

Tarea 2 – Raimundo Herrera

Para esta tarea se pretende experimentar con la selección de características sobre un *dataset* de caras de gente. La intención es usar SFS para obtener las features extraídas de las imágenes de entrenamiento que mejor discriminan entre las caras, para luego ser capaz de armar robustamente un clasificador para ellas.

La extracción de características se realiza con la implementación de Local Binary Patterns (LBP) de la librería *pybalu*. Para ello, se entregan dos conjuntos de experimentos en dos *notebooks* de *jupyter*¹. El primero es el *default.ipynb*, utilizando LBP con el método por defecto, es decir, variante a las escalas de gris. El segundo es el *nri_uniform.ipynb* que utiliza LBP pero con el *mapping* “*nri_uniform*”, que es indiferente a los cambios de brillos en la escala de gris. La decisión de realizar el segundo experimento viene dada por dos razones, en primer lugar que los métodos invariantes a alguna dimensión de la imagen tienen menos características y por ende son más rápidos de computar, y en segundo lugar, y más importante, debido a la observación de que el *dataset* contenía imágenes con bastante variación en la luminosidad. No hay problema con ser variante a las rotaciones dado que todas las imágenes tienen orientación vertical y esto no cambia, por lo que no es útil usar métodos invariantes a la rotación.

El LBP en ambos conjuntos de experimentos, y por ende en ambos archivos, se realizó con distintas divisiones horizontales y verticales, dejando finalmente las que obtuvieron mejores resultados. Se obtuvo empíricamente que una división horizontal de 4 partes y una vertical de 2 era la mejor, pero se probó con distintas combinaciones variando entre 1 y 7 divisiones para cada sentido.

Cabe señalar que anterior a la extracción de características se realizó una limpieza de las mismas utilizando el método *clean* explicado en clases, e implementado en *pybalu*. Esto con el objetivo de eliminar características altamente correlacionadas. También se realizó normalización, ya que con ella se reportó mejores resultados, y como se comentó en clases, no existe una regla universal para saber si utilizarla o no. Esta normalización es la de media y desviación estándar, implementada en la librería. Luego de la limpieza y la normalización se realizó el SFS con distintas cantidades para *N*, el número de *features*. Se llegó a la conclusión empírica de que con no más de 30 características se obtenían resultados de gran nivel por lo que la restricción de máximo 100 fue cumplida a cabalidad.

Una observación relevante es que en ambos conjuntos de experimentos utilizar las *features* seleccionadas directamente después de SFS no otorgaba los mejores resultados, pero al computar subconjuntos o combinaciones de dichas características de largos menores, se obtenían los mejores. Eso se comprobó con más de 200 experimentos realizados, dejando plasmado en los archivos solamente los mejores. En otras palabras, si se realizaba por ejemplo un SFS de 30 *features*, sucedía que con 28 de esas 30 se obtenía mejor rendimiento en entrenamiento tras realizar el KNN de 1 vecino.

También es muy importante señalar que en ambas partes de la tarea, la 1 y la 2, se siguieron estándares para que los resultados no estuvieran comprometidos, esto es, utilizar la misma normalización y *clean* en *training* que en *testing*, y no entrenar con datos de *testing* los modelos.

Los métodos utilizados comprobaron ser robustos y esto se muestra en que se obtuvieron resultados del 100% con ambos tipos de LBP en la parte 1, y de 100% y 98% en la parte 2 con el LBP normal y el invariante a las escalas, respectivamente. La robustez se vio verificada al cambiar los *datasets* utilizando para una parte las caras pares y para otra las impares, como era pedido y que los resultados fueran altos.

Este trabajo es relevante para observar la utilidad de métodos como SFS para seleccionar características a partir de un gran conjunto de ellas, y de LBP para extraer las mismas. Sin embargo, también sirvió para mostrar que un análisis acabado del comportamiento de los algoritmos puede ayudar a mejorar los resultados, como lo fue el hallazgo de las combinaciones de *features* seleccionadas, post SFS.

¹ Se necesita, para la correcta ejecución del código, que en la misma carpeta donde se encuentran los *Python notebooks* se encuentre también una carpeta llamada *faces_ARLQ* con las imágenes del *dataset*.