Image Addition

import cv2 as cv  
import numpy as np  
x = np.uint8([250])  
y = np.uint8([10])  
print( cv.add(x,y) ) # 250+10 = 260 => 255  
print( x+y ) # 250+10 = 260 % 256 = 4

[[255]]  
[4]

Mezcla de imágenes Esto también es una adición de imágenes, pero se les da diferentes pesos para dar una sensación de fusión o transparencia. Aquí tomé dos imágenes para combinarlas. A la primera imagen se le asigna un peso de 0,7 y a la segunda imagen se le asigna 0,3. cv.addWeighted() aplica la siguiente ecuación a la imagen:

img1 = cv.imread('resources/ml.png')  
img2 = cv.imread('resources/opencv-logo.png')  
print("img1 shape:", img1.shape)  
print("img2 shape:", img2.shape)  
  
assert img1 is not None, "file could not be read, check with os.path.exists()"  
assert img2 is not None, "file could not be read, check with os.path.exists()"  
dst = cv.addWeighted(img1,0.7,img2,0.3,0)  
cv.imshow('dst',dst)  
cv.waitKey(0)  
cv.destroyAllWindows()

Operaciones bit a bit Esto incluye las operaciones bit a bit AND, OR, NOT y XOR. Serán muy útiles al extraer cualquier parte de la imagen (como veremos en los próximos capítulos), definir y trabajar con ROI no rectangulares, etc. A continuación veremos un ejemplo de cómo cambiar una región particular de una imagen. .

Quiero poner el logo de OpenCV encima de una imagen. Si agrego dos imágenes, cambiará el color. Si los difumino, obtengo un efecto transparente. Pero quiero que sea opaco. Si fuera una región rectangular, podría usar el ROI como hicimos en el capítulo anterior. Pero el logotipo de OpenCV no tiene forma rectangular. Entonces puedes hacerlo con operaciones bit a bit como se muestra a continuación

# Load two images  
img1 = cv.imread('resources/messi5.jpg')  
img2 = cv.imread('resources/opencv-logo-white.png')  
assert img1 is not None, "file could not be read, check with os.path.exists()"  
assert img2 is not None, "file could not be read, check with os.path.exists()"  
# I want to put logo on top-left corner, So I create a ROI  
rows,cols,channels = img2.shape  
roi = img1[0:rows, 0:cols]  
# Now create a mask of logo and create its inverse mask also  
img2gray = cv.cvtColor(img2,cv.COLOR\_BGR2GRAY)  
ret, mask = cv.threshold(img2gray, 10, 255, cv.THRESH\_BINARY)  
mask\_inv = cv.bitwise\_not(mask)  
# Now black-out the area of logo in ROI  
img1\_bg = cv.bitwise\_and(roi,roi,mask = mask\_inv)  
# Take only region of logo from logo image.  
img2\_fg = cv.bitwise\_and(img2,img2,mask = mask)  
# Put logo in ROI and modify the main image  
dst = cv.add(img1\_bg,img2\_fg)  
img1[0:rows, 0:cols ] = dst  
cv.imshow('res',img1)  
cv.waitKey(0)  
cv.destroyAllWindows()

Exercise

import cv2  
import os  
import glob  
from threading import Timer

def read\_images(folder\_path):  
 """  
 Read all images from a folder and return a list of image paths.  
 """  
 image\_files = sorted(glob.glob(os.path.join(folder\_path, '\*.jpg'))) # Change the extension if needed  
 return [cv2.imread(img) for img in image\_files]

def blend\_images(img1, img2, alpha):  
 """  
 Blend two images using cv2.addWeighted.  
 """  
 return cv2.addWeighted(img1, alpha, img2, 1 - alpha, 0)

def show\_slideshow(images, delay\_between\_images=2000):  
 """  
 Display a slideshow with smooth transitions between images.  
 """  
 window\_name = 'Slideshow'  
 cv2.namedWindow(window\_name, cv2.WINDOW\_NORMAL)  
  
 current\_index = 0  
  
 def transition():  
 nonlocal current\_index  
 img1 = images[current\_index]  
 current\_index = (current\_index + 1) % len(images)  
 img2 = images[current\_index]  
  
 alpha = 0.0  
 while alpha <= 1.0:  
 blended = blend\_images(img1, img2, alpha)  
 cv2.imshow(window\_name, blended)  
 alpha += 0.01 # Adjust the step size for smoother transitions  
 cv2.waitKey(10)  
  
 timer = Timer(delay\_between\_images / 1000.0, transition)  
 timer.start()  
  
 transition() # Start the initial transition  
  
 cv2.waitKey(0)  
 cv2.destroyAllWindows()

folder\_path = 'resources' # Replace with the path to your image folder  
images = read\_images(folder\_path)  
  
if not images:  
 print("No images found in the specified folder.")  
else:  
 show\_slideshow(images)