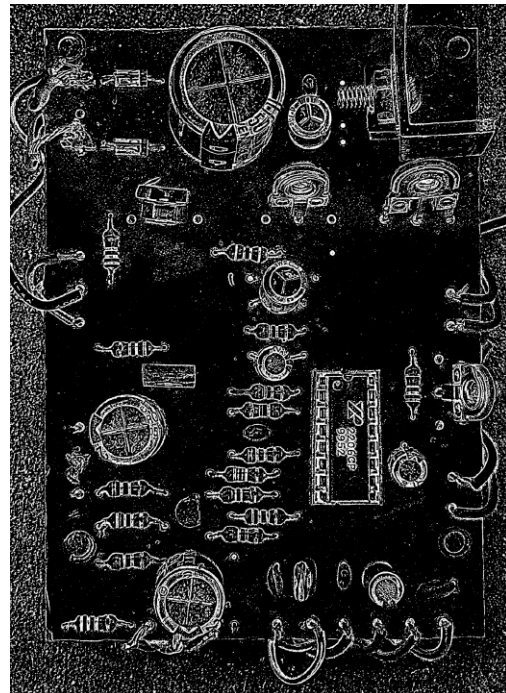
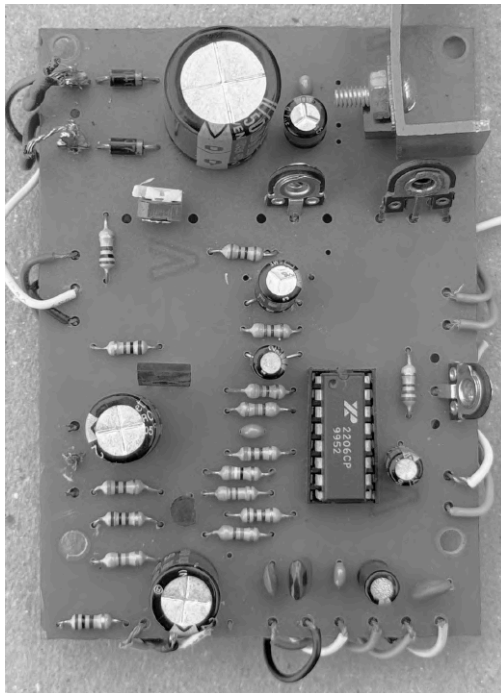


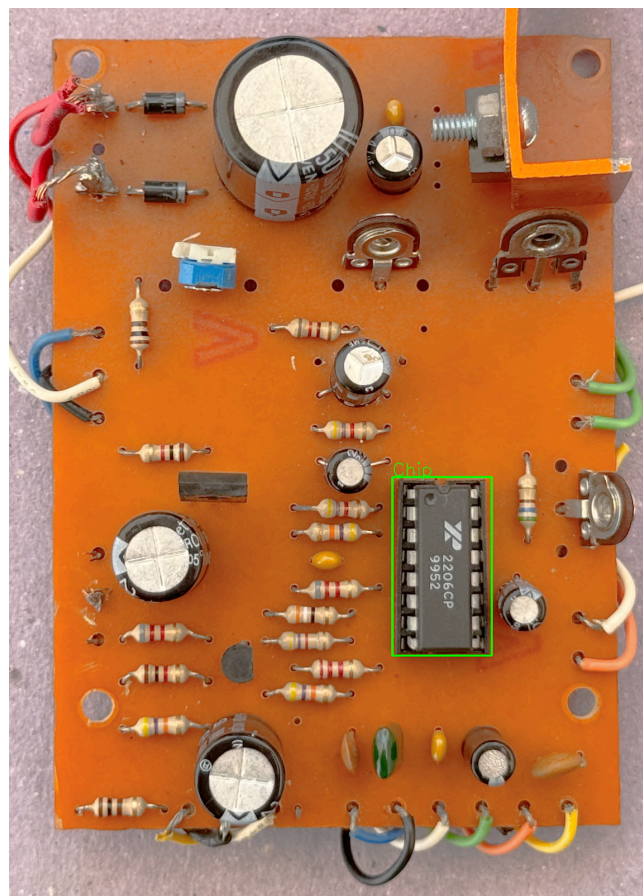
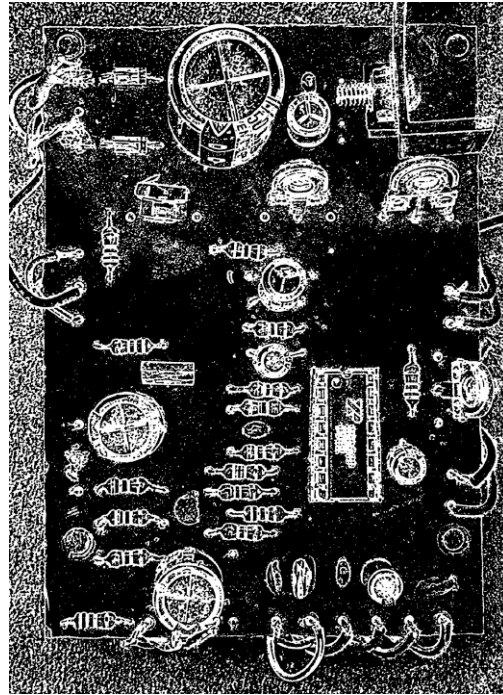
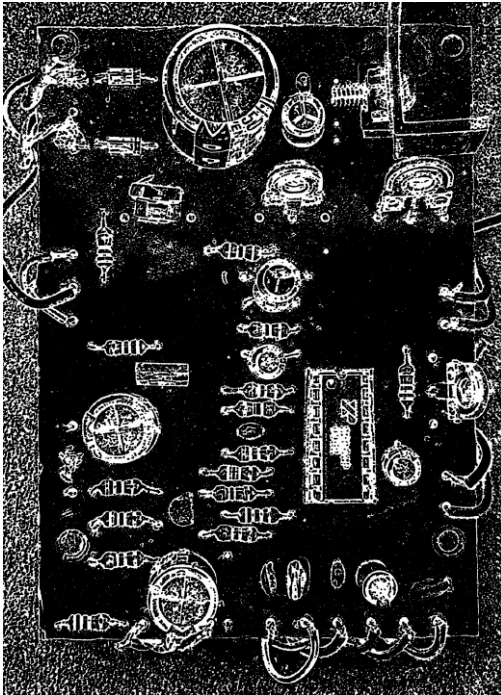


Proceso para la detección del chip de circuito integrado:

- **Umbralización adaptativa:** Se utilizó `cv2.adaptiveThreshold` con un enfoque gaussiano para manejar mejor la iluminación desigual.
- **Cierre morfológico:** Se aplicó un cierre para unir regiones discontinuas del chip.
- **Dilatar regiones:** Con `cv2.dilate`, se expandieron las áreas relevantes para asegurar una segmentación más precisa.
- **Detección de componentes conexos:** Se utilizó `cv2.connectedComponentsWithStats` para identificar regiones separadas.
- **Clasificación por área y proporción:** Se consideró el tamaño del chip y su relación de aspecto para diferenciarlo de otros componentes.

Imágenes del proceso en orden.

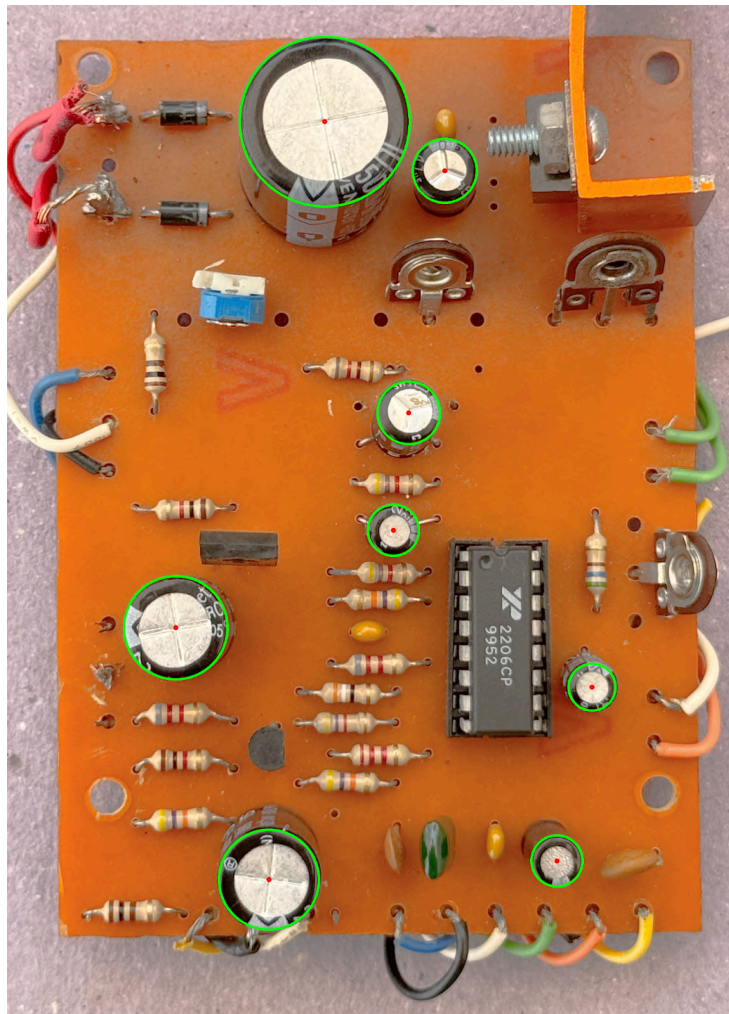




Proceso para la detección de los capacitores electrolíticos:

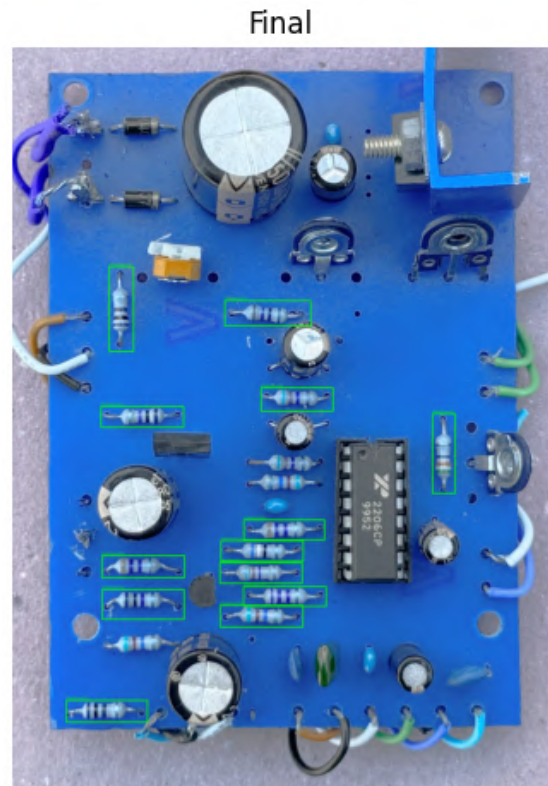
- **Filtro de mediana:** Se aplicó un filtro para reducir el ruido en la imagen antes de detectar círculos.
- **Transformada de Hough:** Se utilizó cv2.HoughCircles para detectar los capacitores en la imagen, basándose en su forma circular.
- **Dibujo de círculos:** Se marcaron los capacitores detectados en la imagen de salida con círculos verdes.
- **Clasificación por tamaño:** Se realizó un análisis adicional para agrupar los capacitores según su radio utilizando KMeans de la biblioteca scikit-learn.

Imágenes de los capacitores detectados.



Problemas que tuvimos hasta encontrar la solución:

- Ajustar los valores de umbralización para diferentes condiciones de iluminación.
- Eliminar artefactos o componentes que no eran resistencias pero cumplían con los criterios iniciales, como por ejemplo, los diodos de la parte superior izquierda de la placa.
- Encontrar un rango de tamaño óptimo para identificar el chip sin incluir falsos positivos. Se detectaban rectángulos del mismo tamaño y proporciones parecidas en el disipador y otras zonas de la placa.
- Ajustar los parámetros de la transformada de Hough (umbral y rango de radio) para detectar todos los capacitores sin incluir falsos positivos. De igual manera se detectaban círculos en los cables curvados al borde de la placa y en los potenciómetros preset.
- La falta de detección de varias resistencias que estaban muy cerca de otros componentes.



3. Segundo problema - Valores de resistencias

Descripción del problema:

El objetivo de este ejercicio fue identificar y calcular los valores de resistencias eléctricas a partir de un conjunto de imágenes que mostraban resistencias sobre un fondo azul. El proceso incluye etapas como la transformación de la

imagen para recortar el fondo azul, la segmentación de la resistencia, la detección de bandas de colores, la verificación de orientación y el cálculo del valor en ohms de acuerdo con el código de colores comercial.

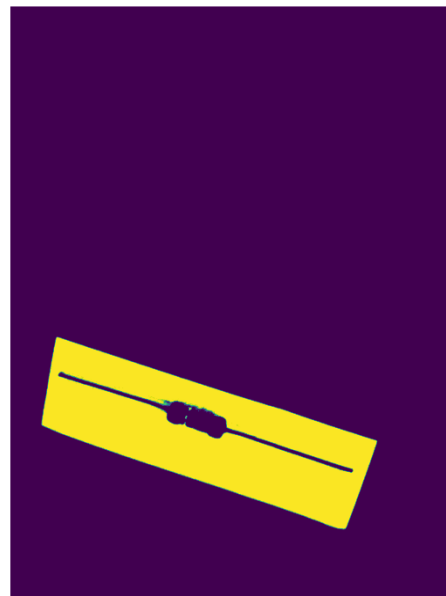
Resolución del problema:

El problema fue subdividido en 4 etapas para una mejor organización

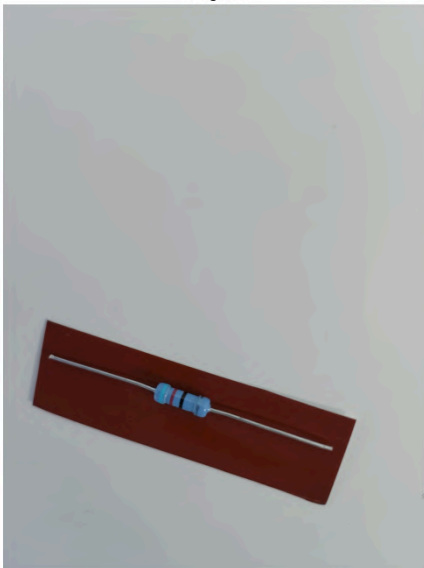
Primera etapa: Recorte del rectángulo azul con la resistencia

- **Mascara de color azul:** Se generó una máscara para aislar las regiones azules en cada imagen utilizando valores de HSV ajustados a tonos de azul fuerte.
- **Transformación en perspectiva:** A partir de los contornos detectados o de un rectángulo de mínimo área, se ajustó la imagen para mostrar la resistencia en una vista superior.

Mascara azul



Original



Transformada



Dificultades presentadas en la primer etapa:

- Ajustar los valores del rango HSV para diferentes condiciones de luz y variaciones en el color del fondo.
- Manejar casos en los que no se detectaban contornos precisos debido a bordes difusos o cortados por el cuerpo o patas de la resistencia en sí. Para esto se aplicaron dos métodos (en caso de fallar el primero, se utiliza el segundo):
- “contorno_preciso” se intenta detectar un contorno rectangular preciso que encierre la región azul. Este método funciona bien cuando los bordes de la región azul son claros y están bien definidos. Ej:



- “fallback_min_area_rect” usa un enfoque más general para encontrar un rectángulo que encierra la región azul. Este método se utiliza si no se encuentran polígonos válidos en el método principal. Se recorta por las coordenadas de las puntas del rectángulo azul. Ej:



Segunda etapa:

a. Segmentación del cuerpo de la resistencia

- **Umbralización por HSV:** Se utilizó un rango ajustado para aislar los colores del cuerpo de la resistencia.
- **Operaciones morfológicas:** Se aplicaron operaciones de cierre y apertura para eliminar ruido y unir regiones.
- **Detección de contornos:** Se seleccionó el contorno más grande que representaba el cuerpo de la resistencia.

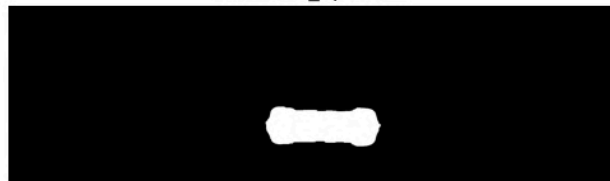
maska_umbrales



maska_clausura



maska_apertura



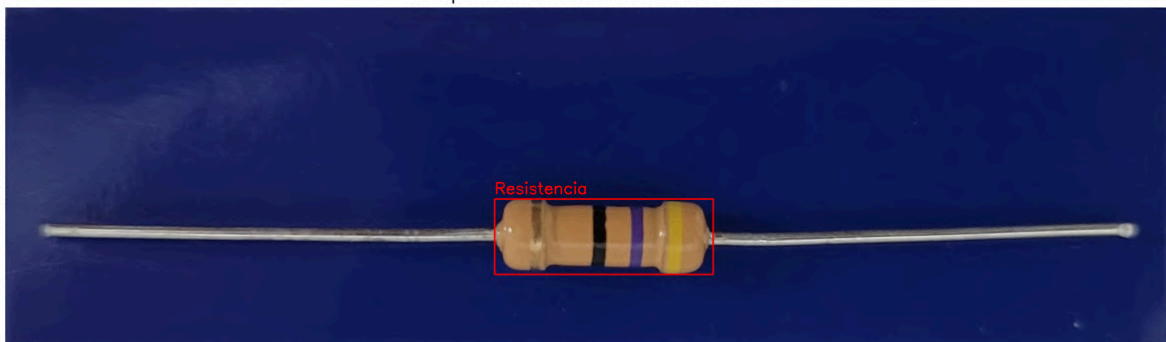
output



roi

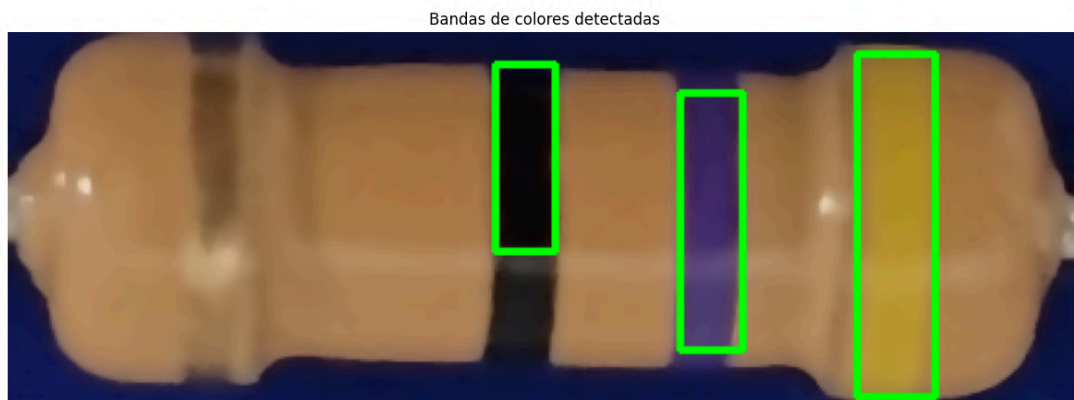


Cuerpo de la resistencia detectado -> ROI



b. Detección de bandas de colores

- **Filtrado de colores por rango HSV:** Se definieron rangos HSV específicos para cada color presente en las bandas.
- **Identificación de contornos:** Se procesaron las máscaras generadas para detectar las bandas como regiones discretas.
- **Clasificación por posición horizontal:** Se ordenaron las bandas detectadas para su correcta interpretación.



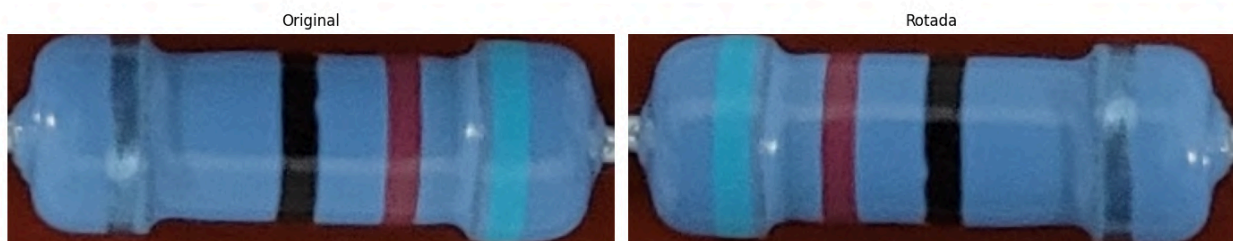
Dificultades encontradas en la segunda etapa:

- Manejar las variaciones en los colores de las bandas debido a reflejos o sombras. Esto ocasiona que en la misma banda se detecten varias partes cortadas. Pasar la imagen por un `cv2.bilateralFilter` para suavizar un poco los colores y determinar un valor de área mínima para evitar falsos positivos ayudó en la detección de las bandas.
- Se tuvo que adecuar los rangos HSV de algunos colores para generar una mejor detección de las bandas. Principalmente el negro, que por la iluminación y los reflejos se tornaba un poco gris.

c. Verificación de orientación de la resistencia

- **Posiciones de las bandas:** Se analizó la distribución horizontal de las bandas. Si la mayoría de bandas estaban en la parte derecha de la imagen, debe rotarse.

- **Orden de los colores:** Si la banda debe rotarse, el orden de los colores en la lista debía invertirse para que coincida con la imagen y el cálculo sea correcto.



Dificultades encontradas en la tercer etapa:

- No hubo dificultades mayores. La rotación se aplicó luego de haberse detectado los colores de las bandas, lo cual ayudó a ignorar la banda de tolerancia y enfocarse en la posición horizontal relativa de las tres bandas.

Tercera etapa: Cálculo del valor de las resistencias

- **Interpretación de los colores:** Se utilizaron diccionarios para asignar valores a cada color según el código de colores comercial.
- **Cálculo del valor:** Se aplicó la fórmula estándar para determinar el valor en ohmios, considerando las dos primeras bandas como coeficiente y la tercera como multiplicador.
- **Conversión de unidades:** Los valores fueron convertidos automáticamente a kOhms o MOhms si eran lo suficientemente grandes.

Cuarta etapa: Función principal

Recibe por parámetro un directorio con las imágenes a procesar y recorre buscando aquellas imágenes que coincidan con el formato de nombre establecido utilizando expresiones regulares. Devuelve una lista de tuplas con tres valores, el nombre de la imagen, una lista con los colores y el valor de la resistencia en ohms.

4. Conclusión

El desarrollo de los ejercicios evidencia la gran versatilidad y utilidad de las técnicas de procesamiento de imágenes. Estas herramientas permiten extraer información precisa y significativa incluso de imágenes complejas y detalladas, donde coexisten múltiples colores, formas y objetos.

En el primer ejercicio, se trabajó con una imagen de una placa con diversos componentes electrónicos. A pesar de la diversidad y complejidad visual de la escena, fue posible detectar y clasificar elementos específicos como resistencias, integrados y capacitores. Este proceso mostró cómo el ajuste de parámetros y el uso de técnicas apropiadas, como el análisis de contornos, transformadas de Hough y umbralización, pueden simplificar una tarea inicialmente desafiante.

En el segundo ejercicio, centrado en las resistencias eléctricas, se exploró la posibilidad de desarrollar un enfoque genérico y reutilizable para identificar las bandas de colores, calcular sus valores según un estándar comercial y ajustar la orientación para facilitar la lectura. Este ejercicio destacó la potencia del procesamiento de imágenes en aplicaciones específicas y repetitivas, mostrando que es posible automatizar tareas técnicas con precisión y eficiencia.

En conjunto, ambos ejercicios resaltan el potencial del procesamiento de imágenes para resolver tareas técnicas de forma eficiente y adaptable. La capacidad de simplificar y transformar las imágenes abre un sinfín de posibilidades para extraer datos útiles en distintos contextos, demostrando que estas herramientas no solo son versátiles, sino también indispensables para automatizar y optimizar procesos visuales.