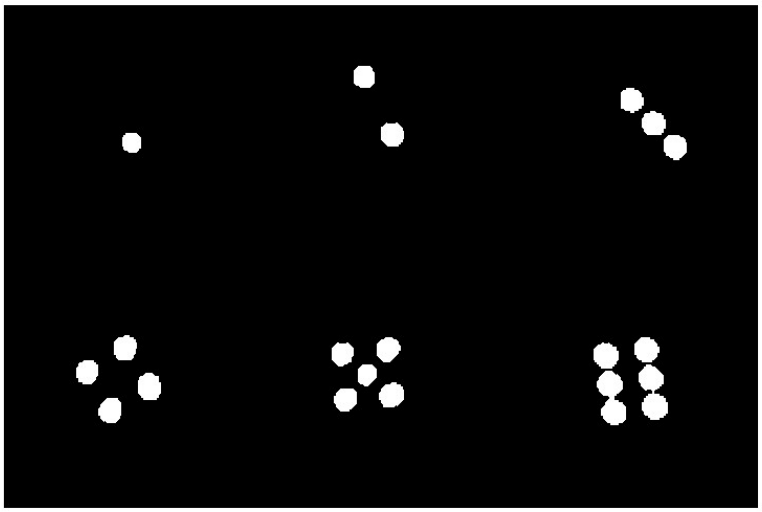
Universidad Francisco de Vitoria

Grado en Ingeniería Matemática

Procesamiento de imágenes de dados



Noviembre 2024

Índice

1. Selección de Dataset de Imágenes Relevantes **1**

2. Exploración visual **1**

3. Algoritmo de Clasificación **2**

3.1. Smoothing 2
3.2. Closing 2
3.3. Adjusting 3
3.4. Black Hat 3
3.5. Thresholding 4
3.6. Identificación de puntos 4
3.7. Filtro sobel 6

4. Conclusión **8**

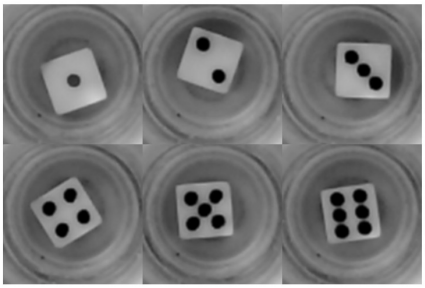
1. Selección de Dataset de Imágenes Relevantes

Para esta práctica se ha elegido un dataset de dados vistos desde arriba. Se trata de imágenes que ya se encuentran en escala de grises y en ellas los dados se encuentran sobre fondos algo complejos por contener sombras y elementos circulares. El dataset cuenta con 2400 imágenes de las cuales 400 pertenecen a cada tipo de dado (1-6). Existen dados blancos con puntos negros y dados negros con puntos blancos.

Dataset: https://github.com/tomitomi3/DiceRecognitionDatasetForML.git

2. Exploración visual

Inicialmente vemos el formato que tiene un dado, y nos gustaría saber que valor del 1 al 6 tiene de manera automática. Esto podría tener aplicación para un casino o si quisiésemos crear un bot que con computer vision tuviese una herramienta para videojuegos que incluyan dados.

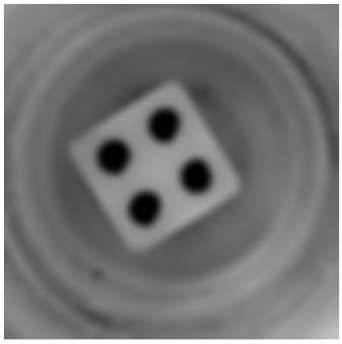


Por tanto, característica a obtener: **Valor del dado**

3. Algoritmo de Clasificación

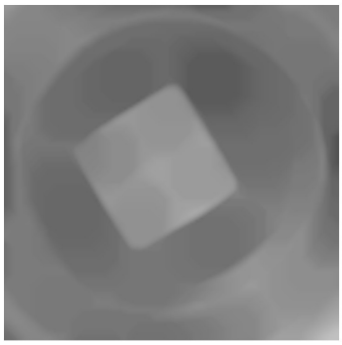
3.1. Smoothing

Se comienza con la preprocesamiento de la imagen. Se realiza un suavizado de la imagen con un filtro de media que tiene una ventana 4x4. Esto se hace para eliminar ruido de la imagen y mejorar la detección de bordes.



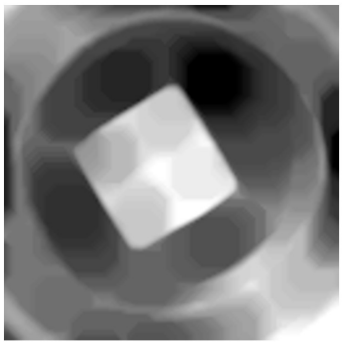
3.2. Closing

Se realiza un cierre de la imagen con un elemento estructurante con forma "disk"de radio 10. Esto se hace para eliminar los bordes de la imagen y mejorar la detección de los contornos.



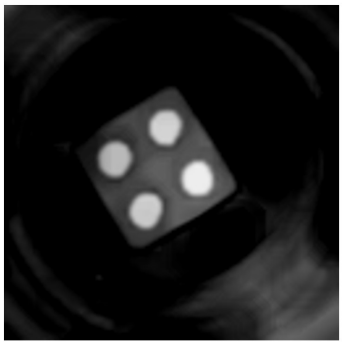
3.3. Adjusting

Se aumenta el contraste de la imagen con un ajuste de contraste. Esto facilita técnicas como la creación de máscaras, aunque en nuestro caso utilizaremos la operación morfológica de blackhat.



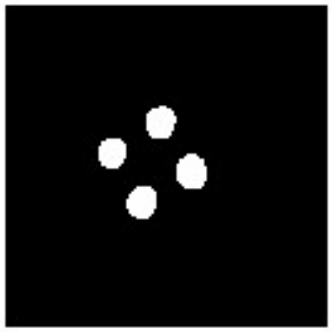
3.4. Black Hat

Se realiza una operación morfológica blackhat a partir del close obtenido anteriormente, esto nos permite eliminar el fondo y dejar solo los puntos de los dados.



3.5. Thresholding

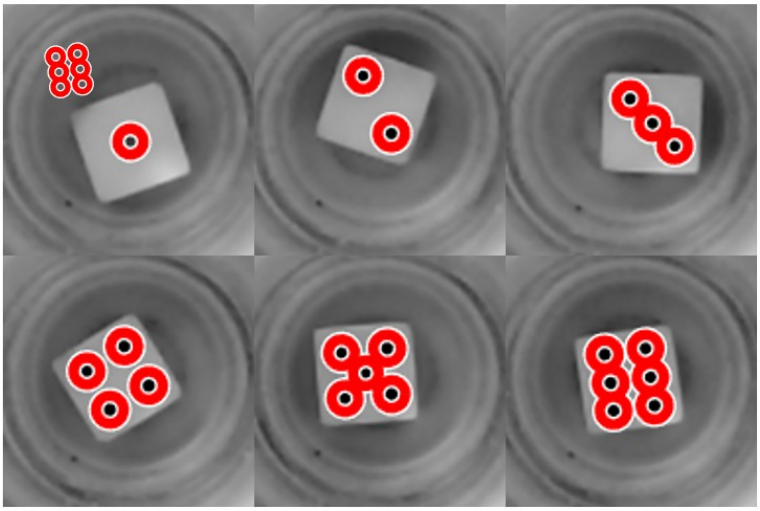
Utilizando un umbral de 128 obtenemos una imagen binaria de los puntos de los dados.



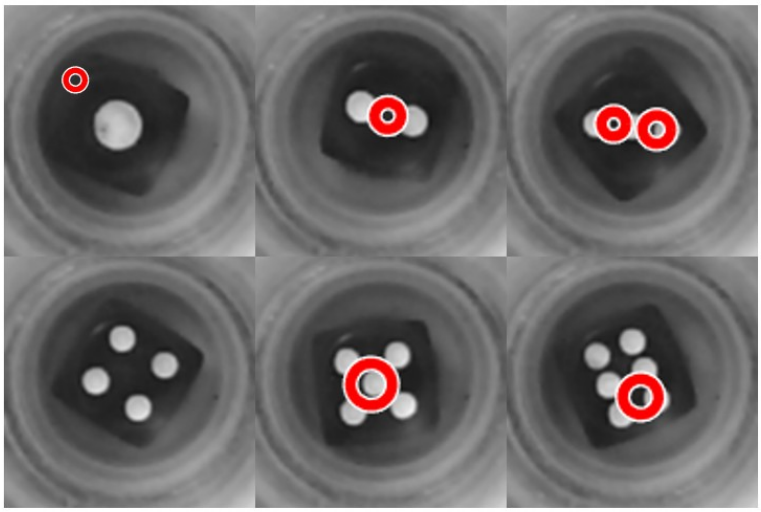
3.6. Identificación de puntos

Utilizamos la función imfindcircles de MATLAB con parámetros adecuados para identificar los puntos de los dados. Para este tipo de imágenes se buscan círculos con radios entre 6 y 12. También se reduce la sensibilidad con el objetivo de identificar puntos muy juntos que se dan en los casos de dados de valores 5 y 6. Y adicionalmente si no se detectan círculos se elige el valor 1 para el dado y si el número de círculos detectados es superior a 6 se elige el valor 6.

Obtenemos resultados bastante buenos para dados blancos.



No obstante fallamos en los dados negros.



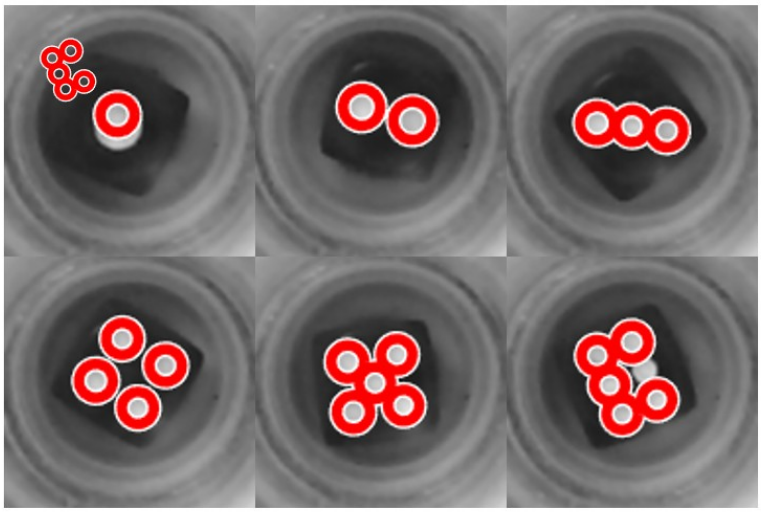
Aunque se ha conseguido obtener resultados muy buenos para los dados blancos, no es posible utilizar la misma técnica para los dados negros.

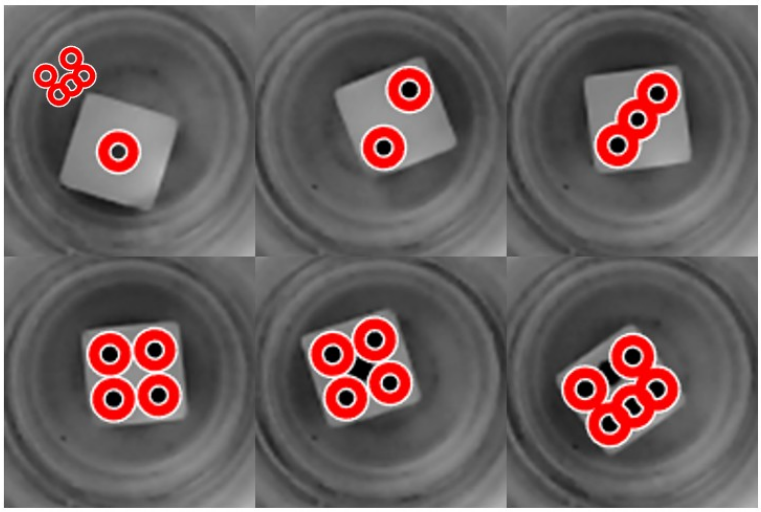
Para el total del dataset se ha propuesto un algoritmo sencillo pero funcional. La probabilidad de acertar el dado lanzando un dado es de un 16.6 % para mejorar el caso de este generador de números aleatorios, el siguiente algoritmo obtiene un 72.79 % de precisión.

3.7. Filtro sobel

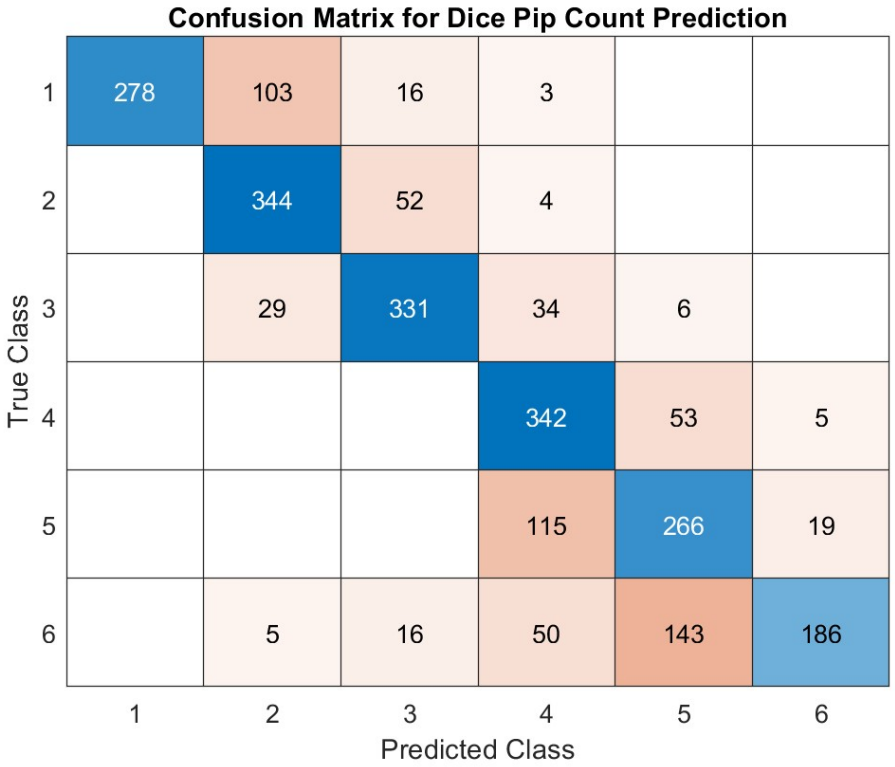
Se trata de un filtro para la identificación de contornos, y aunque menos sofisticado que el filtro Canny obtiene resultados simples que funcionan bien para nuestro caso.

Al combinarlo con la función **imfindcircles** obtenemos resultados bastante buenos tanto para dados blancos como para dados negros.





Hemos obtenido la matriz de confusión que muestra el error cometido en el algoritmo de clasificación.



Observándose una precisión del 72.79 % mejorando considerablemente el caso de utilizar un generador de números aleatorios (1/6).

4. Conclusión

El modelo de clasificación para este dataset que se utiliza en la fuente se basa en redes de neuronas convolucionales, sin embargo y dependiendo del problema es posible aplicar técnicas más sencillas y rápidas (aunque requieran de una comprensión profunda para obtener una buena precisión) como operaciones morfológicas y análisis de contornos mediante filtros.

Como conclusión, se ha logrado desarrollar algoritmos capaces de detectar el valor de un dado y se deja a futuro la posibilidad de mejorarlos analizando las formas que definen los dados estudiando los contornos y áreas diferenciadas caracterizadas por la forma cuadrada de los dados.