

Tarea 1 – IIC3724

Benjamín Farías V.
Pontificia Universidad Católica de Chile

Motivación— La clasificación de caracteres en imágenes es un desafío crucial para ciertos tipos de actividades hoy en día. Dichas actividades van desde el reconocimiento de tipografías poco claras en textos y detección de patentes de vehículo, hasta la indexación eficiente en ciertas bases de datos. La tarea entrega una primera experiencia en esta área de investigación, mediante la construcción de un clasificador de letras A, S, D, F y G.

I. SOLUCIÓN PROPUESTA

La solución propuesta utiliza un clasificador k-NN (k vecinos más cercanos) como modelo predictivo, usando la distancia euclidiana entre vectores de caracteres como la métrica de similitud. Las componentes de estos vectores corresponden a distintas características relevantes que fueron extraídas de las imágenes de entrenamiento. Estas son:

- **Roundness:** La redondez del contorno del objeto.
- **Hu-Moments:** Momentos invariantes respecto a la traslación, escala y rotación del objeto.

La relevancia de estas características fue determinada mediante visualización de histogramas y experimentación (sección experimentos). A partir de esto, se encontró que los *Momentos de Hu* 2, 5 y 6 no aportaban mucho a la separación de las clases, por lo que fueron omitidos en el clasificador final. Además, debido a los valores muy pequeños que entregaban los *Momentos de Hu* de grado alto, fue necesario normalizar todos los datos utilizando el método *z-score*, de forma que tuvieran el mismo peso en la métrica de similitud. Finalmente, el valor de *k* para el clasificador fue determinado mediante ensayo y error, llegando a que el mejor valor para la imagen de testing corresponde a *k* = 1.

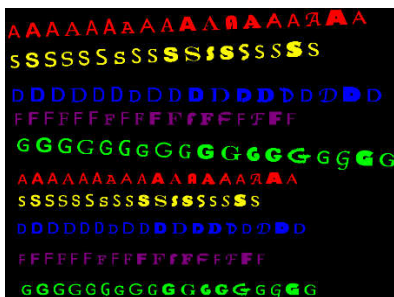


Fig. 1. Resultados de la clasificación

(A: Rojo, S: Amarillo, D: Azul, F: Morado, G: Verde)

Los módulos `char_classifier.py` y `utils.py` contienen todas las funciones necesarias para el funcionamiento del programa, el que corre al ejecutar el módulo `main.py`.

II. EXPERIMENTOS REALIZADOS

Para realizar los experimentos, se utilizaron las 2 imágenes entregadas como dataset de entrenamiento, cada una con 180 caracteres (36 por clase). Para validar las características se observó la separación de las clases mediante histogramas (uno por cada característica). El histograma asociado a la *Roundness* (Fig. 2), indica que las letras S, G y F suelen tener menores valores, mientras que las A y D tienen una alta Redondez. De esto se desprende que es una característica aceptable para el clasificador. Al analizar los histogramas para los *Momentos de Hu*, se obtienen formas similares para ciertos grados de los momentos, mostrados a continuación:

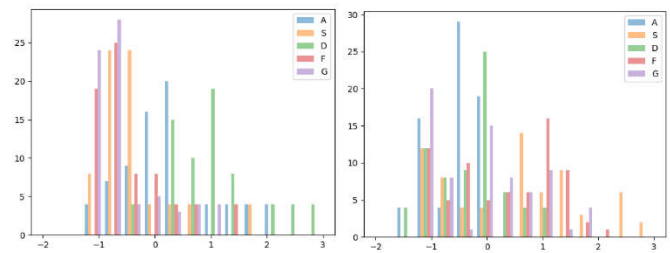


Fig. 2-3. Roundness y Hu (1,2)

Los momentos menores a 2 logran separar las letras A y D de las S y F. Los momentos mayores a 2 presentan una gran concentración de clases alrededor de la media, por lo que funcionan bien como discriminantes para las letras A y F, que están más alejadas de dicha concentración. Luego, a través de ensayo y error, se encontró que basta con los momentos 1,3,4 y 7 (más la redondez) para lograr el 100% de **exactitud** en la clasificación de la imagen de testing (ver matriz de confusión, Fig. 5).

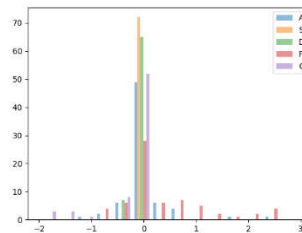


Fig. 4. Hu (3,4,5,6,7)

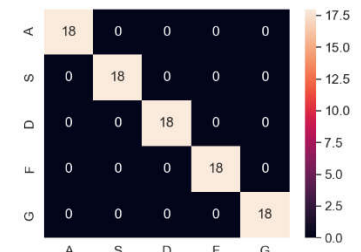


Fig. 5. Matriz de Confusión

III. CONCLUSIONES

Como conclusión, podemos afirmar que, para estas letras, los momentos invariantes son suficientes para diferenciarlas. Además, en este caso el *k* óptimo fue 1, por lo que probablemente habían *fonts* de letras que generaban ruido en la separación, empeorando el rendimiento a mayor valor de *k*. Para otros tipos de letra será necesario probar con otros métodos, tales como los *descriptores de Fourier*, debido a un aumento en la complejidad de la forma de los caracteres.