

Tarea X - Procesamiento y Análisis de Imágenes

Nombres y Apellidos
Universidad de Santiago de Chile

Motivación—El reconocimiento y clasificación de patrones ha tomado suma importancia en el último tiempo por todas las aplicaciones que se pueden hacer en distintas áreas. Por ejemplo, la detección de una placa patente en un estacionamiento o la detección facial en un aeropuerto. Esta tarea consiste en el primer acercamiento a esta área, creando un clasificador de letras C y S .

I. SOLUCIÓN PROPUESTA

La solución propuesta utiliza un árbol de decisión binario como modelo predictivo, en donde el primer nodo corresponde a la simetría de la figura. El valor separador de esta característica corresponde a 0,4125, en donde si esta es mayor o igual a ese valor, es probable que sea una C .

El siguiente nivel del árbol utiliza el centro de masa en x de la figura. Para que esta sea una C se debe cumplir que el com_x debe ser menor al com_x del primer valor de S después del último valor de C (0,5061). Para que sea una S , se debe cumplir que el com_x debe ser mayor al com_x del último valor de C antes del primer valor de S (0,4644).

Si lo anterior no se cumple, se recurre a la última característica, el centro de masa en y , donde el valor de separación corresponde a la intercepción de las curvas de densidad de la figura 5 (0,502).

El orden de las características y los valores del árbol fueron obtenidos a partir de los experimentos realizados, explicados en la sección II. El árbol de decisión se muestra en la figura 1.

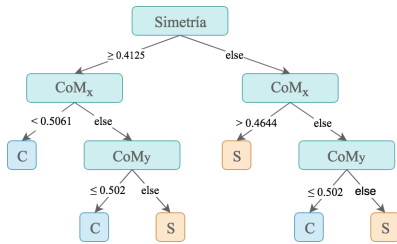


Fig. 1. Árbol de decisión

El clasificador se encuentra en el archivo `classifier.py`, en donde hay que utilizar la función `reconocedorSC(X)` para realizar la clasificación. Notar que el primer paso de este método es segmentar la figura mediante el método `crop_img(X)`.

II. EXPERIMENTOS REALIZADOS

Para realizar los experimentos, se creó un *dataset* de 80 figuras. Este tiene imágenes de distinto tamaño, tipografía y ubicación de la figura dentro de la imagen, con igual proporción de letras C y S .

Lo primero que se hizo fue segmentar la imagen, para eliminar todo el fondo innecesario de la figura. Luego, el

primer experimento se realizó con el área de la figura, en donde la hipótesis era que la S tendría un área mayor. Los resultados se muestran en la figura 2 y vemos que a pesar que se logran separar parcialmente según la hipótesis, hay *outliers* que se separan de la media por lo que se decidió no utilizar esta característica.

El segundo experimento se realizó con la simetría de la figura respecto al eje x , en donde la hipótesis era que la C tenía un índice mayor. Los resultados se muestran en la figura 3 y vemos que logramos una separación completa del *dataset*. Dado lo anterior, se decide que el punto de separación corresponde a $x = 0,4125$.

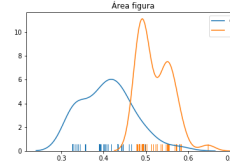


Fig. 2. Área figura

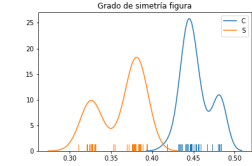


Fig. 3. Simetría figura

El último experimento se realizó con el centro de masa de la figura en x y y . La hipótesis era que en x , el centro de masa en x de la letra C estaba más a la izquierda que la letra S . Los resultados se muestran en las figuras 4, 5 y vemos que a la separación en x es levemente mejor que en y , cumpliendo con la hipótesis planteada. Se decidió utilizar el centro de masa en x como segundo nodo en el árbol de decisión y el centro de masa en y como el final.

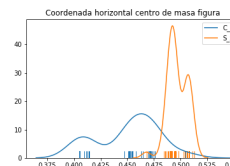


Fig. 4. com_x figura

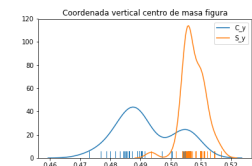


Fig. 5. com_y figura

Para probar el clasificador, se creó un set de testeo de 34 imágenes, en donde la exactitud fue de un 94%. Los experimentos se encuentran en el archivo `train.py`.

III. CONCLUSIONES

A modo de conclusión, podemos afirmar que la simetría es una buena característica para clasificar este problema específico. Sin embargo, si consideramos todo el abecedario, probablemente este no sea el *approach* correcto y al menos habría que considerar un modelo predictivo basado en los descriptores de Fourier. El próximo desafío sería ampliar las letras a todo el abecedario probando distintos modelos y posterior a eso, a distintas tipografías.