

Resultados y conclusión.

Para entrenar al clasificador, utilizamos un dataset en el que encontramos letras y números.

Las imágenes del proceso serían las siguientes, y consisten en detectar el carácter y señalarlo, hacer un crop para quedarnos solo con el propio carácter, y finalmente hacer un resize a las dimensiones requeridas.



Para comprobar el buen funcionamiento, tenemos otro dataset de palabras completas.

Las imágenes obtenidas serían las siguientes:

- Comenzamos con la imagen original:



- Elegimos y recortamos el primer carácter:



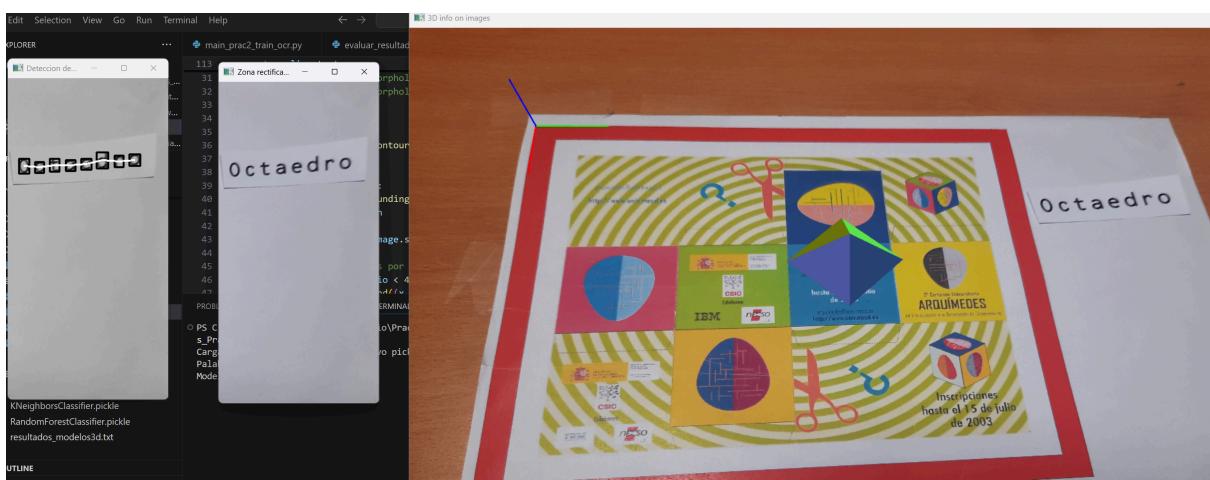
- Hacemos resize del mismo para que sea igual que los caracteres con los que hemos entrenado al clasificador:

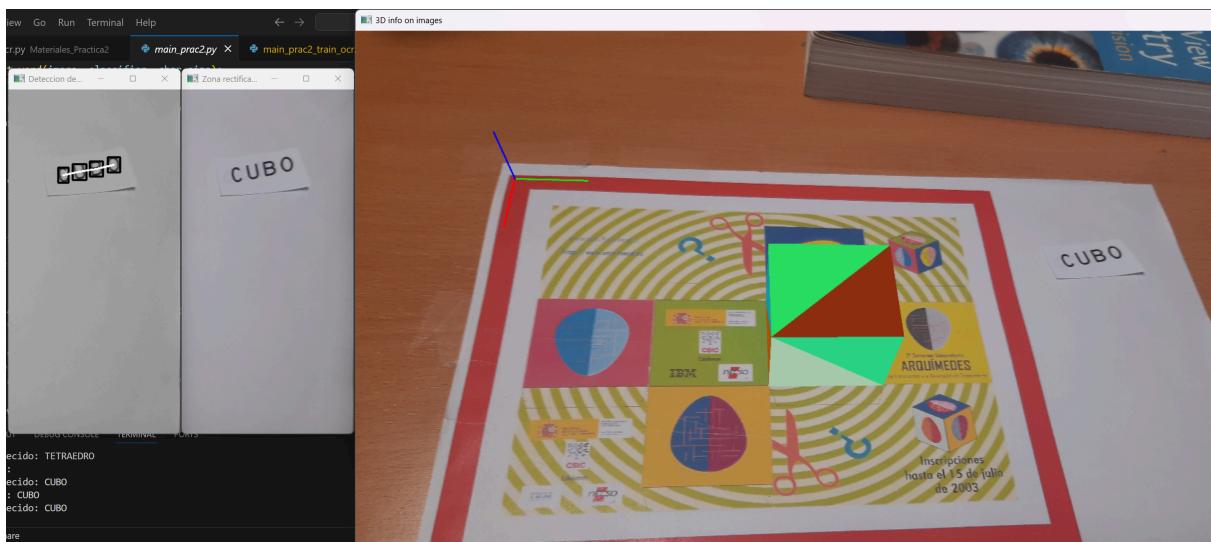
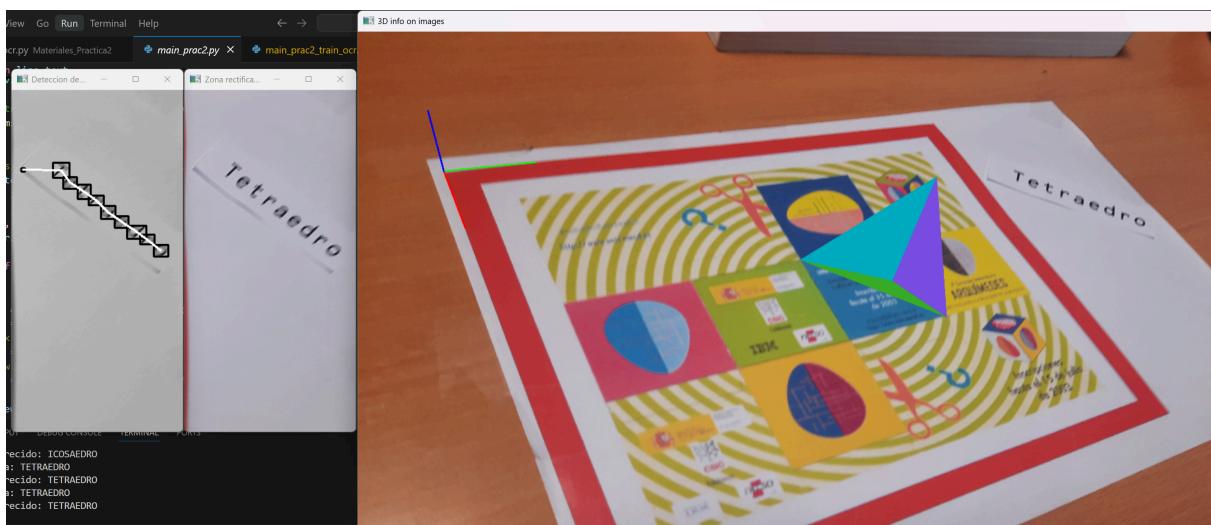
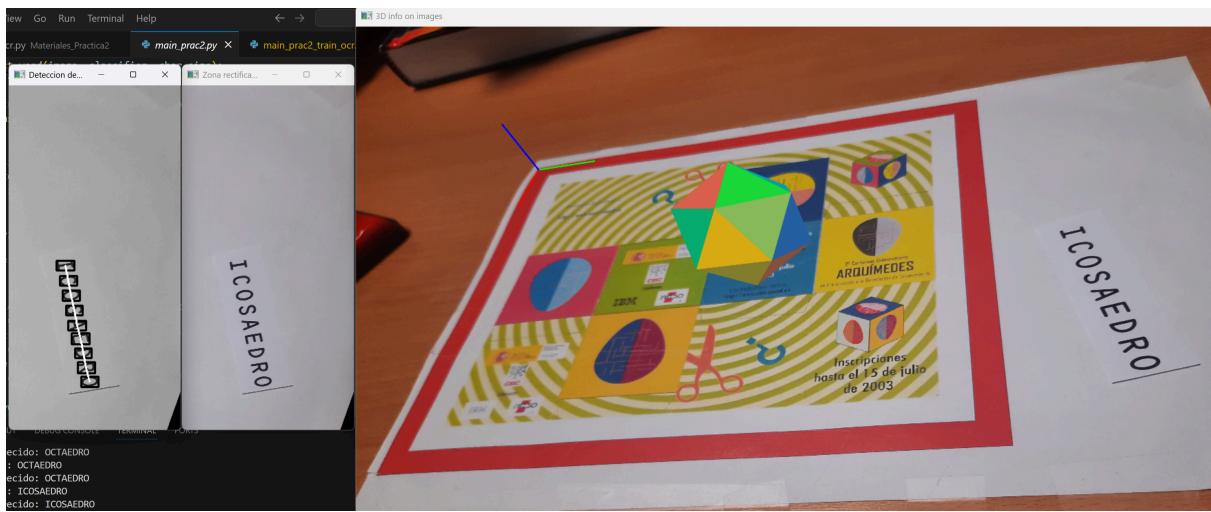


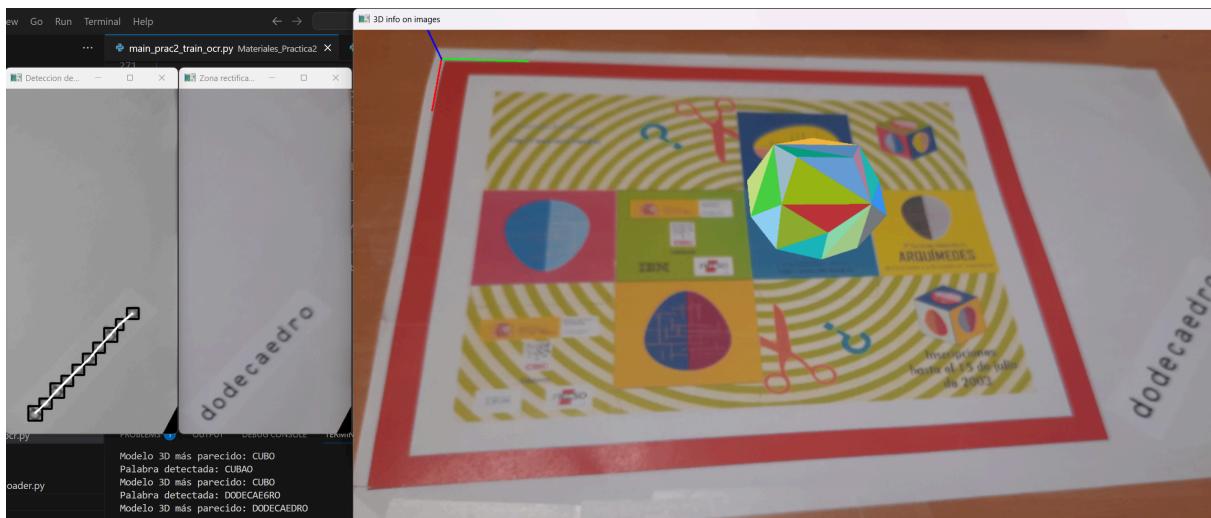
Finalmente, utilizamos nuestro modelo para la finalidad deseada, que en este caso es detectar el polígono que vamos a colocar en 3d en la imagen.

En las siguientes imágenes, veremos el comportamiento de nuestro clasificador para cada polígono en la práctica.

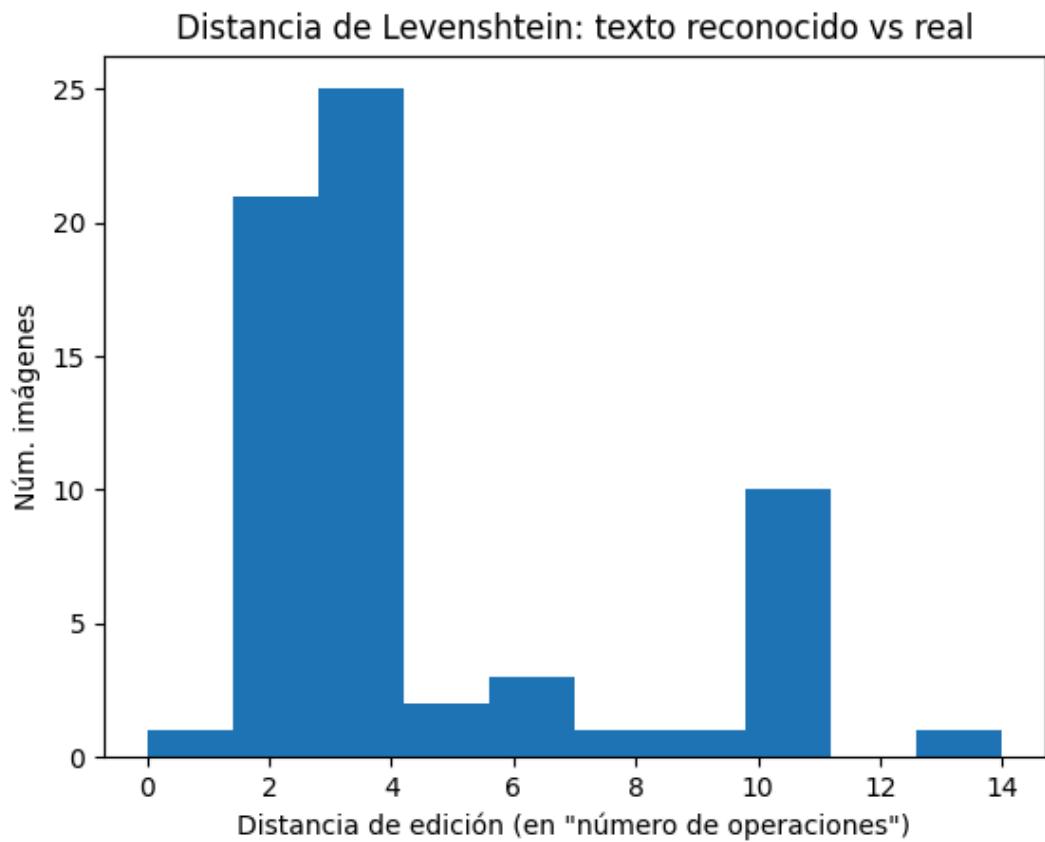
Primero se rectifica la imagen recortada a nuestra zona de interés, acto seguido, utilizamos RANSAC para conseguir el orden de caracteres y la colocación de los mismos para su predicción.







Podemos ver en la siguiente gráfica los resultados obtenidos.



Como vemos, los resultados a pesar de no ser perfectos, gracias a minimizar los errores de la predicción con la distancia Levenshtein, conseguimos tener bien prácticamente todos, algo que consideramos un logro, dada las

dificultades que hemos tenido para conseguir que funcione este último apartado. Estos resultados reflejan la calidad de predicción previa a la elección del modelo 3D más cercano con la distancia Levenshtein, sin embargo debemos haber cometido algún error a la hora de hacer el gt.txt, ya que simplemente tomando las 3 primeras predicciones, nos encontramos con que 2 de estas son exactamente iguales a la que teníamos que predecir, sin embargo, en la gráfica vemos como sólo aparece 1 palabra con distancia 0.

```
Cargado clasificador desde el archivo pickle.  
Palabra detectada: OCTAEDRO  
Modelo 3D más parecido: OCTAEDRO  
Palabra detectada: OCTAEDRO  
Modelo 3D más parecido: OCTAEDRO  
Palabra detectada: OCTAEDRO  
Modelo 3D más parecido: OCTAEDRO
```