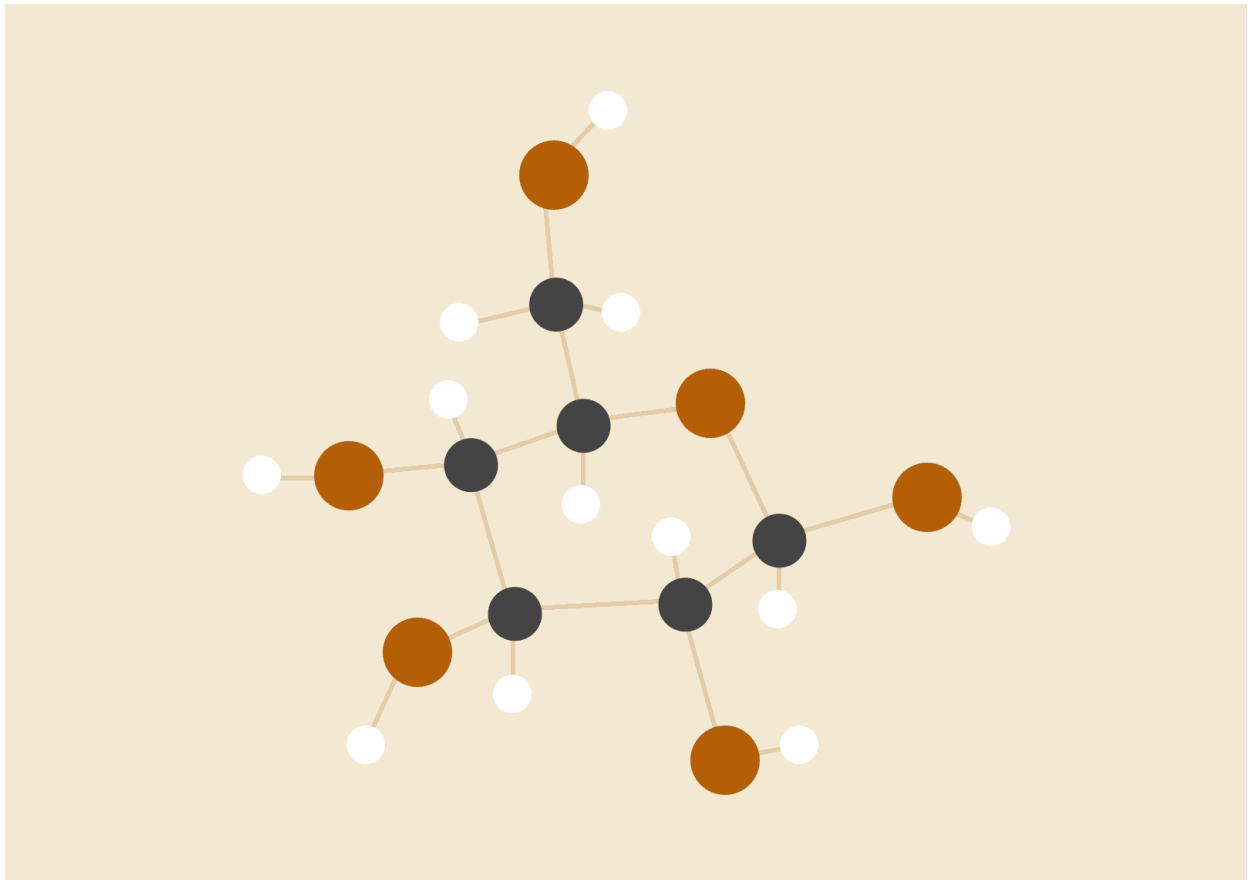


# TRABAJO PRÁCTICO INTEGRADOR

*IA4.4 Procesamiento de Imágenes I*

*Tecnicatura Universitaria en Inteligencia Artificial*



**Brizuela Cipolletti Sofía**

**Fontana Gustavo Julián**

19/11/2023

Universidad Nacional de Rosario  
Facultad de Ciencias Exactas, Ingeniería y Agrimensura

## INTRODUCCIÓN

Se plantean dos problemas para detección en imágenes, en primer lugar se pide segmentar monedas y dados y luego clasificarlos. Por ultimo se pide procesar imágenes de autos, obtener sus patentes y segmentar los valores dentro de ella.

### PROBLEMA 1 – Detección y clasificación de Monedas y Dados

La imagen monedas.jpg, adquirida con un smart phone, consiste de monedas de distinto valor y tamaño, y de dados sobre un fondo de intensidad no uniforme.

- a) Procesar la imagen de manera de segmentar las monedas y los dados de manera automática.
- b) Clasificar los distintos tipos de monedas y realizar un conteo, de manera automática.
- c) Determinar el número que presenta cada dado mediante procesamiento automático.

### PROBLEMA 2 – Detección de patentes

Se tienen imágenes de la vista anterior o posterior de diversos vehículos donde se visualizan las correspondientes patentes.

- a) Implementar un algoritmo de procesamiento de las imágenes que detecte automáticamente las patentes y segmente las mismas. Informar las distintas etapas de procesamiento y mostrar los resultados de cada etapa.
- b) Implementar un algoritmo de procesamiento que segmente los caracteres de la patente detectada en el punto anterior. Informar las distintas etapas de procesamiento y mostrar los resultados de cada etapa.

## PROCEDIMIENTO

### PROBLEMA 1 – Detección y clasificación de Monedas y Dados

En primer lugar se carga la imagen y se pasa a escalas de grises, luego se la aplican diferentes transformaciones para generar una mejor detección de bordes.

- Se aplica un filtro Gaussiano para suavizar la imagen y reducir el ruido;
- Se aplica Canny para detectar bordes continuos y definidos;
- Se realiza procesamiento morfológico aplicando erosión y dilatación;

- Se aplica además luego de binarizar la imagen clausura para rellenar huecos.

Una vez obtenida la imagen procesada completamente, se encuentra lista para la detección y clasificación.

Se van a definir dos funciones para esta tarea:

Por un lado se crea la función ‘contar\_huecos’ que recibe como argumento a la imagen original, la imagen procesada y una posición que servirá para detectar coordenadas. La función detecta los contornos y de acuerdo a la posición en la que se encuentra extrae las coordenadas del contorno y obtiene una ROI de la imagen original. Esta ROI es procesada con diferentes filtros mejorando la detección de bordes. Luego de esto se obtienen los contornos internos, se los cuenta accediendo a la jerarquía de la función de detección y se los almacena en una variable. Finalmente se retorna un ploteo de la ROI con los contornos de los huecos dibujados y el total de huecos encontrados. (además se devuelven otras variables con el fin de presentar los pasos de procesamiento).

Por otro lado se define la función ‘detectar\_objeto’ que toma la imagen ya procesada y la imagen original y retorna los objetos en la imagen original ya clasificados y contados. La función en su implementación detecta las componentes conectadas de la imagen procesada y encuentra los contornos externos, para esto utiliza funciones de opencv. Luego a través de ciclos se va a iterar sobre los contornos, y con umbrales establecidos de manera experimental se van a filtrar los diferentes objetos sobre la imagen y se va a guardar información en variables sobre la cantidad individual y total de cada uno de ellos. Además se van a colocar etiquetas sobre las coordenadas de los contornos detectados indicando el tipo de objeto y en el caso de ser un dado, el número que muestra (para detectar el número de los dados se invoca a la función definida previamente ‘contar\_huecos’). Por último se retorna el ploteo de la imagen con los resultados obtenidos.

## Resultados del procesamiento:

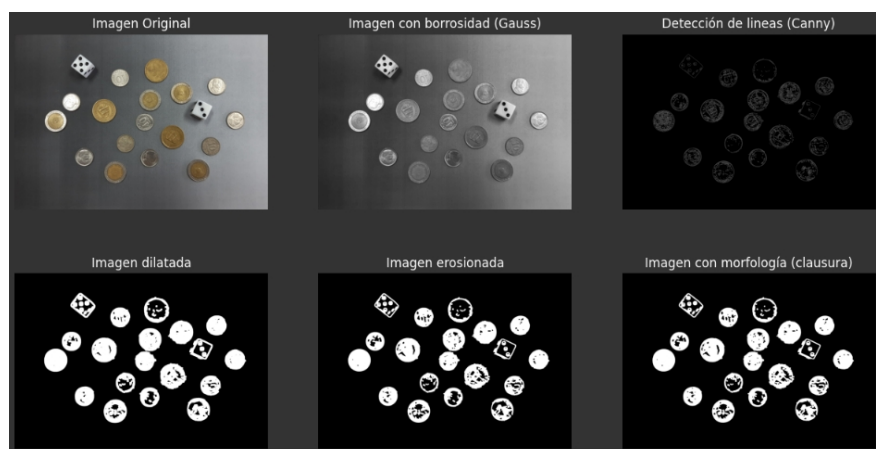


figura 1: Fases del procesamiento de la imagen

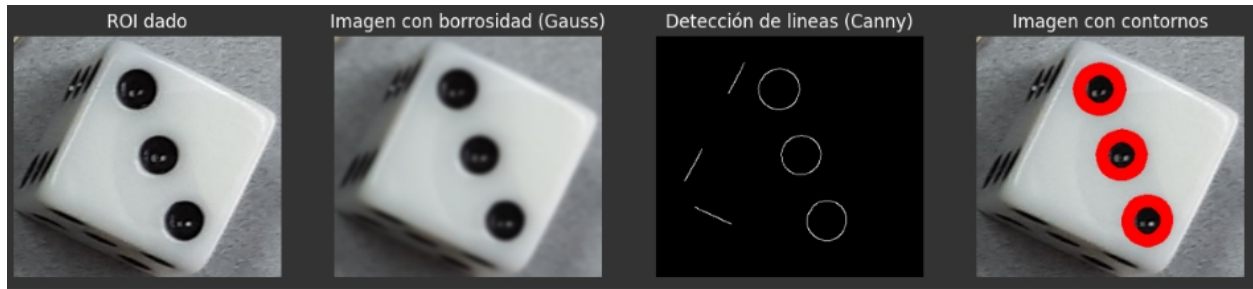


figura 2: Detección de contornos internos

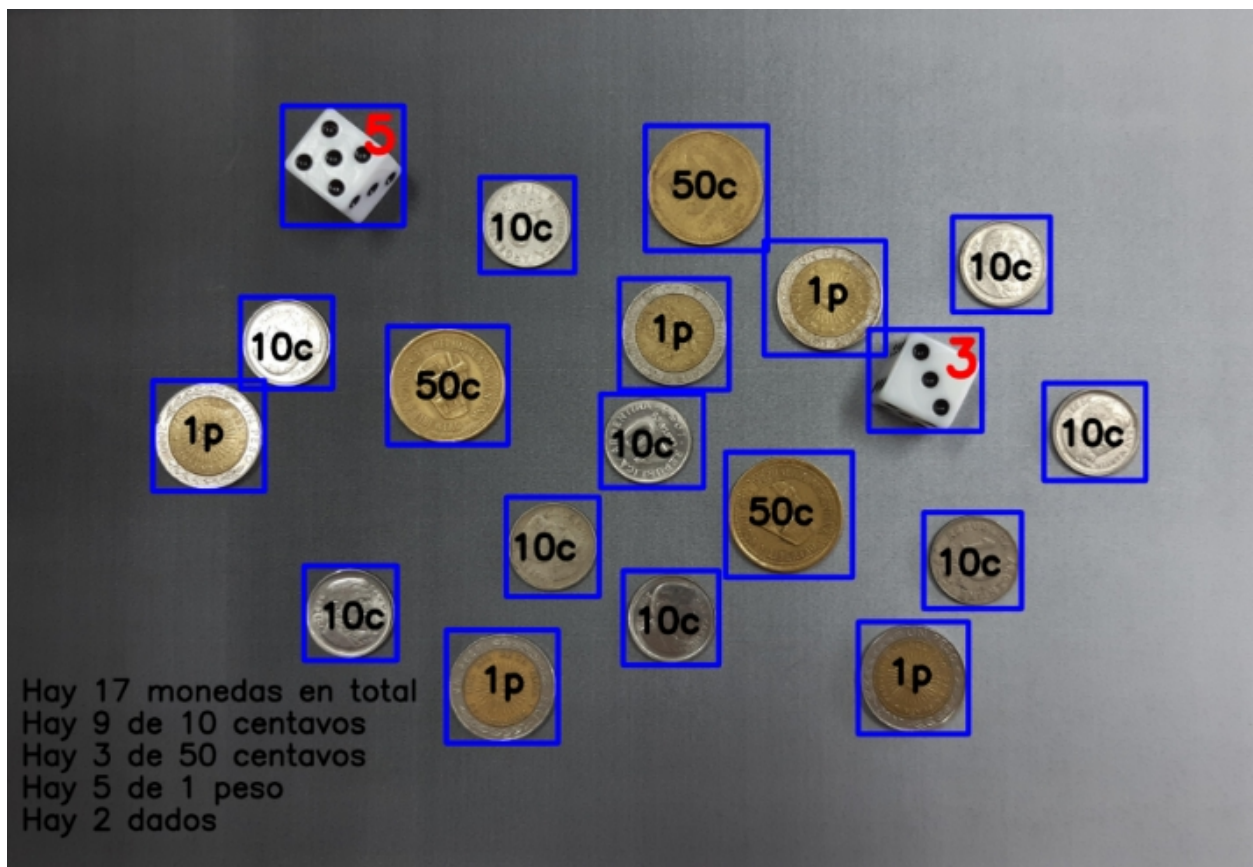


figura 3: Resultado de aplicar el procesamiento y las funciones

## PROBLEMA 2

Para la detección y segmentación de patentes se van implementar dos funciones principales:

Por un lado la función ‘obtener\_patentes’ que va a recibir una imagen y la va a procesar con diferentes métodos para mejorar la detección de las componentes conectadas.

- Se aplica un filtro Sobel para resaltar zonas de la imagen donde se producen cambios importantes en la intensidad de los píxeles;
- Se binariza la imagen con umbrales adecuados;
- Se aplican operaciones morfológicas, dilatación para engrosar los objetos, y apertura para remover objetos pequeños.

Luego se detectan las componentes conectadas y se iteran sobre los stats con umbrales de áreas definidos experimentalmente que sirven como filtros para detectar solo áreas similares a las patentes. En este procedimiento es posible que otros objetos que no sean patentes tengan áreas similares y sean detectados (luces, logos de marcas, etc), es por eso que se coloca un filtro mas que calcula la relación de aspecto y se queda con aquellos con formas similares a una patente. Cuando se filtra correctamente la patente se almacenan sus coordenadas y se extrae una ROI de la imagen original. La función retorna la ROI y otras variables de interés para la visualización de los pasos de procesamiento.

La segunda función definida para la tarea de segmentación es ‘segmentar\_patentes’ que recibe la imagen de una patente y retorna los caracteres internos ya segmentados. La función carga la imagen, la binariza y detecta las componentes conectadas. Luego se itera sobre stats para extraer las coordenadas de la componentes (solo aquellas que cumplen ciertas condiciones establecidas previamente de manera experimental) con las cuales se definen ROIs que se las almacenan en una variable. La función retorna las ROIs y otras variables de interés para la presentación de procesamiento.

Por último se define una función para plotear las imágenes obtenidas como resultado de aplicar las funciones comentadas previamente. La función recibe las imágenes y el alto y ancho de la figura.

## Resultados del procesamiento:

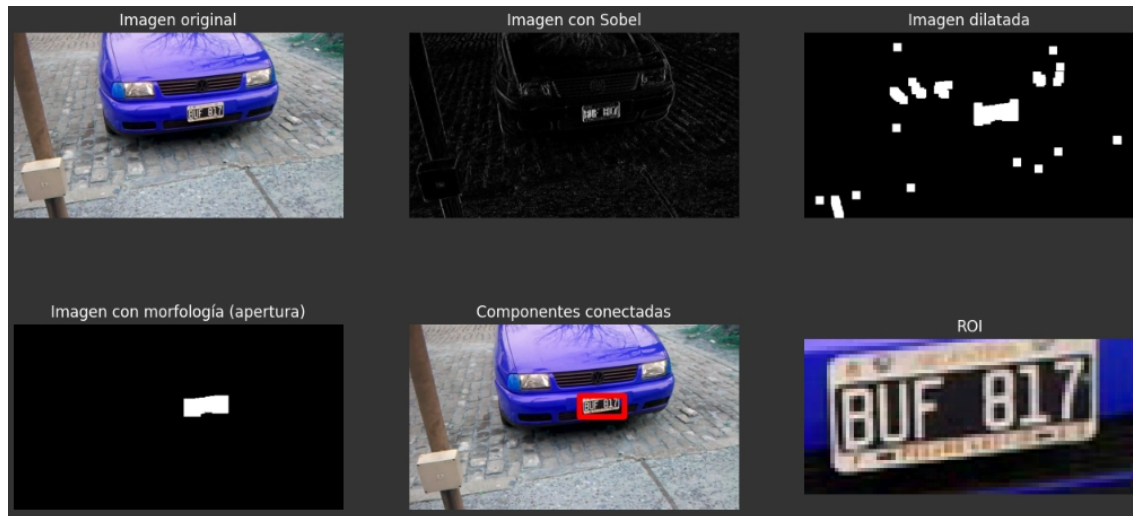


figura 4: Resultado de detectar las patentes



figura 5: Procesamiento de una patente

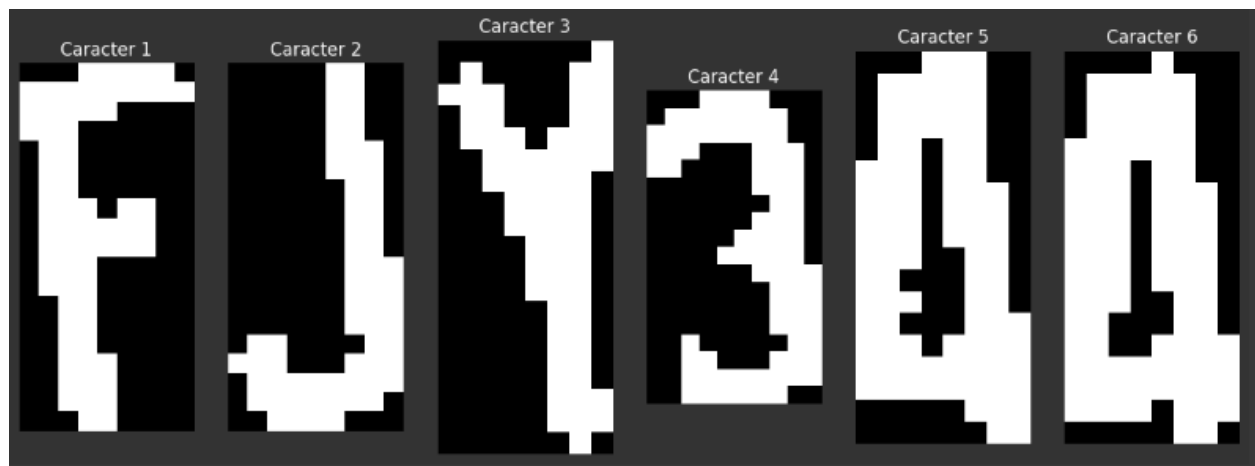


figura 6: Segmentación de una patente



figura 7: Todas las patentes detectadas



figura 8: Bounding box detectados sobre los caracteres de las patentes