

Informe TP°2 PDI

Tecnicatura Universitaria en Inteligencia Artificial

Integrantes:

- Mateo Rovere
- Valentín Dalmau

Profesores:

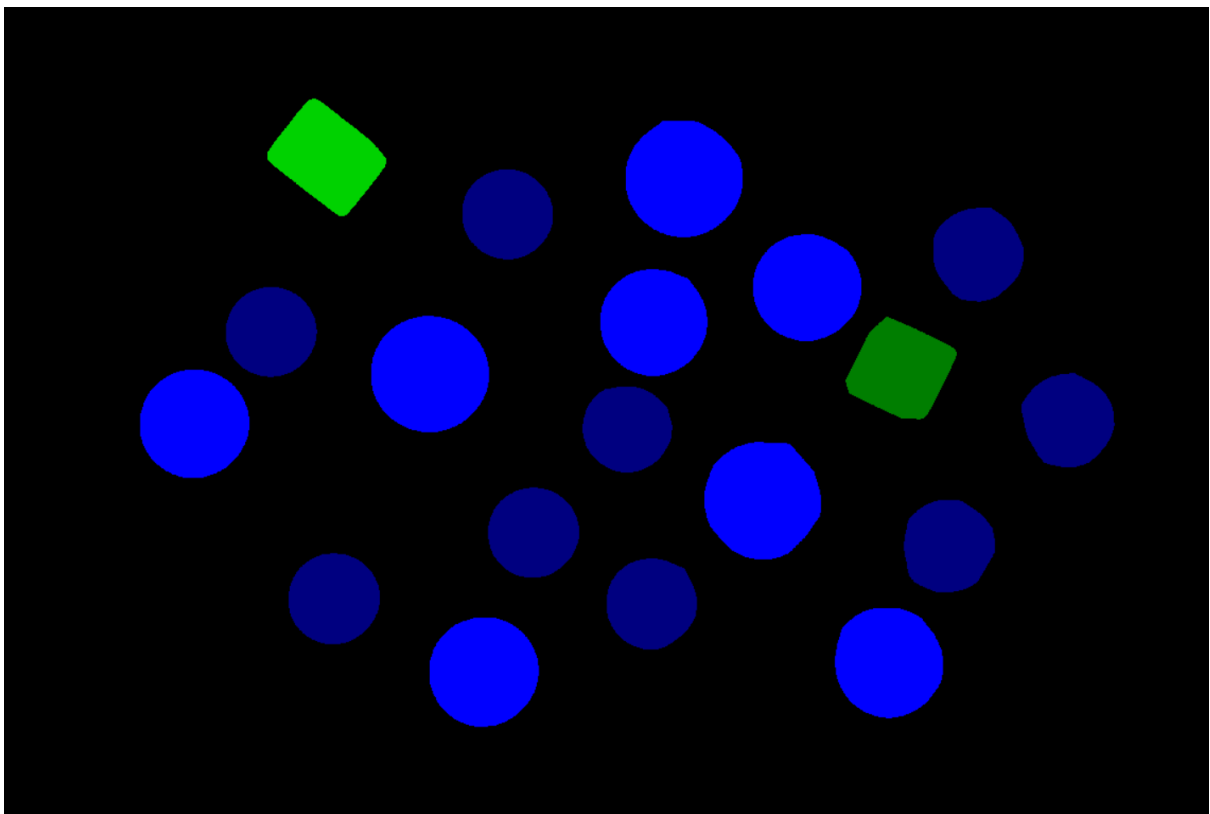
- Gonzalo Sad
- Facundo Reyes
- Julián Álvarez

Ejercicio 1:

Para resolver este ejercicio, lo que se nos ocurrió fue:

- Pasar a gris la imagen
- Hacerle un blur Gausiano
- Luego aplicamos Canny
- Hicimos una dilatación y una erosión
- Encontramos los contornos, y a los que tenían un área suficientemente grande, les aplicamos convexhull para convertir los contornos cóncavos a convexos, y a esos contornos los mandamos a una imagen nueva.
- Luego, a partir de esa imagen nueva, sacamos las componentes conectadas, y de ahí separamos los círculos con un umbral de $FP = \text{area} / \text{perimetro}^{**2}$, y separamos por área las monedas más pequeñas, para los dados decidimos colorear con diferente intensidad dependiendo de cuántos puntos había en la cara del dado
- Luego retorna la máscara con las monedas segmentados por tamaño y los dados con diferente intensidad de verde dependiendo de su valor

Y quedaria algo asi:



Ejercicio 2:

Para realizar este ejercicio seguimos los siguientes pasos:

- Asumimos que el color no nos daba ninguna información útil, por lo que pasamos a escala de grises.

- Luego realizamos un umbralado local, ya que el umbralado global no daba buenos resultados en la mayoría de patentes con unos parámetros fijos. Con el umbralado local conseguimos que en 10 de 12 imágenes no desaparezca parte de la patente al realizar las componentes conectadas más adelante. Los parámetros que modificamos eran el tamaño de la ventana en la que se aplica el umbralado y una constante que se le agrega al brillo medio para calcular el valor de cada píxel (0 o 255).

- Aplicamos la función `ConnectedComponentsWithStats` para poder filtrar el ruido por área y relación de aspecto. Como cada imagen tenía un área y una relación de aspecto diferente para cada patente estos parámetros no podían ser lo suficientemente estrictos como para eliminar todo el ruido

- Creamos y aplicamos una función para eliminar ruido utilizando la cercanía a otras componentes conectadas, ya para este punto algunas imágenes quedaban completamente limpias

- Intentamos filtrar ruido aplicando análisis morfológico (operación de apertura más clausura), pero el filtrado era más invasivo sobre la patente que sobre el ruido de fondo, por lo que no fue de utilidad. También intentamos erosionando pero no pudimos conseguir nada útil.

- Para deshacernos del ruido final en algunas imágenes aprovechamos que las patentes estaban cerca del centro y eliminamos parte de los bordes.

- Por último para segmentar por caracteres hicimos un `ConnectedComponentsWithStats` y mostramos cada componente, ya que solo quedaban los caracteres de la patente en la imagen.

- Creamos un bucle que pueda recorrer todas las imágenes aprovechando sus nombres de archivo secuenciales.



Imagen escala de grises



Imagen con umbralado local



Imagen filtrada por área y relación de aspecto



Imagen filtrada por cercanía a una componente conectada



1



2

Primeros dos caracteres de la patente filtrados