## **การดำเนินการเลขคณิตบนรูปภาพ**

เป้าหมาย

* เรียนรู้การดำเนินการเลขคณิตจำนวนมากในภาพเช่นการบวกการลบการดำเนินการแบบบิตเป็นต้น
* คุณจะได้เรียนรู้ฟังก์ชันเหล่านี้: **cv2.add ()** , **cv2.addWeighted ()**ฯลฯ

การเพิ่มรูปภาพ

คุณสามารถเพิ่มภาพสองภาพโดยฟังก์ชั่น OpenCV, cv2.add ()หรือเพียงโดยการดำเนินการ numpy, Res = img1 + img2 ภาพทั้งสองภาพควรมีความลึกและชนิดเดียวกันหรือภาพที่สองก็สามารถเป็นค่าที่เป็นนาม้ได้

บันทึก : มีความแตกต่างระหว่างการเพิ่ม OpenCV และ Numpy นอกจากนี้ OpenCV นอกจากนี้การดำเนินการsaturated operationในขณะที่ Numpy นอกจากนี้ยังมีการดำเนินการแบบโมดูล

ตัวอย่างเช่นพิจารณาตัวอย่างด้านล่าง:

**>>>** x = np.uint8([250])

**>>>** y = np.uint8([10])

**>>> print** cv2.add(x,y) *# 250+10 = 260 => 255*

[[255]]

**>>> print** x+y *# 250+10 = 260 % 256 = 4*

[4]

จะปรากฏให้เห็นเมื่อคุณเพิ่มภาพสองภาพ ฟังก์ชัน OpenCV จะให้ผลลัพธ์ที่ดีขึ้น ฟังก์ชั่น OpenCV ดีกว่าเสมอ

การผสมภาพ

นอกจากนี้ยังมีการเพิ่มรูปภาพ แต่จะให้น้ำหนักที่ต่างกันกับภาพเพื่อให้รู้สึกถึงการผสมหรือโปร่งใส ภาพถูกเพิ่มตามสมการด้านล่าง:

g (x) = (1 - \ alpha) f_ {0} (x) + \ alpha f_ {1} (x)

คุณสามารถเปลี่ยนจากภาพหนึ่งไปยังอีกภาพหนึ่งได้โดยเปลี่ยน\ alphaจาก0 \ rightarrow 1

ที่นี่ฉันเอาภาพสองภาพมารวมกัน ภาพแรกให้น้ำหนัก 0.7 และภาพที่สองจะได้รับ 0.3 cv2.addWeighted ()ใช้สมการต่อไปนี้ในภาพ

dst = \ alpha \ cdot img1 + \ beta \ cdot img2 + \ gamma

ต่อไป\ แกมมานี้เป็นศูนย์

img1 = cv2.imread('ml.png')

img2 = cv2.imread('opencv\_logo.jpg')

dst = cv2.addWeighted(img1,0.7,img2,0.3,0)

cv2.imshow('dst',dst)

cv2.waitKey(0)

cv2.destroyAllWindows()



Bitwise การดำเนินงาน

ซึ่งรวมถึงการดำเนินการบิตและ OR หรือ NOT และ XOR พวกเขาจะมีประโยชน์อย่างมากในขณะที่ดึงส่วนหนึ่งส่วนใดของภาพ (ตามที่เราจะเห็นในบทที่จะมาถึง) การกำหนดและการทำงานกับ ROI ที่ไม่ใช่รูปสี่เหลี่ยมผืนผ้า ฯลฯ ด้านล่างเราจะเห็นตัวอย่างเกี่ยวกับวิธีเปลี่ยนพื้นที่เฉพาะของรูปภาพ

ฉันต้องการใส่โลโก้ OpenCV เหนือภาพ ถ้าเพิ่มภาพสองภาพก็จะเปลี่ยนสี ถ้าฉันผสมผสานฉันจะได้รับผลโปร่งใส แต่ฉันต้องการให้เป็นสีทึบ ถ้าเป็นพื้นที่รูปสี่เหลี่ยมผืนผ้าฉันสามารถใช้ ROI ได้ตามที่เราได้ทำในบทสุดท้าย โลโก้ OpenCV ไม่ใช่รูปสี่เหลี่ยมผืนผ้า ดังนั้นคุณสามารถทำมันด้วยการดำเนินการ Bitwise ดังนี้:

*# Load two images*

img1 = cv2.imread('messi5.jpg')

img2 = cv2.imread('opencv\_logo.png')

*# 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 = cv2.cvtColor(img2,cv2.COLOR\_BGR2GRAY)

ret, mask = cv2.threshold(img2gray, 10, 255, cv2.THRESH\_BINARY)

mask\_inv = cv2.bitwise\_not(mask)

*# Now black-out the area of logo in ROI*

img1\_bg = cv2.bitwise\_and(roi,roi,mask = mask\_inv)

*# Take only region of logo from logo image.*

img2\_fg = cv2.bitwise\_and(img2,img2,mask = mask)

*# Put logo in ROI and modify the main image*

dst = cv2.add(img1\_bg,img2\_fg)

img1[0:rows, 0:cols ] = dst

cv2.imshow('res',img1)

cv2.waitKey(0)

cv2.destroyAllWindows()

ดูผลลัพธ์ด้านล่าง ภาพซ้ายแสดงหน้ากากที่เราสร้างขึ้น ภาพขวาแสดงผลสุดท้าย เพื่อความเข้าใจที่มากขึ้นแสดงภาพทั้งหมดที่เป็นสื่อกลางในรหัสดังกล่าวข้างต้นโดยเฉพาะอย่างยิ่งimg1\_bgและimg2\_fg

