ณสิต ผลัญชัย 65010273

วิวัตร เตชะโกศล 65011001

Lab 4

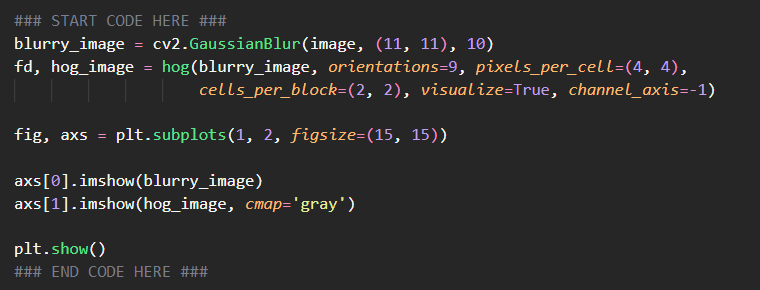
A screen shot of a computer screen

Description automatically generated

ทำการโหลดรูปแล้วเปลี่ยนสีจาก BGR เป็น RGB รวมภึงทำการ resize เป็น (250, 175)

A hummingbird flying in the air

Description automatically generated



ทำการใช้ function GaussianBlur โดยให้ kernel มีขนาดเป็น (11, 11) และ SD เป็น 10



ซึ่งจะนำขนาดของ kernel และ SD คำนวณด้วยสมการด้านบนแล้วนำ matrix ที่ได้ขนาด 11 x 11 นั้นมาทำการ convolution กับ image ที่เรา input เข้าไปแล้วเก็บไว้ใน blurry\_image

ทำการใช้ function hog เพื่อหา Histogram of Oriented Gradients โดยมี parameters ที่สำคัญเป็น

orientations คือจำนวน bin ที่จะใช้แบ่งมุม 0 – 180 องศา เช่นถ้าเป็น 9 มุม 0 – 180 ก็จะถูกแบ่งเป็น 9 ช่วงเท่าๆกัน

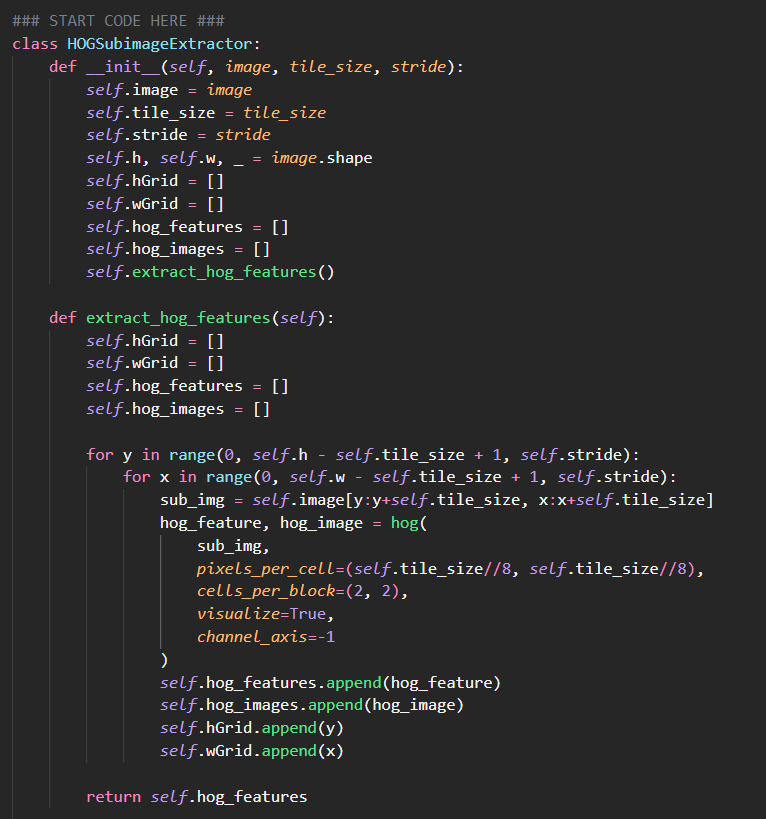
pixels\_per\_cell คือจำนวน pixel ต่อ 1 cell หรือก็คือขนาดของ cell ที่เราจะนำมาคำนวณ

cells\_per\_block คือจำนวณของ cell ต่อ 1 block โดยจะนำ block มาทำ histogram normalization ซึ่งจะมีผลต่อผลลัพธ์หลังการคำนวณโดยเฉพาะการเปลี่ยนแปลงของสีและคอนทราส

จากนั้นก็ทำการแสดงผลรูปที่ได้จากการ Gaussian Blur และ รูปที่ได้จากการทำ HOG

A blurry image of a person's hand

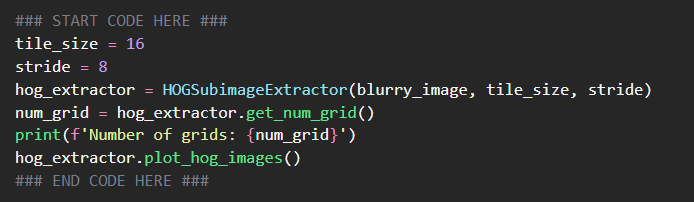
Description automatically generated



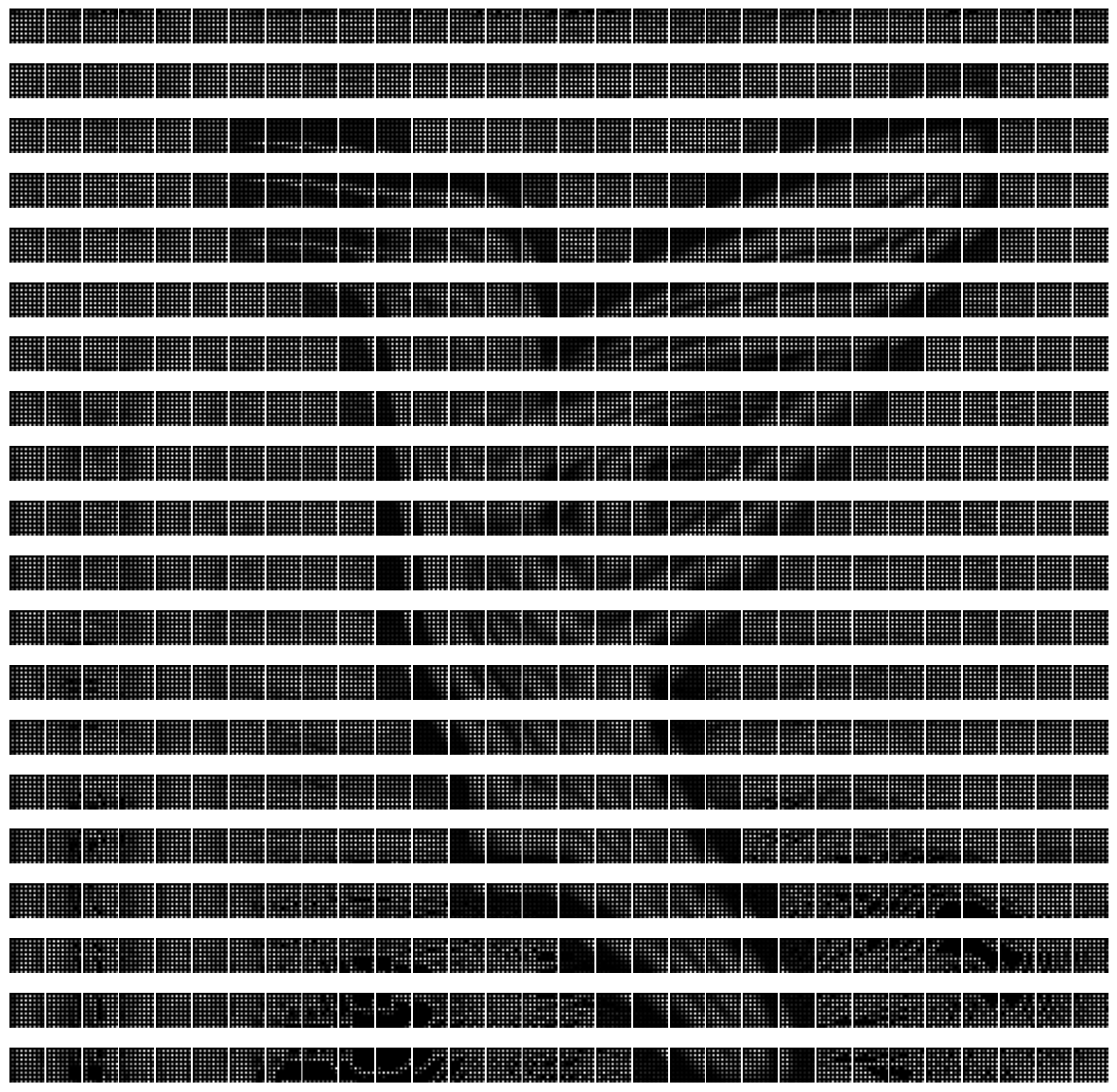
A computer screen shot of a program code

Description automatically generated

ทำการเติมโค้ดใน Class HOGSubimageExtractor โดยจะมี function extract\_hog\_features() ซึ่งจะทำงานเหมือนกับ hog ด้านบนแต่จะแบ่งรูปออกเป็นรูปเล็กๆ ก่อนนำมาหา HOG โดยเราจะต้องกำหนด tile\_size และ stride โดยรูปย่อยที่ถูกแบ่งจะมีขนาด tile\_size \* tile\_size และถูกเลื่อนไปทีละ stride โดยรูปผลลัพธ์จากการทำ hog ของแต่ละ tile จะถูกเก็บไว้ใน list hog\_image เพื่อใช้แสดงผลใน function plot\_hog\_images()



ทดลอง HOG sub-image extract ด้วย tile\_size = 16, stride = 8 จะได้จำนวน grids: (600, 600) และมีผลลัพธ์ดังรูปด้านล่าง



A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

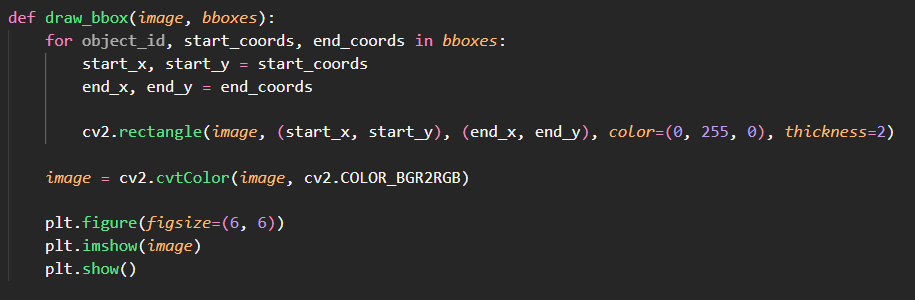
Description automatically generated

สร้าง class KMeansCluster เผื่อนำผล HOG มาทำ K-means clustering แล้วหา connected components แล้วหาของเขตของ bounding boxes โดย class นี้จะรับตัวแปร 3 ตัว ประกอบด้วย hog\_extractor จะรับ object hog\_extractor จากโค้ดด้านบนมาใช้ต่อ n\_clusters คือจำนวน clusters ที่เราจะใช้ใน K-means และ random\_state คือ seed ในการสุ่มทำให้ run แต่ละครั้งจะ random เหมือนเดิม

Function perform\_clustering() จะนำ hog features มาจัดกลุ่มด้วย model KMeans โดยใช้คำสั้ง kmeans.fit() ซึ่งจะได้ผลลัพท์ในการจับกลุ่มมาแล้วเก็บไว้ใน cluster\_labels จากนั้นทำการ reshape cluster\_labels ให้เป็น 2มิติ แล้วใช้ measure.label() เพื่อหา connected components แล้วเก็บไว้ใน all\_labels

Function plot\_cluster\_and\_labels() ใช้เพื่อแสดงผลภาพของการจัดกลุ่ม clustering และ connected components

Function get\_bounding\_boxes() ใช้เพื่อหาขอบเขตของวัตถุที่เราจะตีกรอบโดยใช้ measure.regionprops() ซึ่งจะได้พื้นที่ที่ถูก label ทั้งขนาดและพิกัด จากนั้นทำการกรองพื้นที่ขนาดเล็กเกินไปที่เราไม่ต้องการออกด้วยการลูปเพื่อให้เหลือเฉพาะพิกัดของวัตถุที่ต้องการ จากนั้นนำพิกัดมาปรับ scale ให้ตรงกับรูปจริงแล้วเก็บไว้ใน list bounding\_boxes ก่อน return ค่า bounding\_boxes



Function draw\_bbox() ใช้เพื่อวาดกรอบสี่เหลี่ยมครอบ connected components จากพิกัดที่เราได้มาด้วย cv2.rectangle แล้วแสดงผล

A screen shot of a computer code

Description automatically generated

ทำการทดลองโดยใช้ tile\_size = 16, stride = 8 จำนวน cluster = 3, random state = 32 ได้ผลลัพท์ด้านล่าง

A black and white grid

Description automatically generated

A screenshot of a graph

Description automatically generated

A hummingbird flying in the air

Description automatically generated

A screen shot of a computer code

Description automatically generated

ผลลัพท์ครั้งที่ดีที่สุด tile\_size = 32, stride = 8 จำนวน cluster = 2, random state = 32 ได้ผลลัพท์ด้านล่าง

A black and white grid

Description automatically generated

A yellow and purple squares

Description automatically generated with medium confidence

A hummingbird flying in the air

Description automatically generated