

Tarea 2 - Reconocimiento de Patrones

Pablo Messina

Pontificia Universidad Católica de Chile, DCC, IMFD, IAlab

Motivación - Segunda tarea del curso Reconocimiento de Patrones. El propósito de esta tarea es familiarizarse con el uso de técnicas de extracción, transformación y selección de características de textura para clasificación de imágenes reales. En particular en esta tarea se busca clasificar patches de murallas en dos posibles clases: limpio o rayado.

SOLUCIÓN PROPUESTA

Se decidió extraer 4 tipos de características: Local Binary Patterns (LBP), Histogram of oriented gradients (HOG), filtros de Gabor y texturas de Haralick. Para la extracción de todas estas características se hizo uso de la librería scikit-image (<https://scikit-image.org/>).

En el caso de LBP, se extrajo un vector de 59 dimensiones (58 bins de patrones uniformes y 1 bin para $256 - 58 = 198$ patrones no-uniformes). En HOG se obtuvo un vector de tamaño 128 ($4 \times 4 = 16$ grupos de 16×16 píxeles cada uno, con 8 orientaciones). En el caso de los filtros de Gabor se probaron todas las combinaciones de 2 frecuencias, 8 orientaciones y 2 desviaciones estándar, obteniéndose un vector de 64 dimensiones. Finalmente en el caso de Haralick se consideraron 3 distancias, 4 ángulos y 6 texturas (*contrast*, *dissimilarity*, *homogeneity*, *ASM*, *energy* y *correlation*), con lo cual se obtuvo un vector de 72 características. Por lo tanto, el vector final resultante es un vector de 323 características. Todas las características (columnas) del set de train se normalizaron restando la media y dividiendo por la desviación estándar. Estos mismos valores se re-usaron para normalizar las características en el set de test.

Para transformar y seleccionar características se usó Principal Component Analysis (PCA). La clasificación final se hizo con el método KNN (K nearest neighbors), con $K=3$, por requisitos de enunciado.

EXPERIMENTOS REALIZADOS

Se probaron variaciones de 2 hiperparámetros: la dimensión final de reducción del PCA, y dos funciones de distancia: distancia coseno y distancia euclídeana. Luego de probar muchas combinaciones, la mejor combinación resultó ser distancia euclídeana y PCA(18) (las primeras 18 dimensiones obtenidas por PCA), con lo cual se obtuvo un accuracy de 95.7%. La figura 1 muestra la matriz de confusión correspondiente a esta mejor versión.

CONCLUSIONES

De los resultados empíricos obtenidos podemos concluir que se puede obtener un accuracy de clasificación bastante decente (95.7%) usando sólo un vector de 18 características y

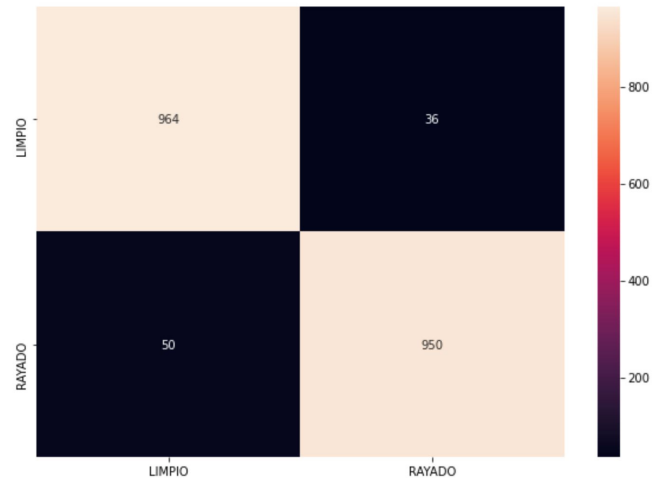


Fig. 1. Matriz de Confusión

un clasificador bien sencillo (KNN con $K=3$). Para lograr un desempeño así con tan pocas características, hay dos elementos claves que destacar: 1) tener buenos algoritmos de extracción de características (ej. LBP, HOG, Gabor, Haralick) y 2) tener un buen algoritmo de selección (ej. PCA). Si uno quisiera mejorar aún más el desempeño de clasificación, se podría explorar el uso de otros descriptores (como BSIF, deep learning, etc.), probar otras formas de transformación y selección de características (ej. autoencoders) y probar otros algoritmos de clasificación (ej. regresión lineal, SVM, red neuronal).