

Informe Tarea 2

El reconocimiento de imágenes está cada vez más en alza en el mundo actual, y esta tarea es fundamental para adentrarse en dicha área. En ésta, se trabaja la elección de una combinación de características que en conjunto logren separar de la mejor forma posible las distintas clases, intentando a su vez encontrar la mejor compatibilidad posible entre el tiempo de simulación y porcentaje de *accuracy*.

La solución propuesta se basa en:

1. Crear la estructura *fotos_pares* y *fotos_impares*, las cuales permiten crear campos que contienen la dirección del *path* de la foto, el ID, y el número de foto de la persona xxx.
2. Extraer el *LBP* de las 50 clases impares en *fotos_impares* e ir añadiendo las características a $X_{train_{impar}}$ en caso de que se esté en presencia de un número de foto entre 2 y 7 (entrenamiento) para cierta clase. En caso contrario, si se trata de la primera foto 01, se añade las características a $X_{test_{impar}}$. Al mismo tiempo se van formando $Y_{train_{impar}}$ y $Y_{test_{impar}}$ para almacenar los *labels* correspondientes.
3. Normalización de $X_{train_{impar}}$, la cual es usada a su vez para normalizar $X_{test_{impar}}$.
4. Transformación matemática para dejar $Y_{train_{impar}}$ de forma consecutiva.
5. Extracción de 100 características mediante *SFS*, para así aplicar el algoritmo *KKN* con $k=1$ de modo de obtener el mayor porcentaje de *accuracy* posible de la foto 01.
6. Caso totalmente análogo para el caso par, solo que no se aplica *SFS* en este caso, sino que se hace la clasificación utilizando las mismas características que fueron extraídas en la etapa impar por el algoritmo *SFS*.

Dado que el número de características obtenidas del *LBP* considerando una partición de 4x4 era un total de 944 características, el algoritmo *SFS* con el criterio de separabilidad de *Fisher* fue elegido dada su optimalidad en cuanto a cómputo y resultados efectivos. El usar otros algoritmos como *SBS*, Búsqueda Exhaustiva, u otros hubiese generado un cómputo mucho más pesado. Cabe mencionar que el código propuesto es invariante al orden en el que se recorran las imágenes dentro de la carpeta donde se encuentren almacenadas.

El cálculo que involucraba el mayor tiempo de simulación estuvo dado por la extracción de las 100 características por parte del algoritmo *SFS*, demorando dicha operación un tiempo de 243 segundos, y entregando a su vez un porcentaje de *accuracy* de 100% tanto para la parte 1, como para la parte 2 de la tarea. Por otro lado, al usar una partición de 2x2 el tiempo de simulación del algoritmo *SFS* bajó considerablemente a 46 segundos, no obstante, el *accuracy* entregado fue de 96% para la parte I y de 92% para la parte II. Cabe mencionar que dichos valores obtenidos consideran una normalización previa, lo cual ayudó a generar un incremento inmediato en el porcentaje de *accuracy*, subiendo desde 98% a 100% en la parte I, y de 94% a 100% en la parte II utilizando una partición de 4x4.

A modo de conclusión, el método utilizado para esta tarea tuvo una gran efectividad (100%) mediante el método *SFS* y una partición de 4x4, con un tiempo de simulación cercano a los 250 segundos. Si bien el algoritmo *SFS* no hace un barrido sobre todas las combinaciones posibles, las características seleccionadas separan de muy buena forma las 50 clases presentes en esta tarea, ganando a la vez tiempo y eficiencia en comparación a otros métodos que se podrían haber usado.