Práctica

Realiza los siguientes ejercicios.

1. Instalación de OpenCV

1. Utilizando pip, instala OpenCV y Numpy.

```
pip install opencv-Python
pip install numpy
```

2. Prueba que se instaló correctamente.

```
Import cv2
Cv2.__version__
```

2. Leer y mostrar imágenes

1. Importa la biblioteca de funciones OpenCV:

```
import cv2
```

2. Descarga de internet y guarda una imagen en la misma carpeta que tu script. Utiliza la función cv2.imread() para leer la imagen que acabas de descargar, ejemplo:

```
img = cv2.imread('lena.jpg',1)
```

3. Prueba cambiar el canal de colores de la imagen cambiando el segundo parámetro de la función cv.imread(). Ejemplo:

```
img = cv2.imread('lena.jpg',1)
```

   - mayor a 0 (Red, Green, Blue)
   - igual a 0 (Escala de grises)
   - menor a 0 (alpha)

4. Despliega la imagen que acabas de leer en una ventana. Esta ventana se debe cerrar al presionar la tecla [ESC] o el botón de cerrar. Ejemplo:

```
cv2.imshow('window image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

5. Utilizando la función cv2.imwrite('nombre.jpg'), guarda una copia de la imagen en escala de grises, otra en los 3 canales (RGB) y otra en canal alpha.

3. Dibujar figuras sobre imágenes

1. Importa la biblioteca de funciones OpenCV
2. Descarga una imagen y utiliza la función cv2.imread() para leer la imagen.
3. Dibuja una línea recta sobre la imagen, la función pedirá los siguientes parámetros:
   - La imagen sobre la que estás dibujando.
   - Las coordenadas donde inicia la línea.
   - Las coordenadas donde finaliza la línea.
   - El color de la línea en formato (Blue, Green, Red). Apóyate con Google y busca RGB color picker".
   - El grosor de la línea.

```
img = cv2.line(img, (0,0), (255,255), (0,0,255), 5)
```

4. Dibuja una línea con flecha, un rectángulo y un círculo. Para ello utiliza las siguientes funciones. Si tienes dudas consulta la documentación de OpenCV.

```
img = cv2.arrowedLine(...)

img = cv2.rectangle(...)

img = cv2.circle(...)
```

5. Despliega la imagen en una ventana o guardala con la función cv2.imwrite().

4. Transformación de la intensidad

Utilizando las fórmulas vistas en clase, **crea una función** para cada transformación. Tendrás que definir el nombre y las variables de la función.

4.1. Transformación logarítmica

1. Importa la biblioteca de funciones OpenCV y Numpy.
2. Descarga una imagen y utiliza la función cv2.imread() para leer la imagen.
3. La fórmula en python es la siguiente:

```
c = 255/(np.log(1 + np.max(img)))
transformacion = c * np.log(1 + img)
transformacion = np.array(transformacion, dtype = np.uint8)
```

4. Dentro del script explica a qué se refiere cada línea de la fórmula en python, utiliza #Comentarios.
5. Con la función puText() dibuja la fórmula utilizada sobre la imagen.
6. Despliega la imagen en una ventana o guardala con la función cv2.imwrite().

4.2. Transformación ley de potencia (gamma)

1. Importa la biblioteca de funciones OpenCV y Numpy.
2. Descarga una imagen y utiliza la función cv2.imread() para leer la imagen.
3. La fórmula en python es la siguiente:

```
gamma = 2.2
transformacion = np.array(
                 255*(img / 255) ** gamma, dtype = '\uint8'
                 )
```

4. Dentro del script explica a qué se refiere cada línea de la fórmula en python, utiliza #Comentarios.
5. Con la función puText() dibuja la fórmula utilizada sobre la imagen.
6. Despliega la imagen en una ventana o guardala con la función cv2.imwrite().

4.3. Transformación *piece-wise linear*

1. Importa la biblioteca de funciones OpenCV y Numpy.
2. Descarga una imagen y utiliza la función cv2.imread() para leer la imagen.
3. La fórmula en python es la siguiente:

```
if (0 <= pix and pix <= r1):
return (s1 / r1)*pix
elif (r1 < pix and pix <= r2):
return ((s2 - s1)/(r2 - r1)) * (pix - r1) + s1
else:
return ((255 - s2)/(255 - r2)) * (pix - r2) + s2
```

4. Dentro del script explica a qué se refiere cada línea de la fórmula en python, utiliza #Comentarios.
5. Con la función puText() dibuja la fórmula utilizada sobre la imagen.
6. Despliega la imagen en una ventana o guardala con la función cv2.imwrite().