Operaciones básicas entre imágenes (Suma, Resta, Multiplicación y División)

López Sánchez Carlos Alberto
Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Zacatenco
Ciudad de México, México
clopezs1100@aulumno.ipn.mx

Resumen – Con la ayuda de un lenguaje de alto nivel se realizaron las operaciones aritméticas básicas (suma, resta, Multiplicación y División) entre dos imágenes haciendo uso de los operadores y además también de métodos que contiene las librerías que se usaron para este trabajo las cuales son openCV y Numpy esto con el fin de comprobar los resultados obtenidos y poder determinar como se programaron esos métodos.

I. INTRODUCCIÓN

En el presente trabajo se hace uso de imágenes digitales, las cuales se puede determinar que son de dos dimensiones al tener un alto y ancho, basándonos en estos datos damos por hecho que una imagen digital es una matriz en la que cada posición va a guardar información de la imagen que se esté ocupando, ya que existen diferentes tipos de imágenes y cada tipo fue creado para diferentes propósitos, teniendo en cuenta que para este trabajo se ocuparon imágenes tipo “jpg” y está en un formato a color, los datos que se almacenan en la matriz ban a ser los valores correspondientes a los pixeles y esto es en el formato RGB (Red, Green, Blue) algunas pueden poseer un cuarto dato que es el A y este va aunado a la transparencia de la imagen.

Para realizar las operaciones aritméticas básicas con imágenes digitales se hace uso el software Colaboratory, o "Colab", es un servicio de Google que permite a cualquier usuario con una cuenta de Gmail poder escribir y ejecutar código del lenguaje de alto nivel Python en el navegador. Colab es un servicio de cuaderno alojado de Jupyter que no requiere configuración previa para su uso y que ofrece acceso sin coste adicional a recursos informáticos, como GPUs, si se requiere más potencia o extensiones de lo que ofrece la versión gratuita se puede hacer uso de los planes que ofrece Google como la versión Colab Pro o Colab Pro +.

II. DESARROLLO DE CONTENIDOS

A. Importación de librerías.

Para le realización de esta practica se hace uso de 3 librerías

OpenCV es una librería para visión artificial para realizar operaciones simples con imágenes como: Abrir y guardar imágenes. Dibujar formas simples en imágenes. Escribir en imágenes. de código abierto y que está disponible para muchos lenguajes de programación.

Numpy es una librería en la que se define un tipo de dato, crean vectores y matrices grandes multidimensionales, junto con algunas funcionalidades básicas para trabajar con ellas matemáticas de alto nivel para operar con ellas.

cv2_imshow debido a que el método imshow es incompatible con Colab se debe implementar esta librería que funciona como un parche para poder visualizar las imágenes en el cuaderno en el que se esté trabajando. La incompatibilidad de este método se debe a que al ser ejecutado hace uso de un servidor que no esta disponible en el navegador

B. Leer y guardar imágenes.

En esta parte del trabajo se leen y se guardan dos imágenes ***Figura 1*** *y la* ***Figura 2****, en variables para poder hacer uso de estas y gracias a las librerías que se importaron acceder a sus métodos para poder* ver que las imágenes son de las mismas dimensiones y así realizar las operaciones que se solicitaron.

Figura 1 *con dimensiones (320, 320, 3).*

Figura 2 *con dimensiones (320, 320, 3).*

C. Conversión de datos.

Conversión de datos: Para las operaciones que se realizan es necesario convertir las imágenes de su tipo original (uint8) a otro tipo de dato que facilite su procesamiento ya que si se intenta realizar con los datos de la imagen original los resultados que se obtienen no serán los esperados.

El tipo de dato matriz, que contendrá una imagen puede ser de varios tipos (según el tipo de dato de cada píxel), En este caso vamos a transformar las *Figuras 1 y 2* a un tipo de formato en el cual se pueda realizar las operaciones aritméticas básicas de uint8 a tipo float32.

D. Suma de Imágenes

Ya que se realizó el cambio de dato de las imágenes a tipo float32 vamos a sumar sus valores, en la ecuación (1) se visualiza como se debe realizar esta operación.

$$Suma = Figura\ 1 + Figura\ 2 \qquad (1)$$

Una vez que se realiza la suma se cambia nuevamente el tipo de dato uint8 para posteriormente mostrar el resultado ***Figura 3.***

Figura 3 *Suma de imágenes.*

Para poder cambiar el brillo de esta imagen se debe dividir la ecuación 1 entre un numero constante en esta ocasión será el número 2 y como resultado nos da la ecuación 2 y la imagen resultante es la ***Figura 4***.

$$Suma2 = \frac{Figura\ 1 + Figura\ 2}{2} \qquad (2)$$

Figura 4 *Imagen de la suma más obscura.*

Para poder comparar los resultados se hace uso del método de la suma de OpenCV (cv2.addWeigted()), el cual lleva cinco parámetros las dos imágenes correspondientes a las *Figuras 1 y 2*, y pesos que son representados por Alpha y beta y gamma, estos son la transparencia que va a tener la imagen asociada para este caso el resultado de la ***Figura 5*** *el método se usa de la siguiente manera*
cv2.addWeigted(***Figura 1, 0.5, Figura 2, 0.5, 0)***

Figura 5 *usando el método cv2.addWeigted*

E. Resta de Imágenes

Haciendo uso de los mismos datos que se usaron en la suma ahora procedemos a realizar una resta ecuación 3, lo que nos produce el resultado de la ***Figura 6***

$$Resta = Figura\ 1 - Figura\ 2 \qquad (3)$$

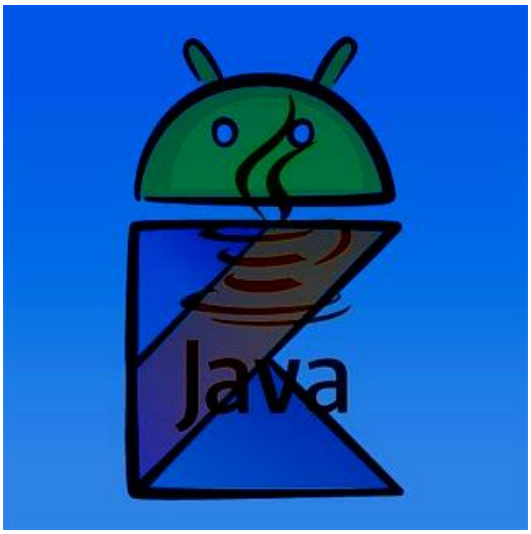

Figura 6 *Resta de Figura 1 - Figura 2*

Como bien sabemos la resta no es una operación conmutativa así que si restamos como la ecuación 4 obtenemos la ***Figura 11***

$$Resta2 = Figura\ 2 - Figura\ 1 \qquad (4)$$

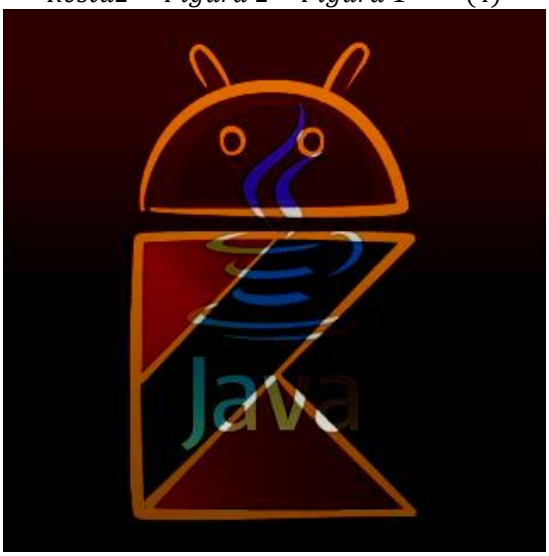

Figura 7 *Resta de Figura 2 - Figura 1*

Si se quiere realizar la comparación de estos resultados con un método de openCV se tiene que hacer uso de cv2.subtract() este método solo hace uso de dos parámetros los cuales son las imágenes que se van a restar como se puede observar cv2.subtract(*Figura 1, Figura 2) o bien cv2.subtract(Figura 2, Figura 1)*

F. Multiplicación de Imágenes

Para poder realizar la multiplicación se ocuparon los datos originales de las imágenes sin transformarlos a float32, para esto se requiere de la ecuación 5 y así obtenemos la ***Figura 8***

$$\text{Multiplicación} = \frac{Figura\ 1 * Figura\ 2}{255} \qquad (5)$$

Figura 8 *Multiplicación de Imágenes*

G. División de Imágenes

De la misma manera que en la multiplicación se ocupan los datos originales de las imágenes, para realizar esta operación hacemos uso de la ecuación 6 para tener como resultado la **Figura 9**

$$\text{División} = \frac{Figura\ 1 / Figura\ 2}{255} * 255 \qquad (6)$$

Figura 9 *División de Figura 1 / Figura 2*

$$\text{División} = \frac{Figura\ 2 / Figura\ 1}{255} * 255 \qquad (7)$$

Figura 9 *División de Figura 2 / Figura 1*

Como la división tampoco es una operación conmutativa se realiza la operación 7

III. CONCLUSIÓN

Es imprescindible que para realizar las operaciones que se mostraron las imágenes deben tener las mismas dimensiones y también que se debe realizar el cambio de dato para que se puedan tener los resultados esperados.

Al querer ocupar los métodos que vienen por defecto con las librerías investigar que argumentos deben tener y para qué sirven.

Se debe tener en cuenta que operaciones son conmutativas.

La practica se puede encontrar en GitHub https://github.com/CarlosAlbertoLS/Practica-Vision