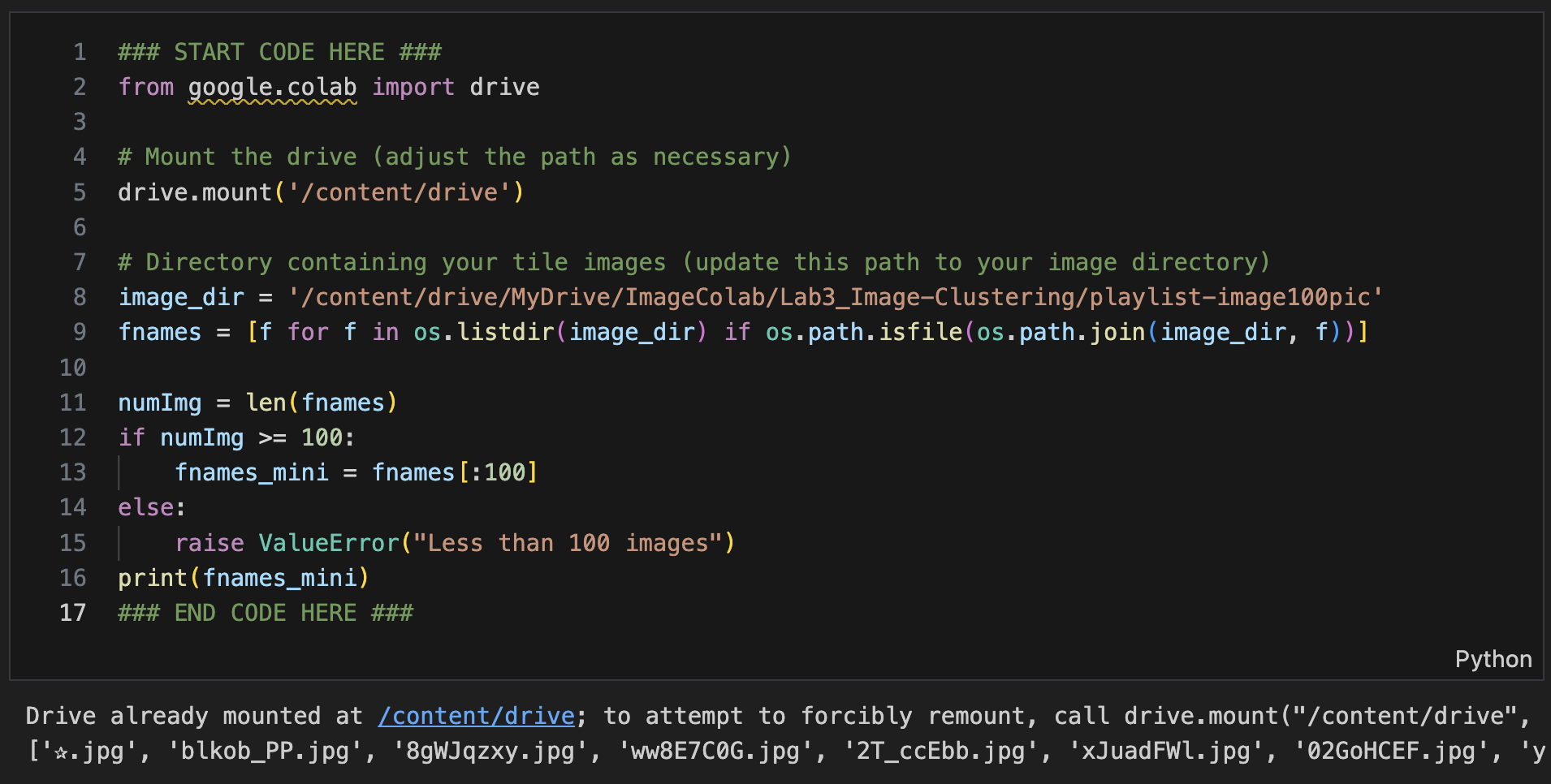
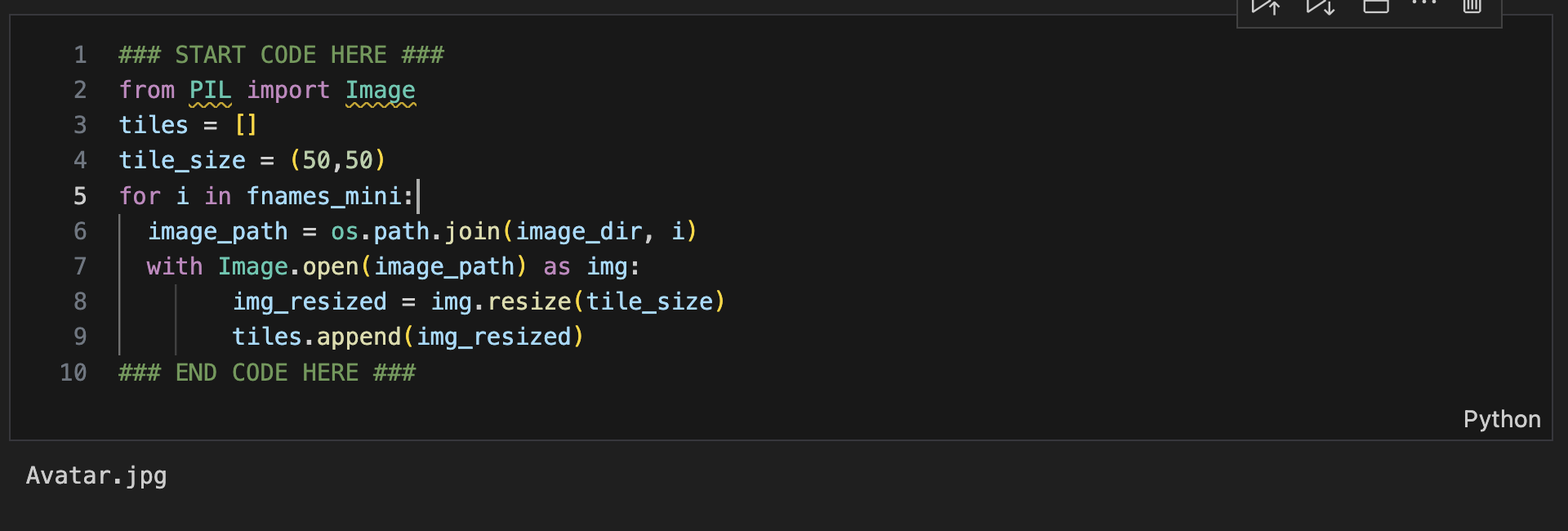
**Lab3 Photo Mosaic KNN**

​Preparation



* กำหนด dataset directory แล้วใช้ function os.listdir() ในการ list ไฟล์ทั้งหมดใน directory นั้น เพื่อเอาชื่อไฟล์ของแต่ละรูปเก็บไปใน fnames
* ให้ fnames\_mini เก็บ 100 element แรกของ fnames



* กำหนด tile size เป็น 50 x 50 pixel แล้วนำรูปแต่ละรูปมา resize แล้วเก็บไปใน list tiles

A screenshot of a computer

Description automatically generated

* สุ่มรูปมาจาก tiles จำนวน 5 รูป แล้วเอามา plot

A screen shot of a computer program

Description automatically generated

* นำทั้ง 5. รูปมาหาสีเฉลี่ย โดยใช้ฟังก์ชัน np.average() แล้วเก็บใส่ list colors โดยผลลัพธ์จะออกมาเป็นค่าสีของแต่ละ channel RGB

A screen shot of a computer program

Description automatically generated

A screen shot of a video

Description automatically generated

* นำสีเฉลี่ยจากข้อที่แล้วมา normalize เพื่อให้อยู่ระหว่าง 0 กับ 1 เนื่องจาก contructor ของ Retangle รับแค่สี 0 -1 จากนั้นใช้ฟังชั่น add\_patch() เพื่อนำ Retangle หลายๆอันมาต่อกันอยู่กราฟเดียว และแสดงออกมาดังรูปผลลัพธ์

A screenshot of a computer

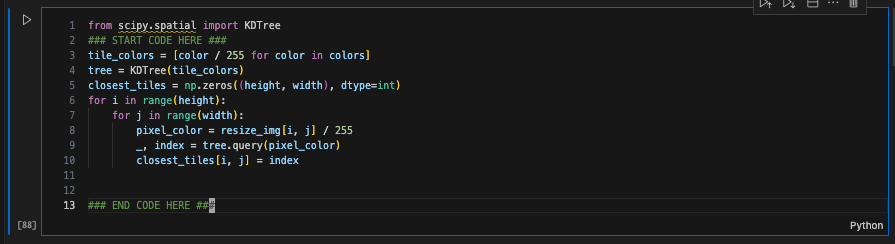
Description automatically generated

* เอารูปมาพล็อตเฉยๆครับ ไม่มีอะไร

A screenshot of a computer program

Description automatically generated

* เรานำรูปจากข้อก่อนหน้ามา resize โดยใส่ interpolation=cv2.INTER\_AREA เพื่อป้องกันไม่รูปพังตอนใส่ tile
* สร้าง mosaic\_img ให้มีขนาดเท่ากับรูปต้นฉบับแล้ว fill ด้วย 0 ทุกๆ channel ทุกๆ pixel
* Loop ที่ความกว้างและความสูงของรูปภาพ โดยมี step เป็น tile size ( ข้ามไปทีละ tile ) และทำการเฉลี่ยสีแต่ละกล่องย่อยๆ (ในที่นี่จะเป็นกล่องขนาด 15 x 15 pixel)
* นำรูปที่ได้มาทำการ plot



* ใช้ฟังก์ชัน query ของ KDTree เพื่อหา index ที่ใกล้ที่สุดของสีในแต่ละ pixel แล้วเก็บใส่ closest\_tiles

A screenshot of a computer screen

Description automatically generated

* พล็อตโดยใช้ cmap=’viridis’

A screen shot of a computer

Description automatically generated

* List tiles directory และอ่านไฟล์ image และ นำ tile images นั้น ใส่ไปใน list ชื่อ tiles
* ลูปที่ละกล่องย่อย -> หาสีเฉลี่ย -> ไปหารูปภาพ tile ที่มีสีใกล้เคียงกับสีเฉลี่ยมากที่สุด

A screen shot of a computer

Description automatically generated

* นำรูปจากข้อก่อนหน้ามา plot โดยมีรูป original , mosaic, รูปที่มี tile ข้างใน, รูป closest\_tile

A screen shot of a computer

Description automatically generated

* นำสิ่งที่ทำไปจากข้อก่อนหน้ามาทำเป็น function ได้แก้ load\_tiles() เพื่ออ่านไฟล์ tiles, closest\_tile\_index() เพื่อหา tile ที่มีสีใกล้กับ color ที่สุด, create\_mosaic() เพื่อนำ tile ไป filled กับรูปต้นฉบับเหมือนโค้ดด้านบน แต่ในนี้จะใช้วิธีการ KNN โดยจะรับ parameter k เข้ามาด้วย เพื่อเป็นจำนวน neighbor ที่ต้องการเทรนใน KNN และใช้ฟังก์ชัน Kneighbors ในการ indicate region color ของแต่ละ pixel แทน

A screen shot of a computer program

Description automatically generated

* เอาฟังก์ชันมาเรียกใช้ โดยปรับ k เป็น 1, 10, 50 ตามลำดับ โดยผลลัพธ์ดังรูปข้างล่าง

A screenshot of a video game

Description automatically generated