

Práctica 0

Blanca Cano Camarero

Esta memoria contiene la resolución de ejercicios de la práctica 0.

Ejercicio 1

Escribir una función que lea el fichero de una imagen y permita mostrarla tanto en grises como en color (`im=leeimagen(filename, flagColor)`). `flagColor` es la variable que determina si la imagen se muestra en escala de grises o en color.

Explicación de la solución

La solución se implementa con la función `leeImagen(filename, flagColor)`: Usaremos la función propia de `cv2` `imread`. `imread(path, flag)`, que muestra el `path` de la imagen y `flag` el método en que la variable será leída.

Los flags que nos van a ser útiles son:

- `cv2.IMREAD_COLOR` para color equivale al entero 1.
- `cv2.IMREAD_GRAYSCALE` Modo de escala de grises, equivale al entero 0.

Recursos utilizados (fecha de consulta 24/09/21): [Geekforgeek](#)

Prueba de que la solución parece estar bien hecho son las figuras 1a y 1b

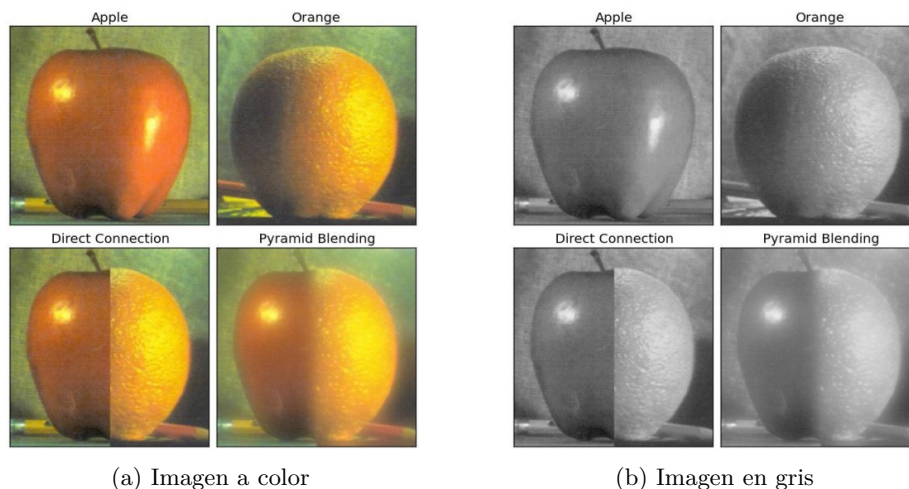


Figure 1: Imágenes mostradas con `imread` (Notebook)

Ejercicio 2

Escribir una función que permita visualizar una matriz de números reales cualquiera/arbitraria, tanto monobanda como tribanda (`pintaI(im)`). Para ello se deberá escalar el rango de cada banda al intervalo $[0,1]$ sin pérdida de información.

Explicación

Para conseguir el objetivo tendremos que resolver dos problemas principalmente:
1. Conseguir un reescalado. 2. Mostrarlo.

Soluciones

1. Para conseguir el reescalado aplicaremos un homeomorfismo del intervalo $[a, b]$ a $[0, 1]$.

Como no se nos ha proporcionado información adicional sobre el ejercicio supondremos que a es el valor mínimo de las entradas de la matriz y b el valor máximo.

Distinguimos las siguientes casuísticas:

- Para $a = b$, puesto que no nos indica nada, tomamos el criterio arbitrario de poner a cero todas las entradas.
- En caso de que sean diferentes: Para todo $p \in \mathbb{R}^d$ con $d \in \{1, 3\}$ un pixel de la matriz (1 si es gris, 3 si es de colores). El homeomorfismo de reescalado R viene dado por:

$$R(p) = \frac{p - a}{b - a}$$

2. Para la muestra. Vamos a mostrarlo con la biblioteca de matplotlib. Cabe destacar que para matplotlib es necesario invertir los canales de colores.

Otra opción (implementada en `pintaI2`) la función `cv2.imshow` no es posible usarla en colab, luego hemos usado una alternativa proporcionada por google Hemos usado este ejemplo [Link ejemplo colab](#)

Un ejemplo de ejecución sería: 1. Matriz de colores aleatoria 2a. 2. Matriz solo de unos 2b. 3. Matriz en blanco y negro 2c.

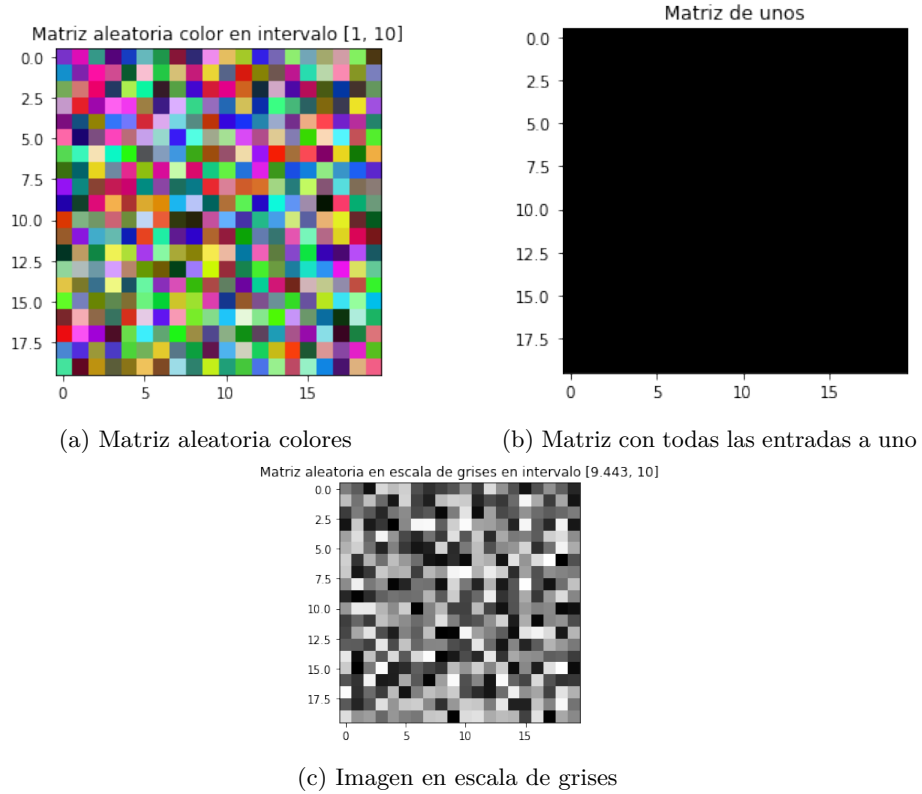


Figure 2: Imágenes mostradas con 'PintaI'

Ejercicio 3

Escribir una función que visualice varias imágenes distintas a la vez (concatenando las imágenes en una última imagen final 1): `pintaMI(vim)`. (`vim` será una secuencia de imágenes) ¿Qué pasa si las imágenes no son todas del mismo tipo?

Descripción

Pretendemos visualizar varias imágenes concatenadas. La descripción general del algoritmo es: 1. Criterio para manejar las que estén en color y las que no. 2. Ajustar tamaño 3. Concatenar.

La soluciones aportadas: 1. Se transforman todas a color gracias a la función `cv2.cvtColor`.

2. Deben de ser del mismo tamaño: hemos optado por escalar la alturas hasta el tamaño de la mayor. Esto se hace por medio de la función de `cv2.resize`.

La documentación seguida ha sido la de la página oficial a día 29 de septiembre [link](#)

3. Concatenamos las sucesivas imágenes.

El resultado de usar la función con diferentes combinaciones de imágenes es:

Como podemos observar en 3c nuestra función admite combinar imágenes en blanco y negro y color gracias a que hemos transformado todas las imágenes a color.

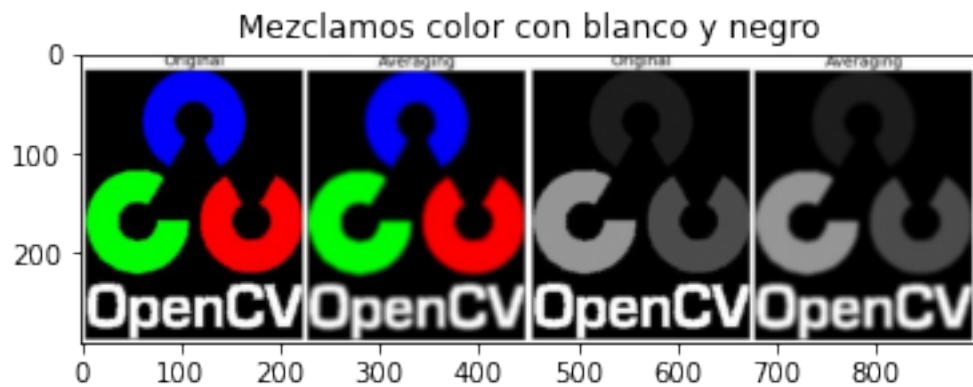
Esto tiene la desventaja de que durante la ejecución consumirá más memoria, este gasto será además innecesario para el caso en que todas las imágenes sean grises. Sin embargo al nivel en que estamos trabajando consideramos esta opción por agilidad en programación.



(a) Todas las imágenes concatenadas



(b) Todas imágenes concatenadas en escala de gris



(c) Combinamos imagenes en blanco y negro

Figure 3: Ejemplos de ejecución ejercicio 3

Ejercicio 4

4. Escribir una función que modifique el color en la imagen de cada uno de los elementos de una lista de coordenadas de píxeles 2 . En concreto, los alumnos deben insertar un cuadrado azul de 100x100 píxeles en el centro de la imagen a modificar.

Descripción general del procedimiento

Dado que las imágenes no dejan de ser matrices, cambiaremos el color de las entradas pedidas. La función `CambiaPíxeles` permite cambiar el color de píxeles de la lista de manera genérica. `InsertaCuadradoAzul` es la función que resuelve nuestro problema.

Detalles que se han tenido en cuenta:

1. El color se encuentra en BGR en vez de RGB y va de 0 a 255 (en el caso de estar en modo entero).

Cabe además destacar que se ha tenido en cuenta que la imagen puede estar en escala de grises, en ese caso se le asociará el gris correspondiente al azul (que no es más que la media).

2. Los ejes están girados 90° en sentido horario con centro en el origen con respecto al sistema de referencia usual en matemáticas.

Por lo demás, las operaciones para calcular el centros son tan simples como dado un lado (ya sea altura o anchura), restarle los 100 píxeles del cuadrado azul y dividir entre dos. El píxel resultante y los 100 siguientes serán un lado del cuadrado.

Ejemplos

- Imagen a color a la que se le ha añadido cuadrado azul 4a.
- Imagen en escala de grises que ha sido transformada a color y por lo que se le ha añadido cuadrado azul 4b.
- Imagen, monocromática a la que se le ha añadido el cuadrado 4c. Por ser monocromática el cuadrado respeta la escala de grises.

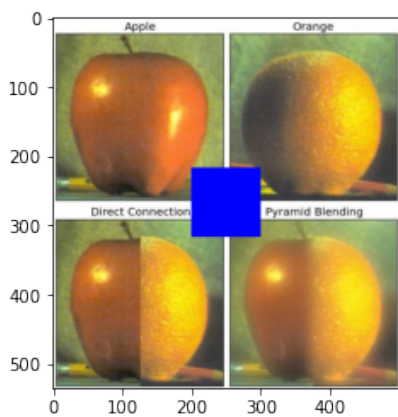
Ejercicio 5

5. Una función que sea capaz de representar varias imágenes con sus títulos en una misma ventana.

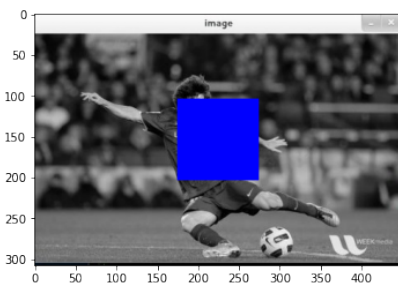
Solución

Puesto que no se especifica se han contemplado los siguientes casos:

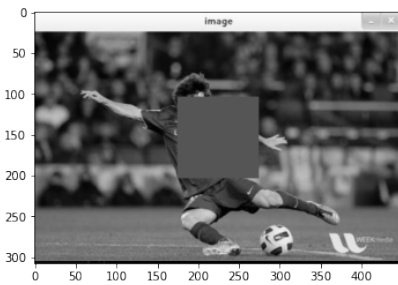
1. Las imágenes se quieren separadas. Para ello nos hemos basado en estos ejemplos . Un ejemplo del resultado en la imagen 5a.
2. Se ha contemplado la opción de que se quieran la imágenes concatenadas. esto se resuelve en la función `multiplesImagenConcatenadasConTitulo(v_image, v_titulo)`. Un ejemplo del resultado en la imagen 5b.



(a) Imagen a color.

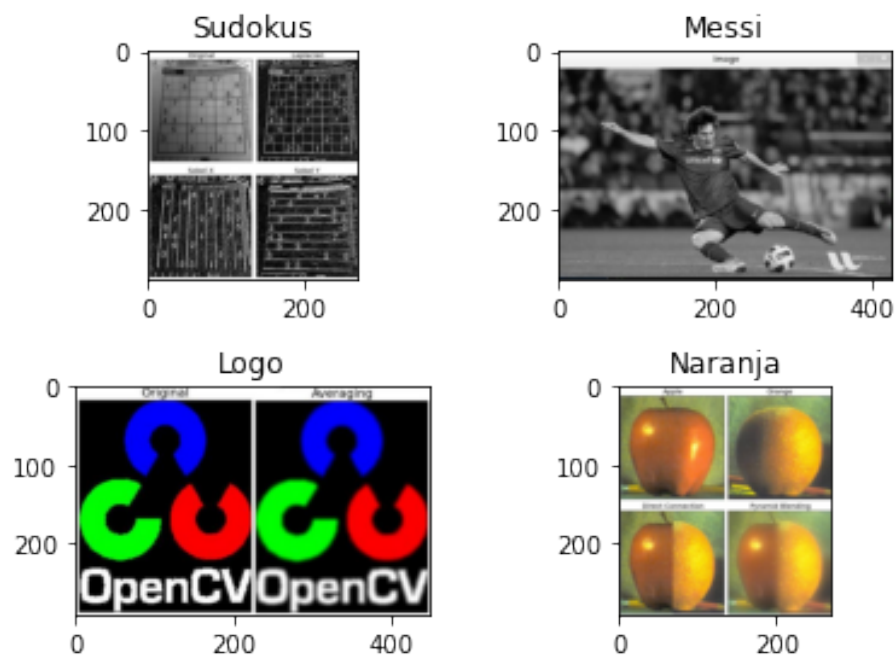


(b) Combinamos imagenes en blanco y negro

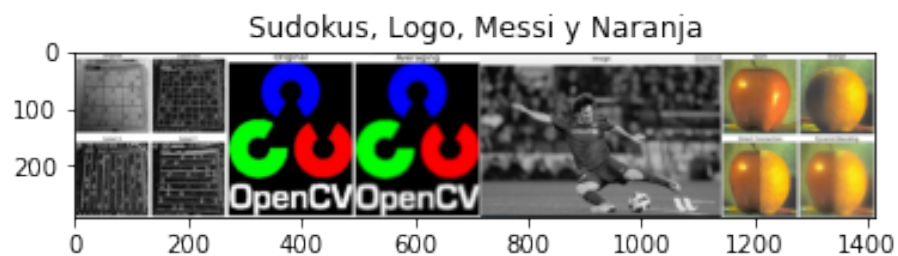


(c) Imagen en escala de grises.

Figure 4: Ejemplos del ejercicio 4, añadir cuadrado azul en el centro



(a) Imágenes juntas en la misma imagen



(b) Se concatenan las imágenes y se le añaden los respectivos títulos

Figure 5: Ejemplos del ejercicio 5