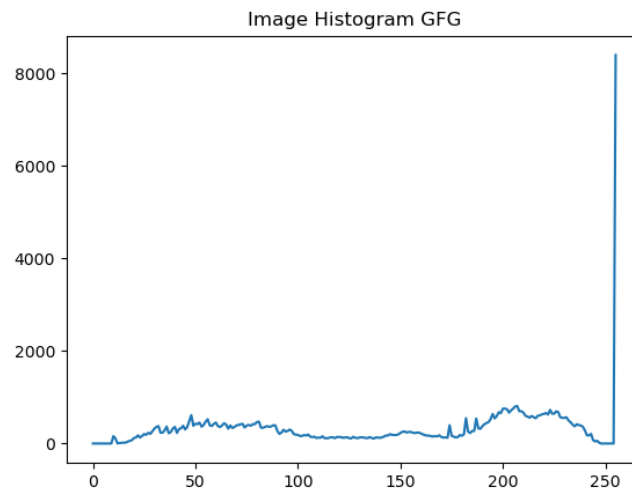
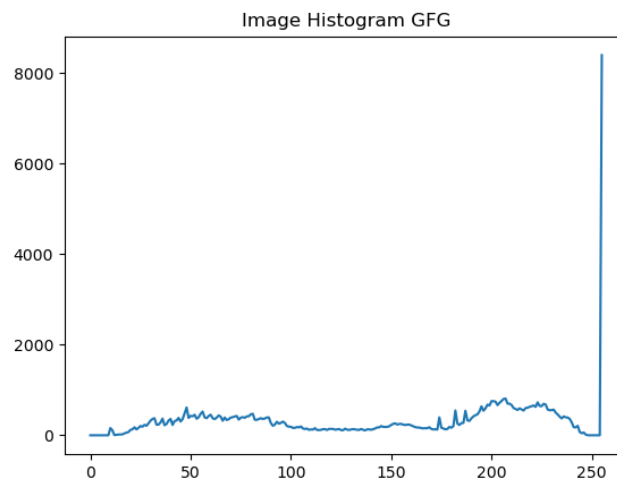


Primera parte:

Segunda parte:



Histograma para img1_tp



Histograma para img2_tp

Conclusiones: tenemos los mismos tipos de grises por cada imagen y la misma cantidad de tipos de grises por cada foto, por ende cada foto es igual. Un algoritmo de clasificación sería inútil en este caso si solo nos basamos en un histograma con diferentes números de bins ya que no nos provee de información espacial de cómo están distribuidos los píxeles que sería la diferenciación en este caso para saber cuándo una imagen es una flor y cuándo no.

Se pueden generar máscaras basadas en segmentos de imagen e ir analizando el valor promedio de los píxeles en zonas de interés. Supongamos que queremos saber si una

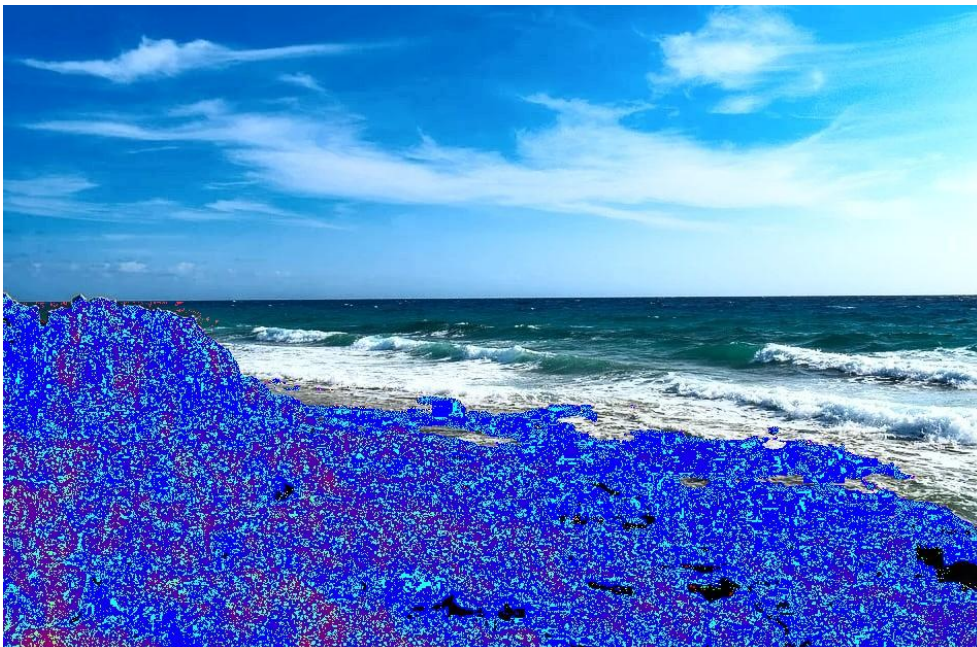
imagen es un paisaje o no, esperaríamos encontrar píxeles que comprendan ciertos rangos para valores RGB y que cubran la mayoría entre las diferentes máscaras generadas para mar, rocas y cielo, sería un aproximación muy naive; veamos el siguiente ejemplo:



Mascara para rocas :

color_lower_bound = (0,0,10)

color_upper_bound = (50,255,255)



Mascara para océano:

color_lower_bound = (50,50,50))

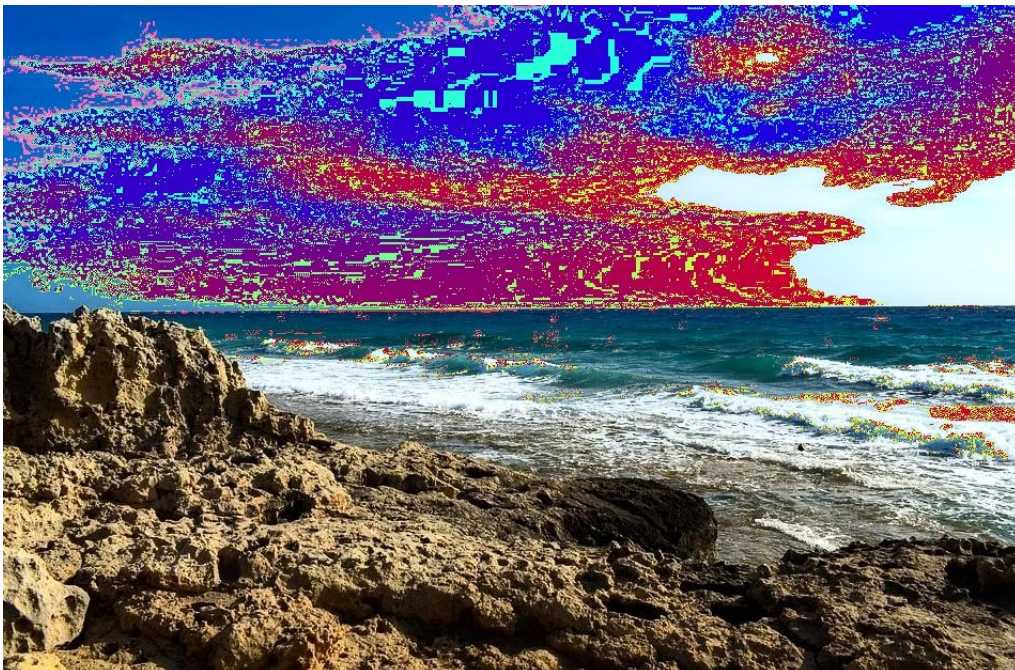
color_upper_bound = (255,255,150)



Mascara para cielo:

color_lower_bound = (50,50,200)

color_upper_bound = (255,255,255)



Esperarías que para imágenes con elementos como rocas, océano y cielo los rangos definidos de `color_lower_bound` y `color_upper_bound` abarcaran la mayoría de la imagen. Dándole un plus sería verificar que la máscara creada de cielo se encuentre arriba de la máscara de océano. Serían validaciones futuras y que serían más fuertes que solo basarse en los histogramas que no albergan información espacial.

Fallas posibles en White Patch, sería una imagen totalmente negra ya que tendríamos una división por cero, pero para nuestro caso eso fue evitado por la función **`div_by_zero`** que devuelve cero si el máximo valor del canal sobre el que estamos trabajando es cero.