

Visión Por Computadora I - TPI

Abril 2024



FABIÁN ANDRÉS URIBE GUERRA
Estudiante

• Parte 1.1

Implementar el algoritmo de pasaje a coordenadas cromáticas para librarnos de las variaciones de contraste.

1.1.1 CoordCrom_1.png:



1.1.2 CoordCrom_3.png:



1.1.3 CoordCrom_2.png:



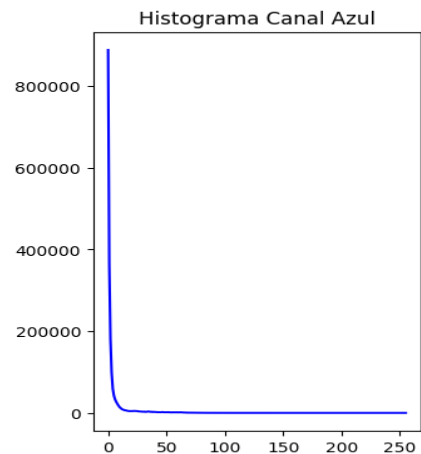
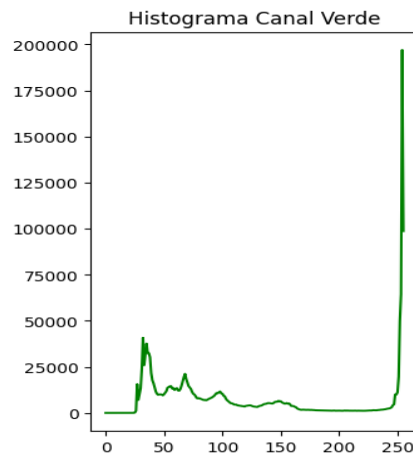
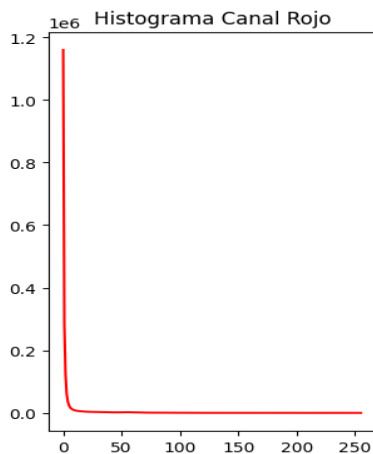
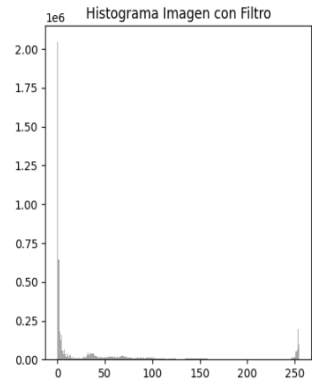
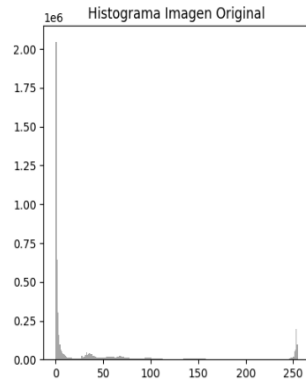
▪ Parte 1.2

Implementar el algoritmo **White Patch** para librarnos de las diferencias de color de iluminación.

1.2.1 wp_green2.jpg:



Imagen: wp_green2.jpg
Intensidad máxima global (Original): 255
Distancia entre los histogramas: 0.02595072862052342



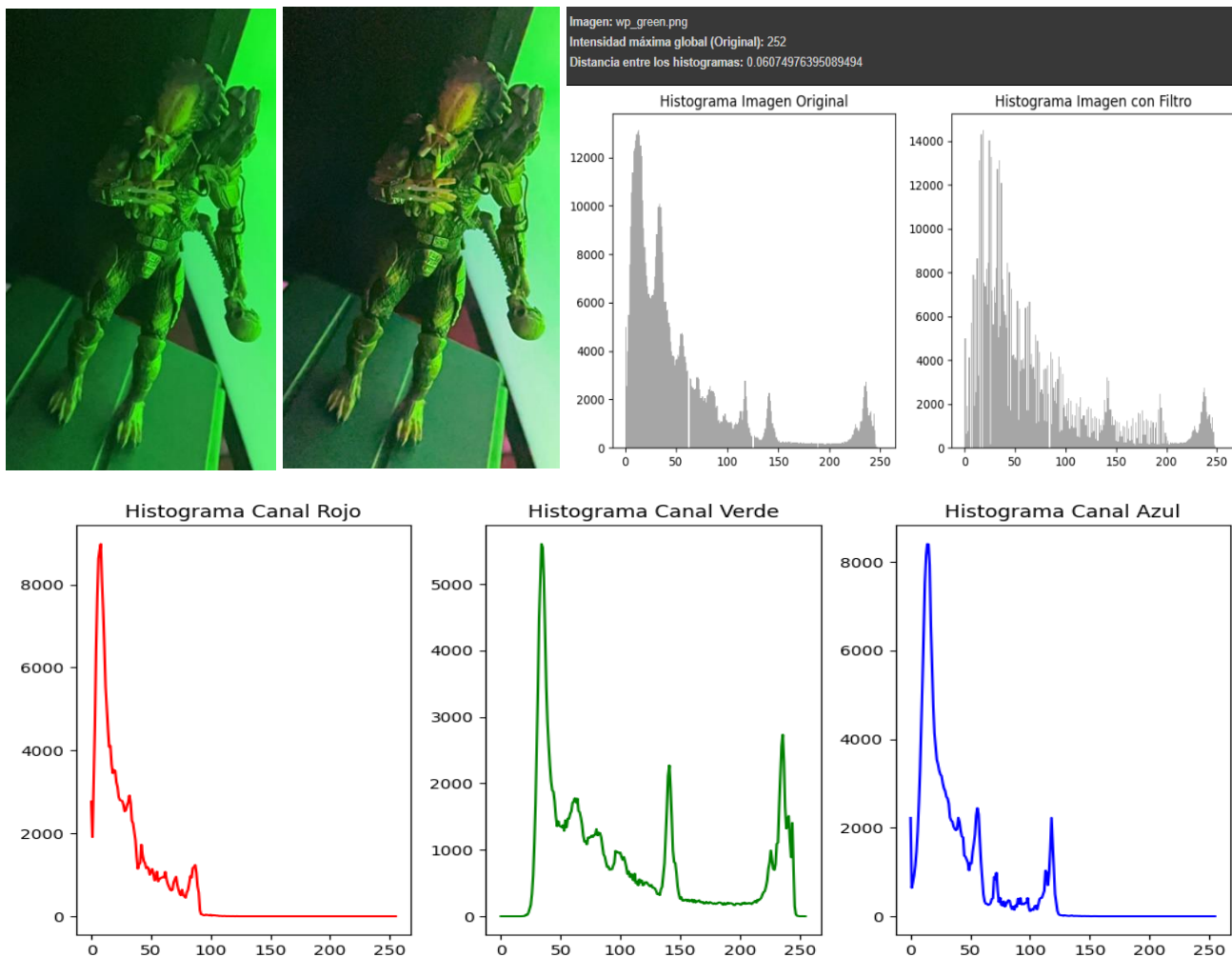
▪ Resultados:

- Subexposición al rojo y azul y sobre exposición al verde.
- Saturación extremadamente baja en Rojo y Azul (Nula). Por ello el filtro no tiene mayor impacto.
- Distancia entre histogramas es baja. Lo cual sugiere que las distribuciones son relativamente similares entre sí.
- No se evidencia un gran cambio debido a los sesgos extremos identificados en el histograma de la imagen.

▪ Parte 1.2

Implementar el algoritmo **White Patch** para librarnos de las diferencias de color de iluminación.

1.2.3 wp_green.jpg:



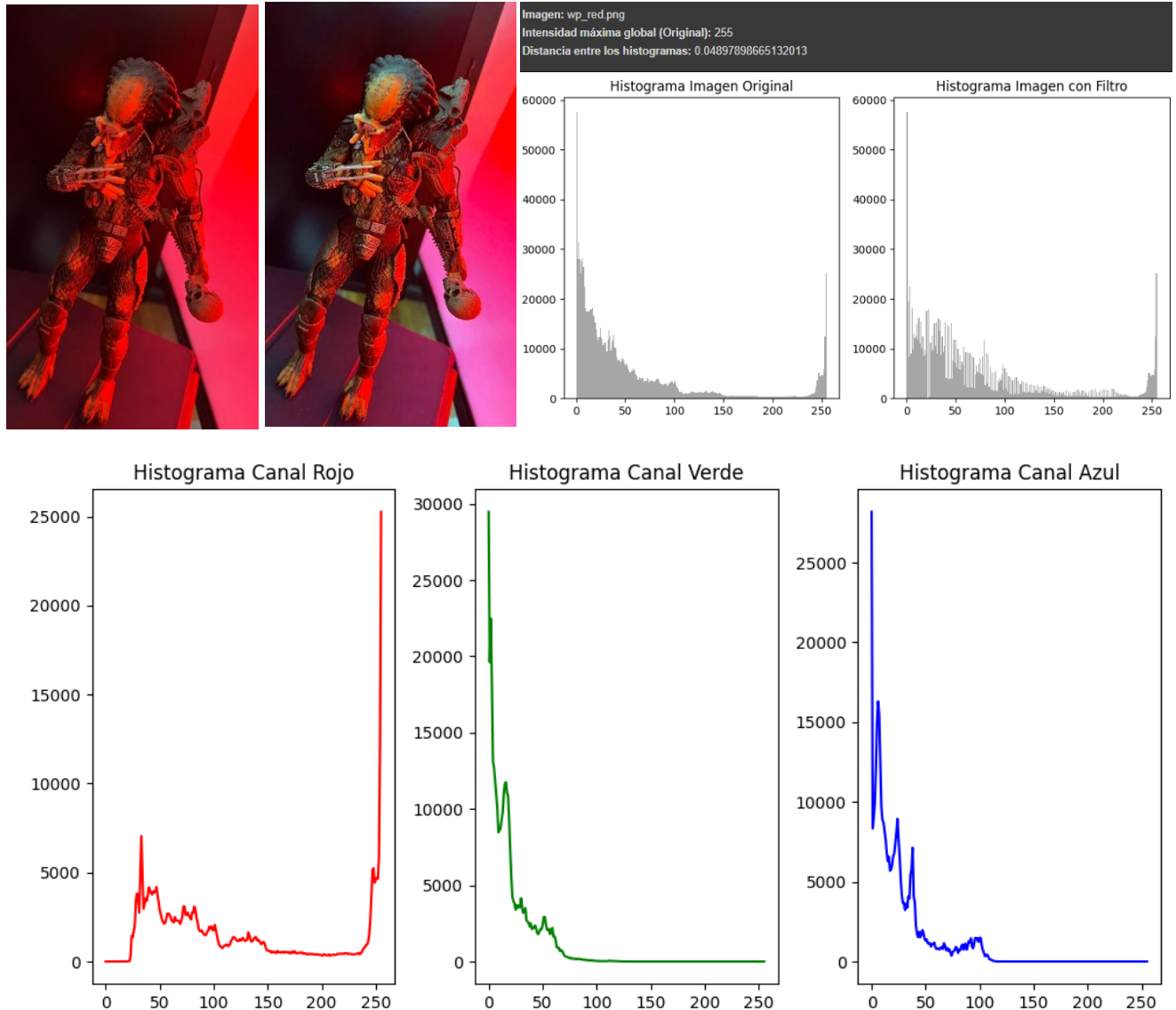
▪ Resultados:

- Subexposición al rojo y azul y sobre exposición al verde.
- Saturación similar en los 3 colores, aunque con mayor énfasis en el verde.
- Distancia entre histogramas es baja. Lo cual sugiere que las distribuciones son relativamente similares entre sí.
- Mejor aplicado el filtro porque los histogramas muestran una distribución similar en los tres canales. Adicionalmente, el histograma general muestra una mejor distribución comparada con la anterior imagen.

▪ Parte 1.2

Implementar el algoritmo **White Patch** para librarnos de las diferencias de color de iluminación.

1.2.6 wp_red.jpg:



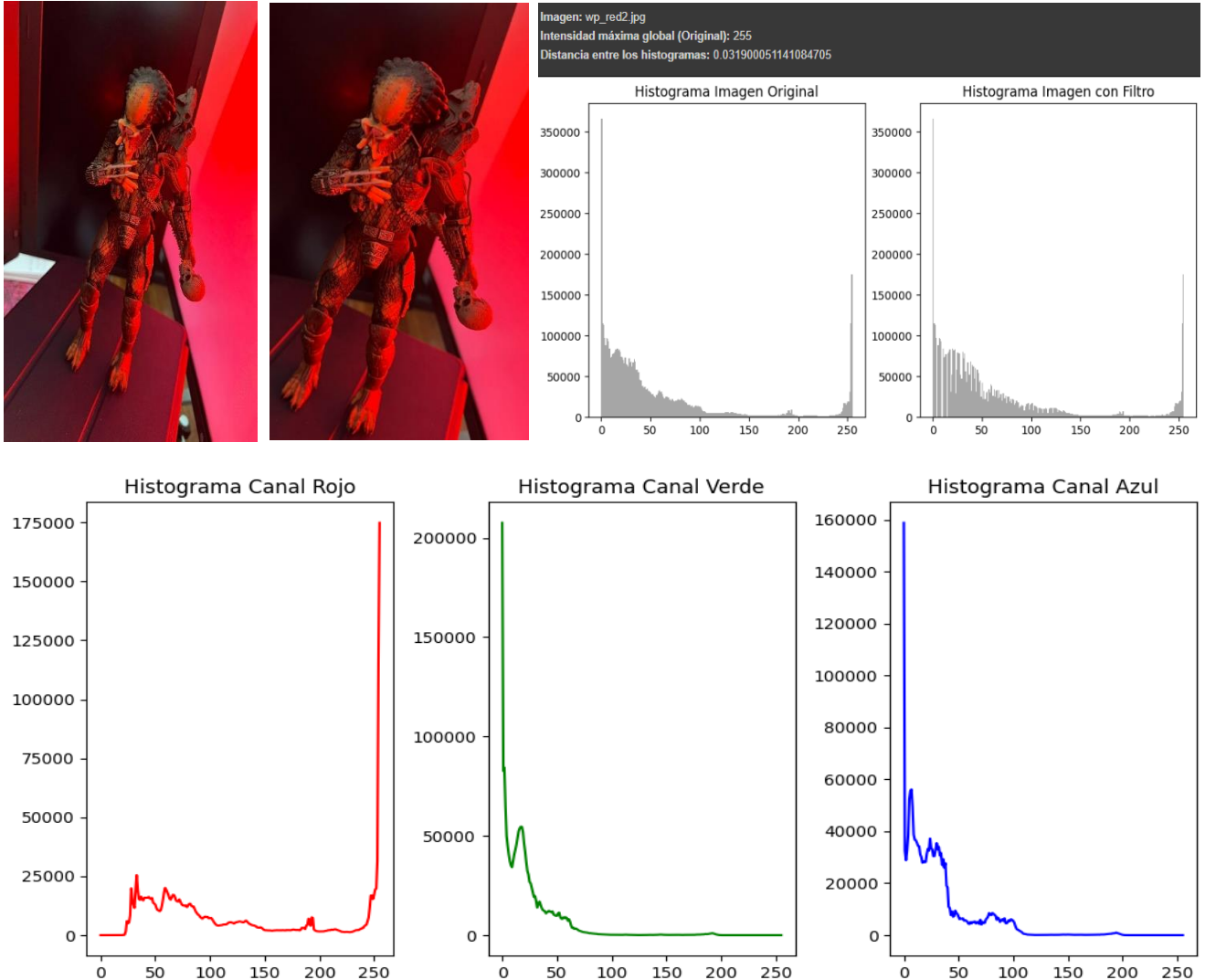
▪ Resultados:

- Subexposición al verde y azul y sobre exposición al rojo.
- Mayor saturación del color rojo.
- Baja saturación de verde y azul pero igual se evidencian pixeles de estos colores.
- Distancia entre histogramas es baja. Lo cual sugiere que las distribuciones son relativamente similares entre sí con o sin filtro.
- Se identifica el filtro.

▪ Parte 1.2

Implementar el algoritmo **White Patch** para librarnos de las diferencias de color de iluminación.

1.2.8 wp_red2.jpg:



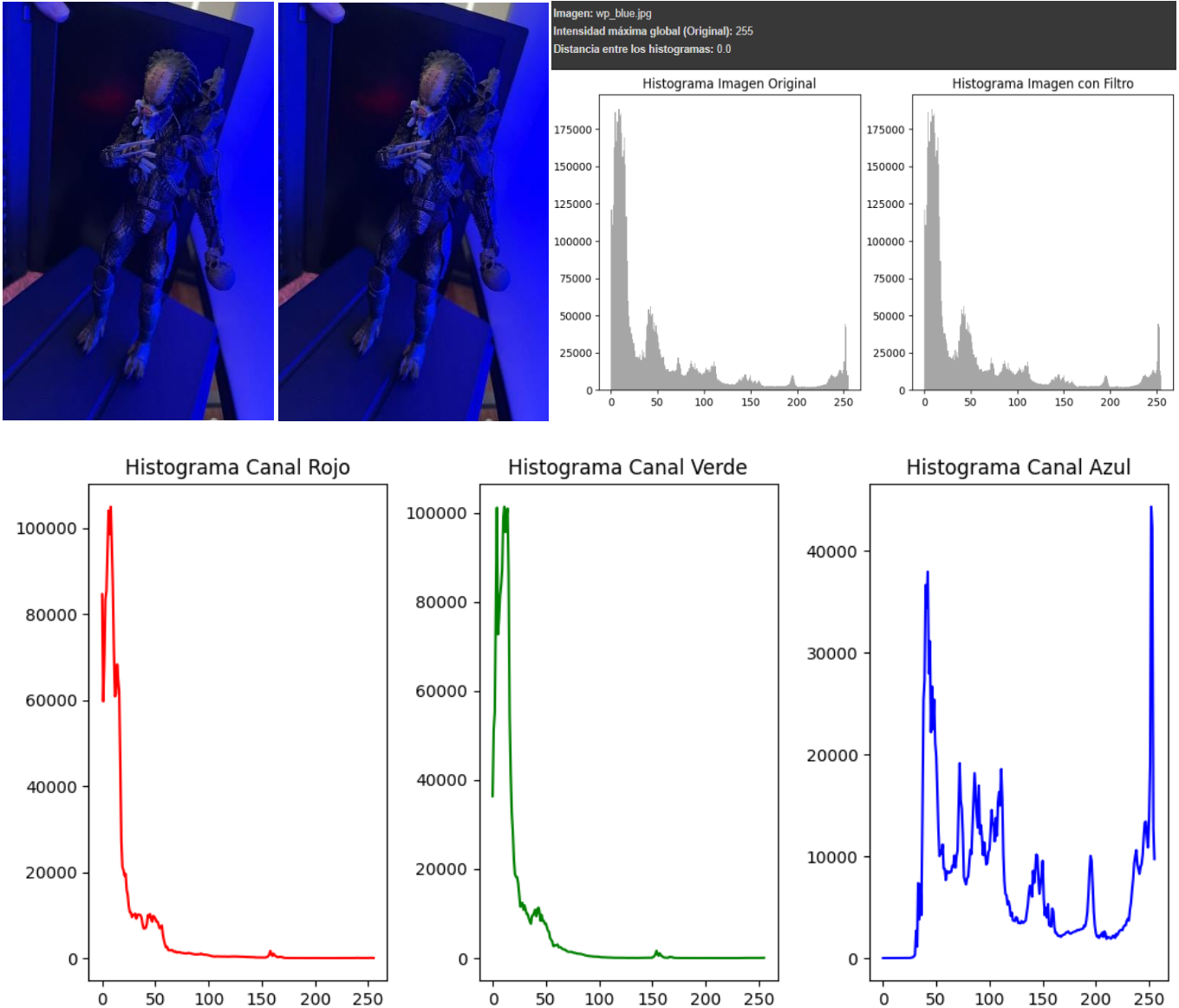
▪ Resultados:

- Subexposición al verde y azul y sobre exposición al rojo.
- Mayor saturación del color rojo.
- Baja saturación de verde y azul pero igual se evidencian pixeles de estos colores.
- Distancia entre histogramas es baja. Lo cual sugiere que las distribuciones son relativamente similares entre sí con o sin filtro.
- Al triplicarse el número de pixeles en los tres canales, el filtro no tiene un impacto similar a la anterior imagen.
- Se identifica el filtro.

▪ Parte 1.2

Implementar el algoritmo **White Patch** para librarnos de las diferencias de color de iluminación.

1.2.7 wp_blue.jpg:



▪ Resultados:

- Subexposición al verde y rojo y sobre exposición al azul.
- Mayor saturación del color azul
- Baja saturación de verde y rojo, pero igual se evidencian pixeles de estos colores.
- Distancia entre histogramas es baja. Lo cual sugiere que las distribuciones son relativamente similares entre sí con o sin filtro.
- Mayor cantidad de pixeles (> 60K) imagen después de aplicar filtro no varía demasiado.

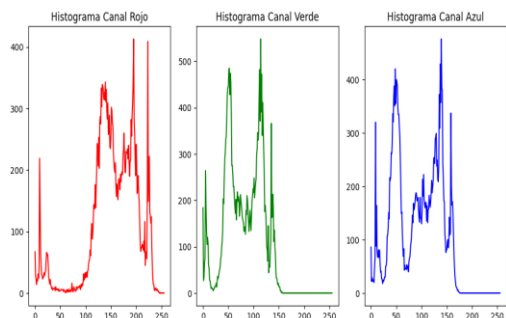
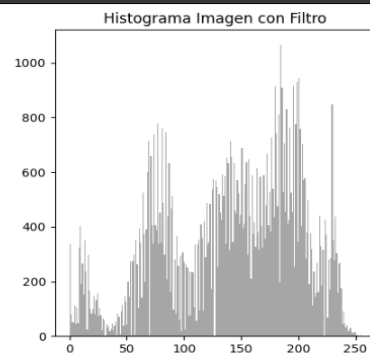
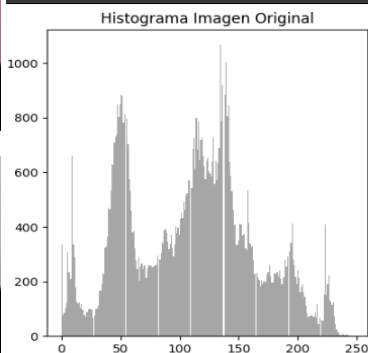
Parte 1.2

Implementar el algoritmo White Patch para librarnos de las diferencias de color de iluminación.

1.2.4 test_red.png



Imagen: test_red.png
Intensidad máxima global (Original): 247
Distancia entre los histogramas: 0.06083691383052903



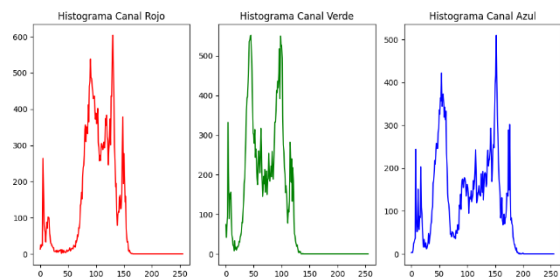
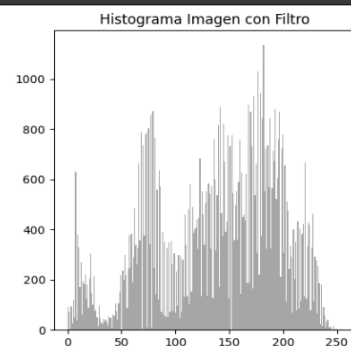
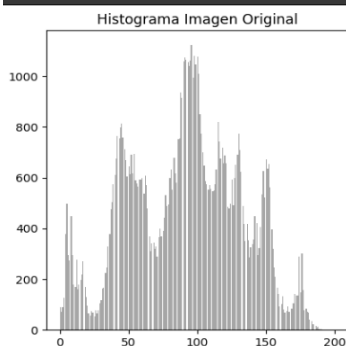
Resultados:

- Exposiciones similares de los 3 canales
- Mayor saturación del color rojo.
- Baja saturación de verde y azul pero igual se evidencian pixeles de estos colores.
- Distancia entre histogramas es baja. Lo cual sugiere que las distribuciones son relativamente similares entre sí con o sin filtro.
- Se identifica el filtro.

1.2.2 test_blue.png:



Imagen: test_blue.png
Intensidad máxima global (Original): 200
Distancia entre los histogramas: 0.0676722016938377



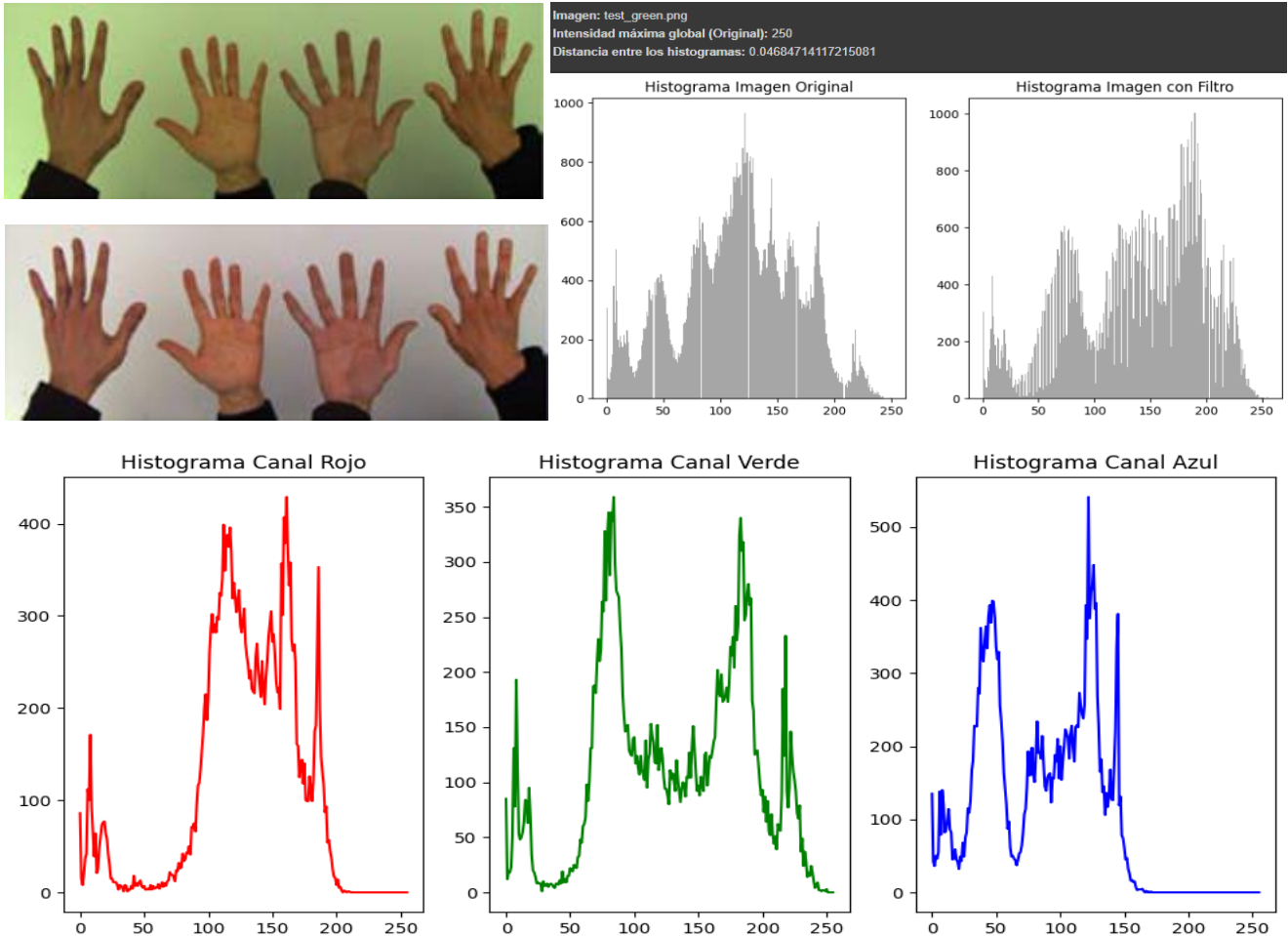
Resultados:

- Exposiciones similares de los 3 canales
- No se evidencia una sobreexposición en el azul.
- Distancia entre histogramas es baja. Lo cual sugiere que las distribuciones son relativamente similares entre sí con o sin filtro.
- Se identifica el filtro.

Parte 1.2

Implementar el algoritmo White Patch para librarnos de las diferencias de color de iluminación.

1.2.5 test_green.png



▪ Resultados:

- Exposiciones similares de los 3 canales
- No se evidencia una sobreexposición significativa en el verde.
- Distancia entre histogramas es baja. Lo cual sugiere que las distribuciones son relativamente similares entre sí con o sin filtro.
- Se identifica el filtro y se tiene buenos resultados considerando que el histograma tiene una distribución gaussiana.

▪ Parte 2.1

Para las imágenes img1_tp.png y img2_tp.png leerlas con OpenCV en escala de grises y visualizarlas.

2.1.1 img1_tp.png:

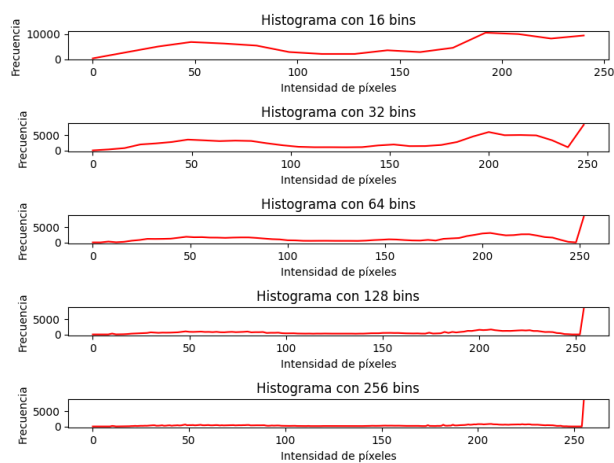


2.1.2 img2_tp.png:

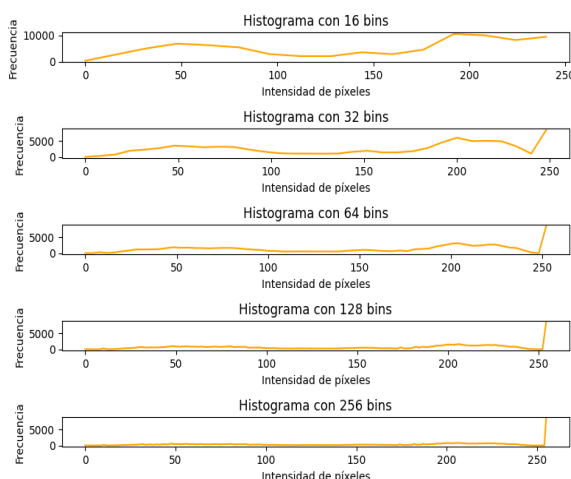


Elija el número de bins que crea conveniente y grafique su histograma, compare los histogramas entre sí. Explicar lo que se observa, si tuviera que entrenar un modelo de clasificación/detección de imágenes, considera que puede ser de utilidad tomar como 'features' a los histogramas?

2.1.1 img1_tp.png:



2.1.2 img2_tp.png:

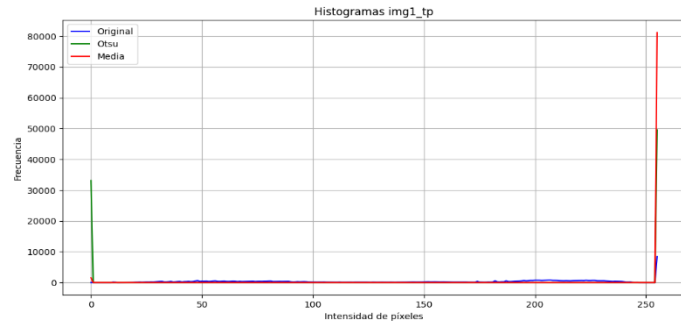
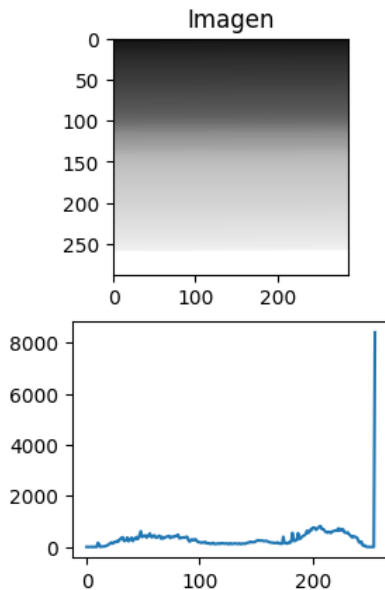


En base a los resultados podemos inferir que el histograma de la img1_tp.png tiene una distribución más uniforme en el rango de tonos lo cual es esperado debido a que es una imagen con gradaciones homogéneas. Por su parte, el histograma de la img2_tp muestra una mayor concentración en los colores más claros. Lo cual tiene sentido al considerar que la imagen es una flor cuyos pétalos son blancos. Ahora, respecto a la pregunta si considero que es de utilidad tomar como 'features' a los histogramas puedo comentar que A NIVEL GENERAL si ya que son útiles para inferir la distribución y frecuencia de los tonos presentes en la imagen lo cual permite diferenciar el tipo de imagen y es de utilidad para mejorar la precisión de los modelos de clasificación que se pretendan usar.

▪ Parte 2.1

Elija el número de bins que crea conveniente y grafique su histograma, compare los histogramas entre sí. Explicar lo que se observa, si tuviera que entrenar un modelo de clasificación/detección de imágenes, considera que puede ser de utilidad tomar como 'features' a los histogramas?

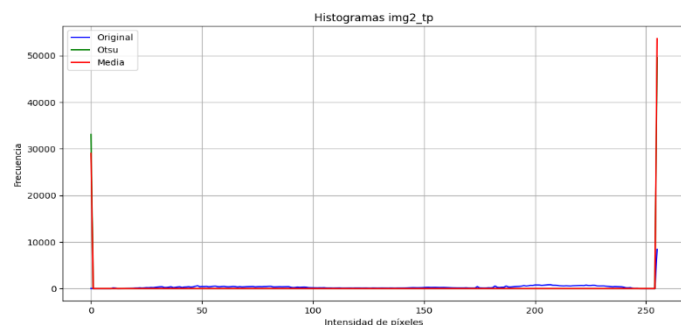
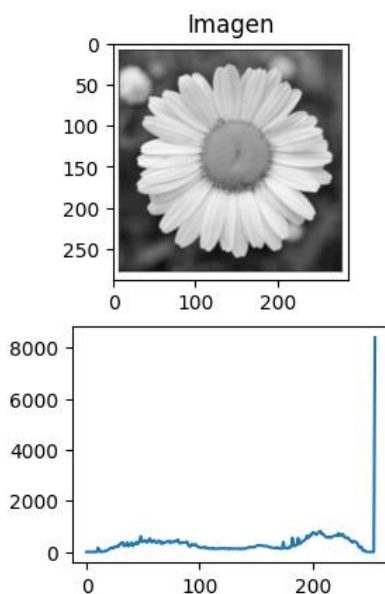
2.1.1 img1_tp.png:



▪ Resultados:

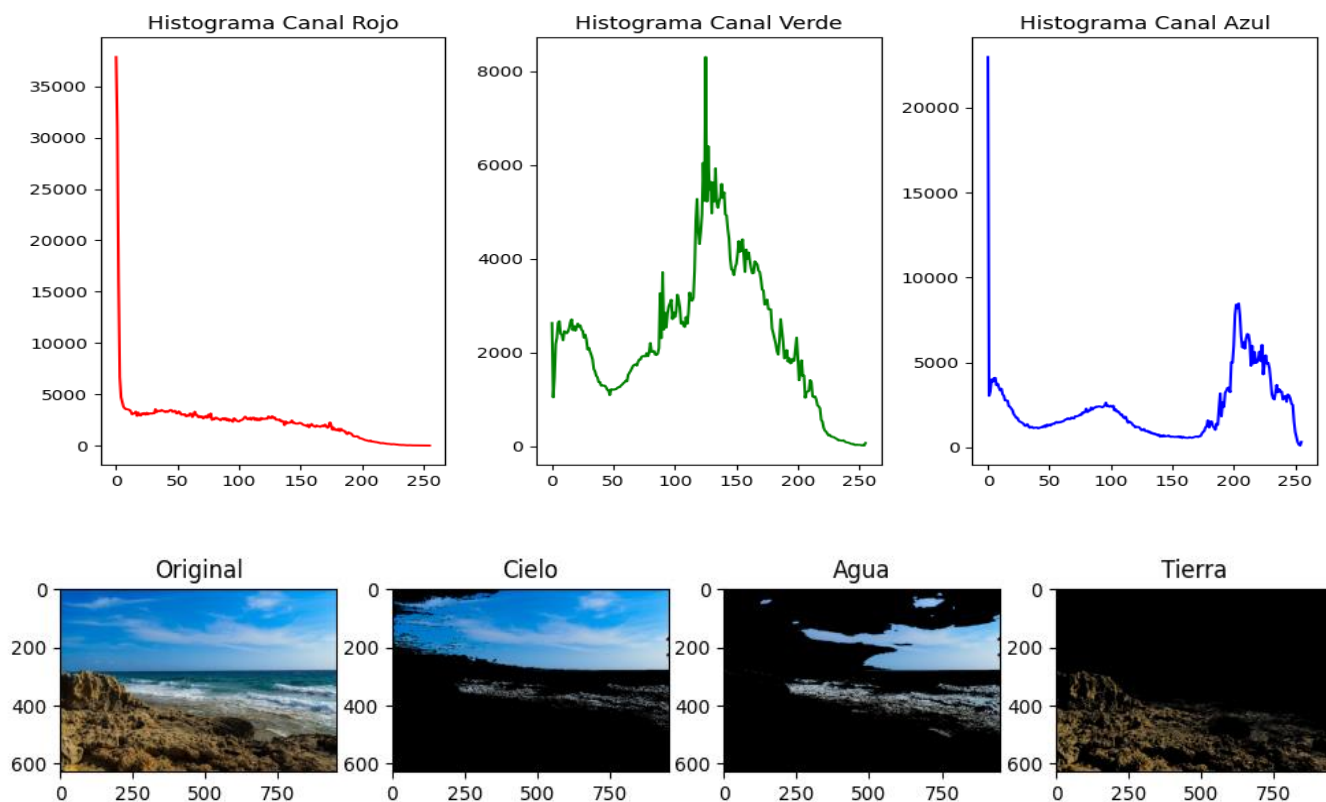
- Mayor cantidad de píxeles en tono 255.
- Al ser una imagen degradada tiene sentido el patrón circadiano antes del incremento en la escala mayor.
- Debido a los extremos, considero que en ESTE TIPO DE IMAGEN no es de utilidad tomar como *features* a los histogramas. El mismo análisis aplica para la imagen 2.1.2.

2.1.2 img2_tp.png:



▪ Parte 3.1

Para la imagen `segmentacion.png` analice el histograma de los canales RGB. Segmente algunos de los elementos presentes en la imagen (agua, cielo, tierra) y muestre, aplicando máscaras, las regiones en imágenes separadas



Para poder desarrollar este punto se extrajo el histograma de cada canal (RGB) y se encontró que el canal rojo presenta una distribución de intensidad más baja a lo largo de todo el rango por que se podría inferir que no hay una presencia dominante de tonos rojos en la imagen. Por su parte, el canal verde mantiene una forma gaussiana con picos en el rango medio lo que sugiere una mayor cantidad de verde en la imagen. Finalmente, el canal azul tiene unos ‘spikes’ en el lado derecho lo cual infiere la presencia de áreas de cielo y del mar. Sin embargo, en los canales azul y rojo se evidencia una subexposición del rojo y azul, aunque en estos canales se evidenció una mayor cantidad de píxeles.

Umbral de la máscara:

```
# Rango de azules (agua)
lower_blue = np.array([100, 150, 150]) # Ajuste para tonos claros de azul
upper_blue = np.array([200, 250, 255]) # Blanco para capturar espuma de mar

# Rango de verdes (cielo)
lower_green = np.array([0, 50, 200])
upper_green = np.array([200, 250, 255])

# Rango de marrones (tierra)
lower_brown = np.array([100, 80, 0]) # Ajuste para arena
upper_brown = np.array([200, 180, 100]) # Ajuste para tonos más oscuros de marrón

# Aplicar máscaras para segmentar los elementos
mask_blue = cv2.inRange(img_rgb, lower_blue, upper_blue)
mask_green = cv2.inRange(img_rgb, lower_green, upper_green)
mask_brown = cv2.inRange(img_rgb, lower_brown, upper_brown)

# Aplicar las máscaras a la imagen original
segmented_blue = cv2.bitwise_and(img_rgb, img_rgb, mask=mask_blue)
segmented_green = cv2.bitwise_and(img_rgb, img_rgb, mask=mask_green)
segmented_brown = cv2.bitwise_and(img_rgb, img_rgb, mask=mask_brown)
```