

Tarea 02 – Reconocimiento de Lunares Cancerígenos

Pedro Pablo Zavala Tejos
Pontificia Universidad Católica

Motivación – El desafío de esta tarea fue encontrar la manera óptima de preprocesar las imágenes y obtener buenas características para asegurar una clasificación precisa de los tipos de lunares. Al abordar la tarea, aprendí que la calidad de los datos es un factor crucial en el desempeño del modelo.

I. SOLUCIÓN PROPUESTA

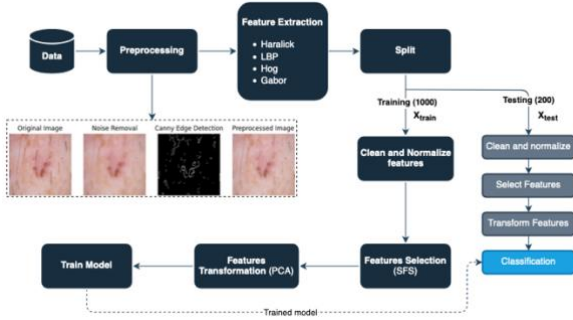


Figura 1: Diagrama de flujo

Iniciamos preprocesando las imágenes para optimizar la calidad de los datos. Esto implicó un tratamiento individual de los canales RGB. Específicamente, redujimos el ruido alrededor del lunar y segmentamos utilizando el algoritmo de *Canny* para detectar los bordes del lunar. Luego, aplicamos transformaciones *Black Hat* y *Top Hat* para mejorar el contraste del lunar mediante la fórmula $enhance = channel + tophat - blackhat$. Finalmente, mezclamos los resultados de cada canal en una imagen RGB.

Seguidamente, extraemos características de textura e intensidad (Hog, Gabor, LBP y Haralick), concatenándolas para formar una matriz X . Cada $x_{ij} \in X$ refleja a una ponderación de la extracción de característica para cada canal, en donde $\alpha = \gamma = 0.125$ y $\beta = 0.75$

$$x_{ij} = x_{ij}^R \cdot \alpha + x_{ij}^G \cdot \beta + x_{ij}^B \cdot \gamma$$

Obtuvimos un total de 1055 características. Utilizamos *Clean* para eliminar la alta correlación, normalizamos X y seleccionamos las 100 mejores características con SFS. Luego, reducimos la dimensionalidad de X a 11 componentes utilizando PCA. Y finalmente, entrenamos un modelo KNN con $k=1$ vecinos cercanos para la clasificación.

II. EXPERIMENTOS REALIZADOS

Durante el preprocesamiento, realizamos segmentaciones para detectar los bordes del lunar utilizando Filtros de *Laplacian* y *Canny Edge*, observando que este último ofreció mejores resultados que el primero. Además, mejoramos el contraste de la imagen, lo que mejoró la separación de las clases.

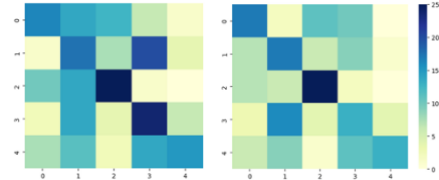


Figura 2: Matriz de confusión de *testing* antes y después del preprocesamiento

Para la clasificación, evaluamos nuestro modelo KNN utilizando diferentes valores de $k \in \{1, 3, 5\}$ para los vecinos cercanos, y diferentes cantidades de componentes $q \in [1, 30]$ que nos permitieron obtener la mejor precisión en el modelo (*accuracy* total óptimo). Utilizando la matriz X con 100 características seleccionadas por SFS, al analizar los resultados en el código (sección 6: *Evaluation*), identificamos que la mejor configuración fue para $q = 11$ y $k = 1$.

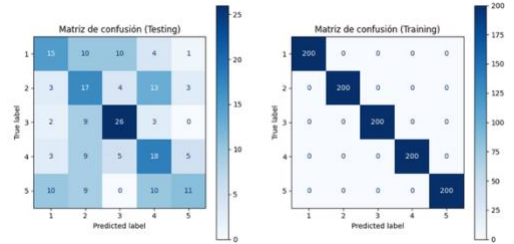


Figura 3: Matriz de confusión de *testing* y *training* para la clasificación final

Como resultado, nuestro modelo logró clasificar las clases con una precisión del 43.5% en los datos de prueba y del 100% en el de entrenamiento, obteniendo así un *accuracy* total del 54.8%.

III. CONCLUSIÓN

Nuestro modelo no logró clasificar correctamente, especialmente los datos de prueba, donde el 56.5% fueron mal clasificados. Esta falta de precisión podría haber sido afectada por factores como el nivel de preprocesamiento de imágenes. Nuestro próximo paso será utilizar técnicas de preprocesamiento más avanzadas para extraer características de mejor calidad y mejorar la precisión del modelo.