



Universidad Nacional de Colombia

ALGEBRA LINEAL

PROYECTO FINAL

Oscar Iván García Montañez
Estudiante de Estadística

1 Introducción

El desarrollo de la tecnología durante los últimos diez años ha permitido algo que en el pasado era impensado, y es que cada persona a nivel mundial, lograrse tener un dispositivo personalizado. En efecto, una de las utilidades más importantes de esta situación, tiene que ver puntualmente con la posibilidad de tener acceso a una cámara y en su defecto, a convertirse indirectamente en un fotógrafo. En consecuencia, el procesamiento de imágenes se ha convertido en una necesidad, puesto que cada vez más son las imágenes que se producen y que requieren muchas veces en su defecto ser manipuladas.

El uso de matrices a través del algebra lineal ha permitido el desarrollo de operaciones matemáticas aplicadas a una imagen, con el objetivo de mejorarlas o en su defecto, lograr gestionarlas. En consecuencia, las imágenes pueden ser representadas como una matriz de píxeles, que en su defecto, permitirán modificar los atributos de la imagen, lo cual evidentemente resulta ser una aplicación muy importante para la actualidad.

2 Planteamiento del problema

El presente proyecto pretende dar respuesta a uno de los problemas más simples en la gestión de imágenes en el lenguaje Python. Fundamentados bajo el concepto de Matrices, el elemento más importante desde el Algebra Lineal, el problema que se pretende solucionar es la manipulación de imágenes hacia la escala de grises, todo enmarcado desde una metodología de Investigación Documental en el repositorio GitHub.

3 Metodología

El presente proyecto investigativo pretende señalar los algoritmos más importantes para la manipulación del color hacia una escala de grises en el procesamiento de imágenes desde una metodología de investigación documental. En primer lugar, se realizó un arqueológico de fuentes utilizando el repositorio GitHub para la selección de los contenidos. Una vez, se realizó la selección, se procedió a la revisión del mismo, descartando el material poco útil. Inmediatamente después de que se comparó el material, se interpretaron los atributos más importantes de los algoritmos para así establecer un producto con muy buenas características desde una perspectiva cualitativa.

4 Resumen

El usuario MonyDCristian en GitHub señala algunos aspectos importantes en el procesamiento de imágenes a partir de matrices. En efecto, las librerías matplotlib y numpy son las más utilizadas; la primera, contiene funciones que permiten la generación de gráficos a partir de datos contenidos en listas o arrays (tipo de lista) y la segunda, contiene funciones matemáticas que sirven para operar con vectores y/o matrices. Para los propósitos de este trabajo, nos fundamentaremos en la librería numpy, pues es la más utilizada para la gestión de imágenes desde el concepto de matrices.

En primer lugar, es imperativo descargar la librería ejecutando en siguiente código en la terminal:

```
pip3 install numpy
```

5 Solución

Es imperativo señalar en este punto que la solución del problema demanda la transformación de la imagen a una matriz de píxeles, y es puntualmente la librería numpy la que nos permite realizar esta tarea.

De acuerdo con la información postada en este repositorio, para realizar esta tarea es imperativo tener presente que la combinación de los tres colores RGB resulta en blanco, así como la ausencia de los mismos resulta en negro. En consecuencia, el promedio de los tres resulta en la escala de grises.

$(R + G + B)/3 = \text{Intensidad en escala de grises}$

No obstante, una búsqueda precisa de la escala de grises en los colores RGB resultarán en la siguiente ecuación (Esta información también es referenciada por nuestro usuario MonyDCristian):

```
intensidad_gris = R * 0.299 + G * 0.587 + B * 0.114
```

En ese orden de ideas, el usuario sugiere el siguiente código para la modificación de cualquier imagen:

```
def escala_de_grises(ruta):  
    #Abrimos la Imagen  
    im = Image.open(ruta)  
    #Obtenemos sus dimensiones  
    x = im.size[0]  
    y = im.size[1]
```

```

#Creamos una nueva imagen con las dimensiones de la imagen anterior
im2 = Image.new('RGB', (x, y))

lista_j = []
while len(lista_j) < y:
    lista_i = []
    while len(lista_i) < x:
        #Obtenemos el valor RGB de cada pixel
        r, g, b = im.getpixel((len(lista_i),len(lista_j)))
        #Obtenemos su equivalente en la escala de gris
        p = (r * 0.299 + g * 0.587 + b * 0.114)
        #Ese valor lo convertimos a entero
        gris = int(p)
        #En la nueva imagen en la posición i, j agregamos el nuevo color
        lista_i.append(gris)
    lista_j.append(lista_i)
imagen = np.array(lista_j)

return imagen

char_w_and_b = escala_de_grises("charmander.png")
io.imsave("charmander_white_and_black.png", char_w_and_b)
mostrar_imagen(char_w_and_b, 255)

```

En primer lugar, es necesario abrir la ruta de la imagen e instanciarla en un objeto al que llamaremos im. Una vez tengamos la imagen ya en una matriz, podemos obtener sus dimensiones y guardarlas en una variable x y y. Luego, nos es necesario crear una nueva imagen con las dimensiones anteriores, la cual denominaremos im2. Ahora viene lo interesante, el usuario MonyDCristian, nos muestra cómo a través de dos recorridos utilizando un while, es posible obtener el valor RGB de cada uno de los píxeles guardados en nuestra matriz para luego así, obtener su equivalente en escala de grises y ubicar el nuevo color en la posición i,j.

6 Bibliografía

MonkeyDCristian. 2019. Procesamiento de imágenes.
https://github.com/MonkyDCristian/Procesamiento_de_imagenes/edit/master/FPI.ipynb.(2022).