## **เริ่มต้นใช้รูปภาพ**

**เป้าหมาย**

## ที่นี่คุณจะได้เรียนรู้วิธีอ่านภาพวิธีการแสดงภาพและวิธีบันทึกภาพ

## คุณจะได้เรียนรู้ฟังก์ชันเหล่านี้: **cv2.imread ()** , **cv2.imshow ()** , **cv2.imwrite ()**

## และคุณจะได้เรียนรู้วิธีแสดงภาพด้วย **Matplotlib**

**อ่านภาพ**

## ใช้ฟังก์ชัน**cv2.imread ()**เพื่ออ่านภาพ ภาพควรอยู่ในไดเร็กทอรีการทำงานหรือต้องมีพาธภาพเต็มรูปแบบ

## อาร์กิวเมนต์ที่สองคือแฟล็กซึ่งระบุรูปที่ควรอ่าน

## cv2.IMREAD\_COLOR: โหลดภาพสี ความโปร่งใสของภาพจะถูกละเลย เป็นธงเริ่มต้น

## cv2.IMREAD\_GRAYSCALE: โหลดภาพในโหมด grayscale

## cv2.IMREAD\_UNCHANGED: โหลดภาพเป็นช่องรวมทั้งช่องอัลฟา

## **บันทึก** คุณสามารถส่งผ่านจำนวนเต็ม 1, 0 หรือ -1 แทนทั้งสามค่านี้ได้ตามลำดับ

**import** **numpy** **as** **np**

**import** **cv2**

*# Load an color image in grayscale*

img = cv2.imread('messi5.jpg',0)

**แสดงรูปภาพ**

## ใช้ฟังก์ชัน **cv2.imshow ()** เพื่อแสดงภาพในหน้าต่าง หน้าต่างจะพอดีกับขนาดภาพโดยอัตโนมัติ

## อาร์กิวเมนต์แรกคือชื่อหน้าต่างซึ่งเป็นสตริง อาร์กิวเมนต์ที่สองคือภาพลักษณ์ของเรา คุณสามารถสร้างหน้าต่างได้มากเท่าที่คุณต้องการ แต่มีชื่อหน้าต่างต่างกัน

cv2.imshow('image',img)

cv2.waitKey(0)

cv2.destroyAllWindows()



**cv2.waitKey ()** เป็นฟังก์ชันการผูกคีย์บอร์ด อาร์กิวเมนต์ของมันคือเวลาในหน่วยมิลลิวินาที ฟังก์ชันรอสำหรับมิลลิวินาทีที่ระบุสำหรับเหตุการณ์แป้นพิมพ์ใด ๆ หากคุณกดปุ่มใด ๆ ในขณะนั้นโปรแกรมจะทำงานต่อ ถ้า **0** ถูกส่งไปจะรออย่างไม่มีกำหนดสำหรับจังหวะที่สำคัญ นอกจากนี้ยังสามารถตั้งค่าให้ตรวจจับจังหวะที่สำคัญได้เช่นถ้ากดปุ่มaเป็นต้นซึ่งเราจะกล่าวถึงด้านล่าง

**cv2.destroyAllWindows ()** เพียงทำลายหน้าต่างทั้งหมดที่เราสร้างขึ้น ถ้าคุณต้องการทำลายหน้าต่างใด ๆ ให้ใช้ฟังก์ชัน **cv2.destroyWindow ()** ที่คุณส่งชื่อหน้าต่างที่ถูกต้องเป็นอาร์กิวเมนต์

มีกรณีพิเศษที่คุณสามารถสร้างหน้าต่างและโหลดภาพลงในภายหลังได้ ในกรณีนี้คุณสามารถระบุว่าหน้าต่างสามารถปรับขนาดได้หรือไม่ มันจะทำกับฟังก์ชั่น **cv2.namedWindow ()** โดยค่าเริ่มต้นธงเป็น**cv2.WINDOW\_AUTOSIZE** แต่ถ้าคุณระบุ **flag** เป็น **cv2.WINDOW\_NORMAL** คุณสามารถปรับขนาดหน้าต่างได้ จะมีประโยชน์เมื่อภาพมีขนาดใหญ่เกินไปและเพิ่มแถบแทร็กลงในหน้าต่าง

cv2.namedWindow('image', cv2.WINDOW\_NORMAL)

cv2.imshow('image',img)

cv2.waitKey(0)

cv2.destroyAllWindows()

เขียนภาพ

## ใช้ฟังก์ชัน **cv2.imwrite ()** เพื่อบันทึกภาพ

## อาร์กิวเมนต์แรกคือชื่อไฟล์อาร์กิวเมนต์ที่สองคือภาพที่คุณต้องการบันทึก

CV2 imwrite ( 'messigray.png' , img )

ซึ่งจะบันทึกรูปภาพในรูปแบบ PNG ในไดเร็กทอรีการทำงาน

ทำงานร่วมกับ IF

ด้านล่างโปรแกรมโหลดภาพในโทนสีเทาแสดงมันบันทึกภาพถ้าคุณกด **'s'** และออกหรือเพียงแค่ออกโดยไม่บันทึกถ้าคุณกด **ESC** สำคัญ

**import** **numpy** **as** **np**

**import** **cv2**

img = cv2.imread('messi5.jpg',0)

cv2.imshow('image',img)

k = cv2.waitKey(0)

**if** k == 27: *# wait for ESC key to exit*

cv2.destroyAllWindows()

**elif** k == ord('s'): *# wait for 's' key to save and exit*

cv2.imwrite('messigray.png',img)

cv2.destroyAllWindows()

ใช้ Matplotlib

Matplotlib เป็นพล็อตห้องสมุดสำหรับ Python ซึ่งจะช่วยให้คุณวางแผนการต่างๆได้หลากหลาย คุณจะเห็นพวกเขาในบทความที่จะมาถึง ที่นี่คุณจะได้เรียนรู้วิธีการแสดงภาพด้วย Matplotlib คุณสามารถซูมภาพบันทึก ฯลฯ โดยใช้ Matplotlib

**import** **numpy** **as** **np**

**import** **cv2**

**from** **matplotlib** **import** pyplot **as** plt

img = cv2.imread('messi5.jpg',0)

plt.imshow(img, cmap = 'gray', interpolation = 'bicubic')

plt.xticks([]), plt.yticks([]) *# to hide tick values on X and Y axis*

plt.show()



**การเตือน** ภาพสีที่โหลดโดย OpenCV อยู่ในโหมด BGR แต่ Matplotlib จะแสดงในโหมด RGB ภาพสีจะไม่แสดงอย่างถูกต้องใน Matplotlib ถ้าอ่านภาพด้วย OpenCV โปรดดูการออกกำลังกายเพื่อดูรายละเอียดเพิ่มเติม