UNIVERSIDAD AUSTRAL DE CHILE

FACULTAD DE CIENCIAS DE LA INGENIERÍA

INGENIERÍA CIVIL ELECTRÓNICA

ELEP 233 - VISIÓN ARTIFICIAL Y REDES NEURONALES

DESARROLLO DE TAREA N°2

PROFESOR:
Gustavo Schleyer.

INTEGRANTES:
Ángel Andrade
Jorge Palavecino.

10 de mayo de 2023

Índice general

La pregunta 1 no se encuentra presente en este documento debido a que no se pide el resultado en pdf. El repositorio en GitHub de la actividad puede encontrarse en `https://github.com/Atrabilis/UACH/tree/main/Trabajos/Vision%20artificial/Tarea%202`

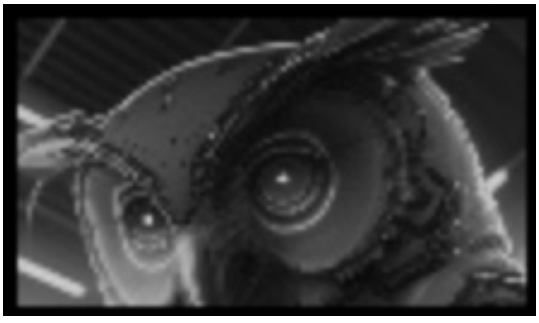
Pregunta 2

En el siguiente ejemplo se comparan el zoom digital recreado en Python y el implementado en openCV.

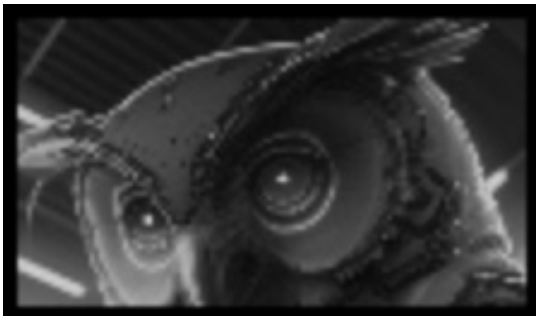
La imagen de referencia es la siguiente:

Figura 1: Imagen de referencia

A continuación se muestra el zoom obtenido mediante la recreación del algoritmo en Python y el zoom implementado en openCV (2x):



(a) zoom obtenido con el algoritmo implementado (2x).



(b) Zoom obtenido con openCV (2x)

Figura 2: Comparación entre el algoritmo implementado y el de openCV.

No se pueden ver mayores diferencias a simple vista, sin embargo el algoritmo implementado por openCV es mucho mas eficiente en runtime.

Comparando las patas del búho con un factor 3x se obtiene:

(a) zoom obtenido con el algoritmo implementado (3x).

(b) Zoom obtenido con openCV (3x)

Figura 3: Comparación entre el algoritmo implementado y el de openCV.

La diferencia es poco perceptible.

Problema 3

Suma

Las imágenes de referencia son:

(a) Referencia para la suma (img1)

(b) referencia para la suma (img2)

Figura 4: Imágenes de referencia utilizadas.

Los resultados del algoritmo comparados con los de openCV son los siguientes:

(a) suma implementada en python.

(b) suma hecha con openCV.

Figura 5: Resultados suma directa

Los resultados son casi idénticos, en este caso la diferencia en runtime es mínima, al menos para imágenes de este tamaño.

Comparando con la suma ponderada:

(a) suma ponderada 0.7 img1 · 0.3 img2.

(b) suma ponderada 0.3 img1 · 0.7 img2.

Figura 6: Resultados suma ponderada

Resta

Las imágenes de referencia son las de la figura 4

los resultados obtenidos fueron los siguientes:

(a) img1 - img2, algoritmo implementado.

(b) img1 - img2 algoritmo de openCV.

Figura 7: img1 - img2.

(a) img2 - img1, algoritmo implementado.

(b) img2 - img1 algoritmo de openCV.

Figura 8: img2 - img1.

Nuevamente los resultados son imperceptibles y la diferencia en runtime es mínima, esto debido a que el algoritmo básicamente es el mismo que el de la suma

Multiplicación

La imagen de referencia es la presente en la figura 4a. Los resultados son los siguientes:

(a) img1 · 3, algoritmo implementado.

(b) img1 · 3, algoritmo de openCV.

Figura 9: img1 · 3.

Nuevamente los resultados son imperceptibles. La diferencia mínima entre ambas puede deberse a que openCV redondea de manera distinta a la función round() de numPy.

División

La imagen de referencia es la presente en la figura 4a. Los resultados son los siguientes:

(a) img1 / 2.5, algoritmo implementado.

(b) img1 / 2.5, algoritmo de openCV.

Figura 10: img1 /2.5.

Nuevamente los resultados son imperceptibles y el runtime es similar, esto se debe a que el algoritmo de división es básicamente el mismo que el de multiplicación.

Problema 4

La imagen de referencia es la presente en la figura 1. Los resultados de los filtros implementados son los siguientes:

(a) Filtro promedio implementado en Python.

(b) Filtro medio implementado en Python.

Figura 11: Resultados de la implementación de los filtros a la imagen de referencia.

Se puede observar que el filtro medio es mas “afilado” que el filtro promedio. Al contaminar la imagen de referencia con ruido sal y pimienta se obtiene lo siguiente:

Figura 12: Imagen contaminada con ruido sal y pimienta

Al filtrar esta imagen con los algoritmos implementados se obtiene lo siguiente:

(a) Filtro promedio implementado en Python aplicado a la imagen ruidosa.

(b) Filtro medio implementado en Python aplicado a la imagen ruidosa.

Figura 13: Resultados de la implementación de los filtros a la imagen ruidosa.

Se puede ver que el filtro promedio suaviza la imagen mientras que el filtro medio crea una imagen mas afilada. Debido al resultado de las operaciones de ambos filtros algunos puntos de sal y pimienta desaparecen en ambos casos.

Comparando con el filtro de mediana de opencv utilizando un tamaño de ventana de filtro de 5 se obtiene lo siguiente:

Figura 14: Filtro medio (medianBlur()) aplicado a la imagen ruidosa.

El filtrado de openCV es mas agresivo con los puntos pero también degrada mas la imagen que los filtrados anteriores, esto debido principalmente al ancho de ventana de filtrado utilizado. Utilizando un tamaño de ventana de filtrado de 3 se obtiene:

Figura 15: Filtro medio (medianBlur()) aplicado a la imagen ruidosa (ksize = 3).

El cual es mucho mas parecido al obtenido mediante nuestra implementación.

Problema 5

A continuación se presentan 3 histogramas obtenidos con el algoritmo implementado y se comenta en como afectarían a la imagen a la cual se ajustan. El diagrama original corresponde al histograma de la imagen en la figura 1.

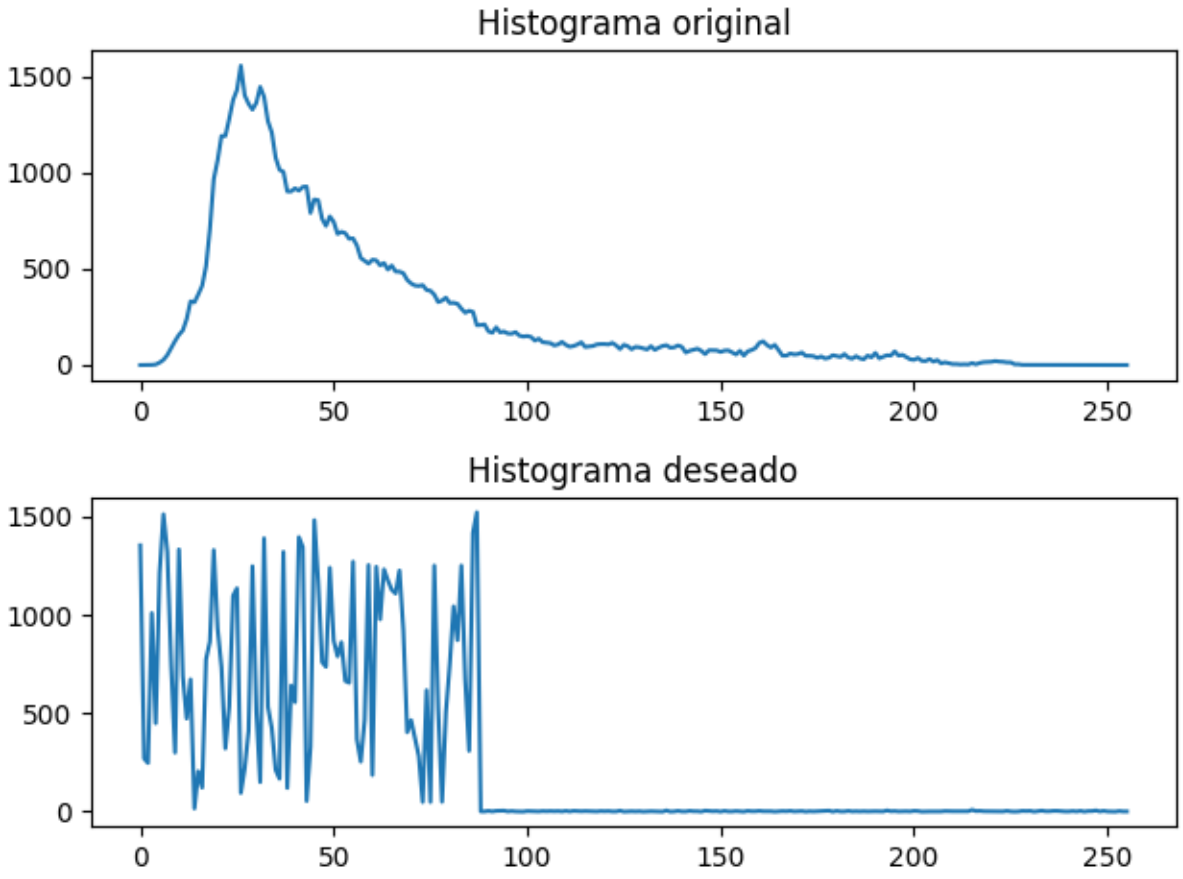


Figura 16: Histograma 1.

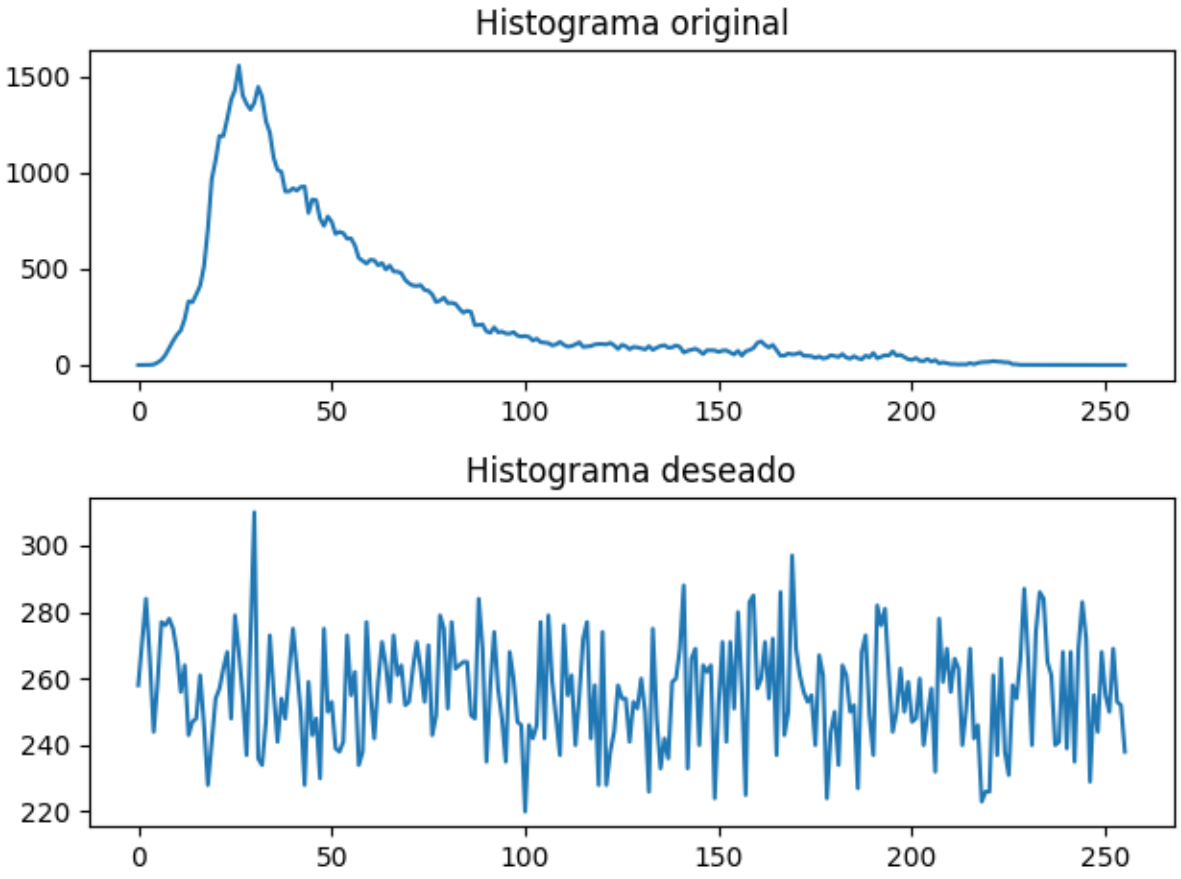


Figura 17: histograma 2.

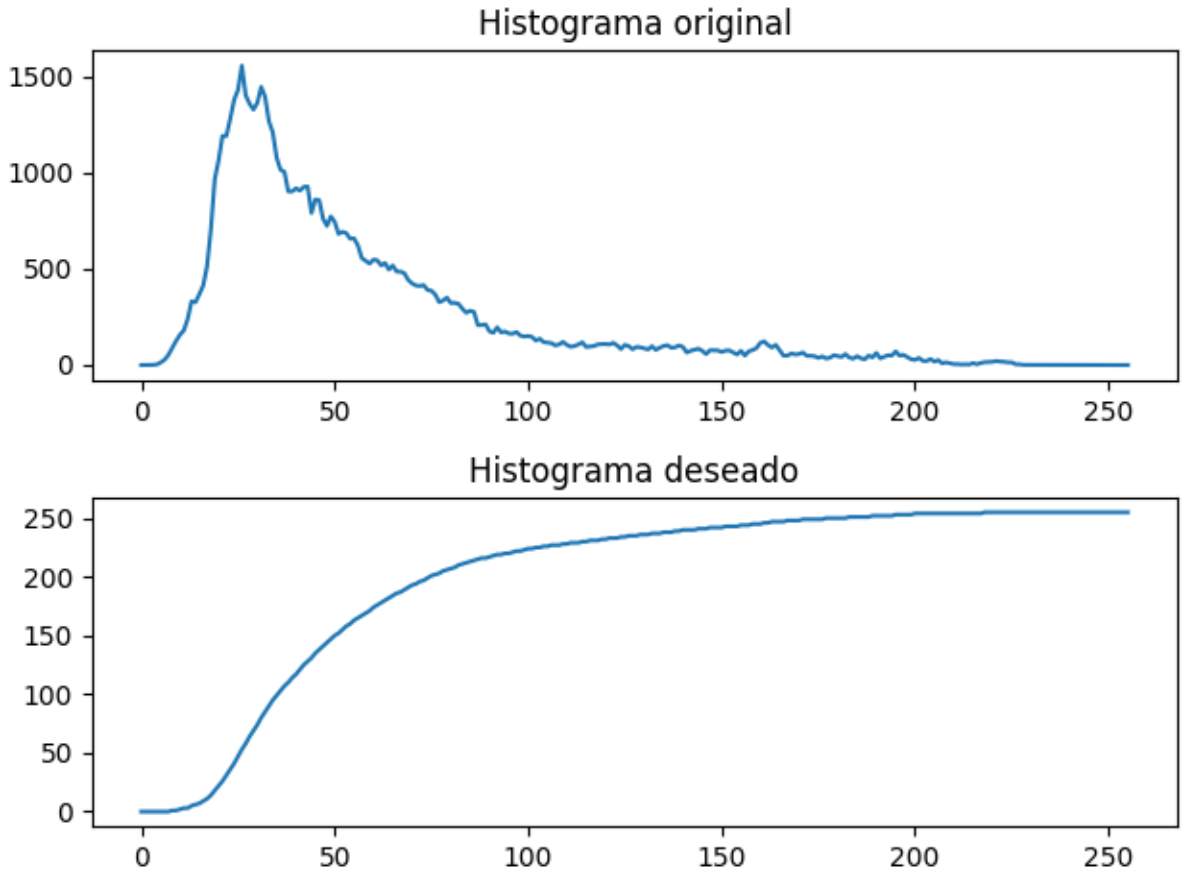


Figura 18: histograma 3.

Aplicar el histograma 1 (16) la imagen sería mucho mas uniforme en sus escalas de grises (ya que casi todos los píxeles presentan valores cercanos a 35), como si hubiera sido contaminada con ruido pimienta, esto debido a la gran cantidad de píxeles con valores de gris menores a 100, este histograma haría dominantes unos pocos valores de gris, que estarían bien diferenciados entre si.

Aplicar el histograma 2 (17) resultaría en un resultado parecido al del histograma 1 (16) pero los valores de grises dominantes estarían repartidos en todo el espectro.

Aplicar el histograma 3 (18) resultaría en un aclaramiento de la imagen, ya que tomaría los píxeles con niveles de grises menores a 100 (la mayoría) aproximadamente y los repartiría en todo el espectro hasta llegar a 255, por lo que los niveles de grises estarian mas elevados en comparación con la imagen original.