# 

# Trabajo Práctico 2

## Lucía Masciangelo, Francisco Rodríguez y Barros, Julieta Texier

## 11-06-2024

## Procesamiento de Imágenes I

## Tecnicatura en Inteligencia Artificial

## Requisitos previos para correr los ejercicios:

1. Clonar el repositorio <https://github.com/franciscoryb1/PDI-TUIA-RodriguezYBarros>
2. Acceder a la carpeta: cd .\TP2
3. Tener python instalado en el entorno a utilizar. Se puede descargar desde <https://www.python.org/downloads/>
4. Instalar las librerías que serán usadas en los ejercicios:
   * OpenCV: pip install opencv-python
   * Matplotlib: pip install matplotlib
   * numpy: pip install numpy

## Ejercicio 1: Detección y clasificación de componentes electrónicos

Primero importamos las librerías y creamos una función para mostrar las imágenes (imshow).

Cargamos la imagen 'Placa.png' que se encuentra en la misma carpeta donde va a correr el script.

Convertimos la imagen a escala de grises, aplicamos el filtro de suavizado MedianBlur y probamos con varios umbrales de Canny para detectar bordes. (elegimos los umbrales 0.2\*255 y 0.8\*255 para encontrar el chip y los capacitores). Después aplicamos Gradiente morfológico con un kernel de (5,5).

Y obtuvimos los siguientes resultados:

Texto

Descripción generada automáticamente con confianza baja

Después calculamos los componentes conectados y aplicamos bounding box para encontrar el chip por un lado y los capacitores por otro.

Una vez que tenemos detectado el chip lo guardamos en una imagen nueva llamada chip.

Un circuito electrónico

Descripción generada automáticamente con confianza media

Hacemos lo mismo con los capacitores, pero sumándole una etiqueta que identifique que tamaño tiene el capacitor (que también guardamos en una imagen llamada capacitores según tamaños, y otra imagen que sea solo los capacitores)

Un circuito electrónico

Descripción generada automáticamente con confianza mediaPantalla de video juego

Descripción generada automáticamente con confianza media

Ahora, para poder detectar las resistencias sin problemas vamos a “eliminar” de la imagen original a los capacitores y al chip, para eso creamos una imagen negra y con componentes conectados detectamos la parte de la imagen donde se encuentran el chip y los capacitores, y después pintamos de la imagen original de negro esos pixeles

Imagen que contiene circuito, computadora

Descripción generada automáticamente

Con esta imagen obtenida volvemos a hacer el proceso de convertir la imagen a escala de grises, binarizamos la imagen, aplicamos el filtro de suavizado MedianBlur y probamos con varios umbrales de Canny para detectar bordes (elegimos los umbrales 0.2\*255 y 0.6\*255 para encontrar las resistencias) aplicamos clausura con el kernel (20,1) y por último, aplicamos Gradiente morfológico con un kernel de (5,5).

Diagrama

Descripción generada automáticamente

Después calculamos los componentes conectados y aplicamos bounding box para encontrar las resistencias.

Un circuito electrónico

Descripción generada automáticamente con confianza media

Por último, agregamos las resistencias a la imagen con fondo negro en la que ya tenemos el chip y los capacitores y creamos otra solo con las resistencias, en ese mismo código pusimos un contador para después imprimir por consola la cantidad de resistencias encontradas.

En resumen, la salida del ejercicio es:

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Ejercicio 2: Detección de patentes