

Visión por Computadora 2025

Lab 1

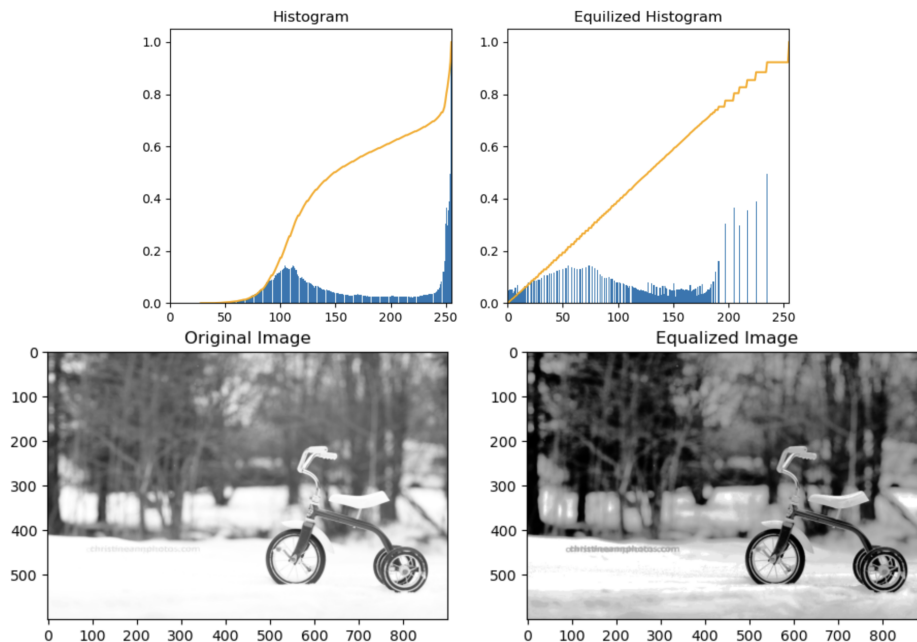
30.enero.2025

1. Implementar en Python una función que haga la ecualización de histograma para una imagen en escala de grises (8 bits).

Mostrar a través de ejemplos la imagen original y la imagen ecualizada. Mostrar también los histogramas normalizados y su distribución acumulada de ambas imágenes.

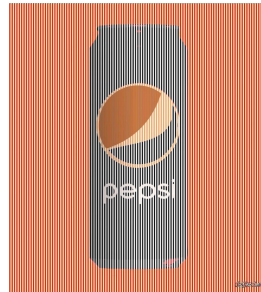
Comparar los resultados de su función contra alguna función ya predefinida en Python para calcular ecualizaciones. Por ejemplo, en scikit-image, tenemos la función `skimage.exposure.equalize_hist`; o en OpenCV, tenemos la función `cv2.equalizeHist`.

¿Son iguales sus resultados con los de la función de Python? Explique por qué son iguales o explique por qué hay diferencias.

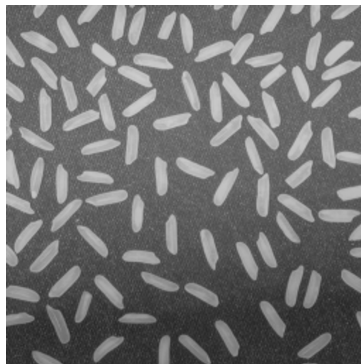


2. Implementar 3 algoritmos para segmentación binaria de imágenes. De estos, uno debe ser un algoritmo local.
Aplicar cada uno de estos algoritmos a al menos 3 imágenes en escala de grises diferentes y comparar los resultados obtenidos de cada segmentador. (Mostrar visualmente la imagen original y las binarizaciones).
Concluya cuál de sus algoritmos obtiene mejores resultados.
3. Implementar un algoritmo de segmentación para imágenes RGB, utilizando de fondo un algoritmo de k -medias. Se debe permitir al usuario elegir el parámetro $k \geq 2$ del número de segmentos que desea obtener.
Mostrar con varios ejemplos (2 buenos y 2 malos) los alcances y limitaciones de este algoritmo. Para cada caso, mostrar
 - a) la imagen original,
 - b) el mapa de segmentos o clases,
 - c) la imagen cuantizada (promedio de color en cada segmento).

4. Considere las imágenes `coca-cola.jpeg` y `pepsi.png`. En cada una, convierta la imagen RGB a un espacio de color cromático adecuado, y elabore el histograma del canal de cromaticidad. A partir de analizar este histograma, concluya qué está ocurriendo con los colores en cada imagen.



5. Implemente un algoritmo de segmentación binaria que identifique correctamente **todos** los granos de arroz en la imagen `rice.png`. (No se puede aplicar un método local.)



6. Implementar un detector de color YELLOW que funcione en tiempo real en un video. Para ello, implementar el algoritmo en OpenCV, de modo que capture las imágenes directamente de la cámara de su PC, y que muestre el resultado de la detección en tiempo real en pantalla.

Si usted lo desea, puede diseñar el detector para que detecte otros colores diferentes. Mostrar en su informe, resultados (*screenshots*) del funcionamiento de su detector.