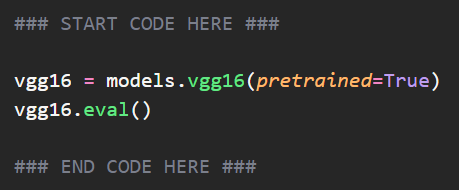
ณสิต ผลัญชัย 65010273

วิวัตร เตชะโกศล 65011001

Lab 5.1

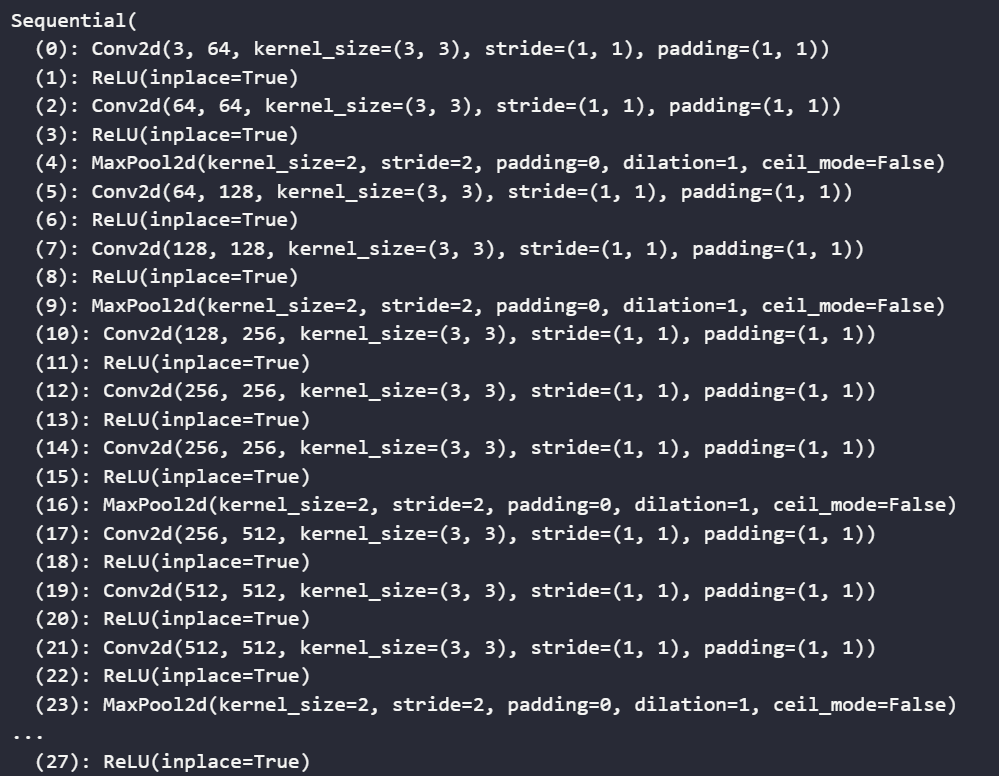


ทำการโหลด model vgg16 แล้วปรับโหมดเป็น evaluation mode

A black screen with white text

Description automatically generated

ทำการแสดงผล Layers ทั้งหมดของ model vgg16 ด้วยคำสั่ง vgg16.features โดยมีทั้งหมด 30 layers



A black screen with white text

Description automatically generated

จาก vgg16.features ด้านบนจะเห็นได้ว่า Layer ที่1 เป็น convolution และ Layer ที่2 เป็น ReLU จึงทำการดึง vgg16.features ใน index ที่ 0, 1 มาเก็บไว้ใน first\_conv และ first\_relu ตามลำดับ

A black background with white text

Description automatically generated

ทำการแสดงผล Biases ของ Layer ที่1 ได้ผลลัพท์ดังรูปด้านล่าง

A screenshot of a computer screen

Description automatically generated

A black background with colorful text

Description automatically generated

ทำการโหลดรูปเข้ามาและเปลี่ยนสีจาก BGR เป็น RGB แล้วแสดงผล

A bird standing on a branch

Description automatically generated

A computer screen shot of text

Description automatically generated

ทำการเตรียมรูปให้พร้อมสำหรับนำไป convolution ด้วยการ resize ขนาดรูปเป็น (224, 224) และ normalized ด้วยค่า mean, std of Imagenet จากเว็ป <https://pytorch.org/vision/main/models/generated/torchvision.models.vgg16.html>

โดยใช้ mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225] จากนั้นทำให้ range ของ image array อยู่ใน range 0 – 1 แล้วแสดงผลรูปที่ผ่านการ normalize ได้ภาพด้านล่าง

A bird standing on a branch

Description automatically generated

A computer screen shot of a computer code

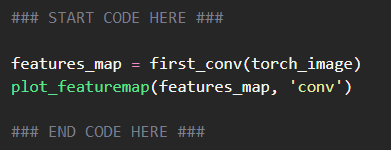
Description automatically generated

ทำการเปลี่ยน normalized\_image จาก numpy ให้เป็น PyTorch tensor แล้วทำการ permute(2, 0, 1) ซึ่งจะเป็นการเรียงมิติของ tensor ใหม่จาก [224, 224, 3] เป็น [3, 224, 224] แล้วทำการ unsqueeze(0) ซึ่งจะเป็นการเพิ่มมิติ1 เข้าไปที่ต่ำแหน่งที่ 0 เมื่อแสดงผล torch\_image.shape จะได้เป็น torch.Size([1, 3, 224, 224])

A computer screen shot of a program code

Description automatically generated

สร้าง function plot\_featuremap() ซึ่งจะนำผลจากการ convolution หรือ ReLU ที่เรียกว่า feature map มาแสดงผลโดยใช้ grid\_size เป็น มิติของ num\_feature\_maps ยกกำลัง ½ แล้วปัดขึ้น เพื่อให้แสดงผลได้ครบทุกภาพ จากนั้นก็ทำการ loop เพื่อ plot feature map ด้วยการนำ img[0, index].detach().numpy() ซึ่งจะเป็นการแปลง tensor ให้เป็น numpy array และลบมิติของ batch ออก



ทำการแสดงผลของการ convolution ครั้งแรกจะได้ผลลัพท์ดังรูปด้านล่าง

A collage of a bird

Description automatically generated

A black screen with white text

Description automatically generated

ทำการแสดงผลของการใช้ activation function ReLU ครั้งแรกจะได้ผลลัพท์ดังรูปด้านล่าง

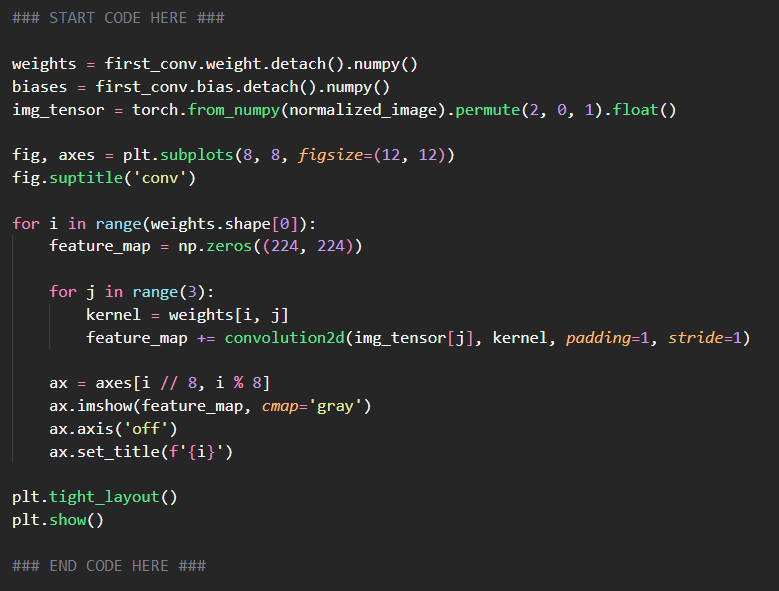
A collage of squares with numbers

Description automatically generated

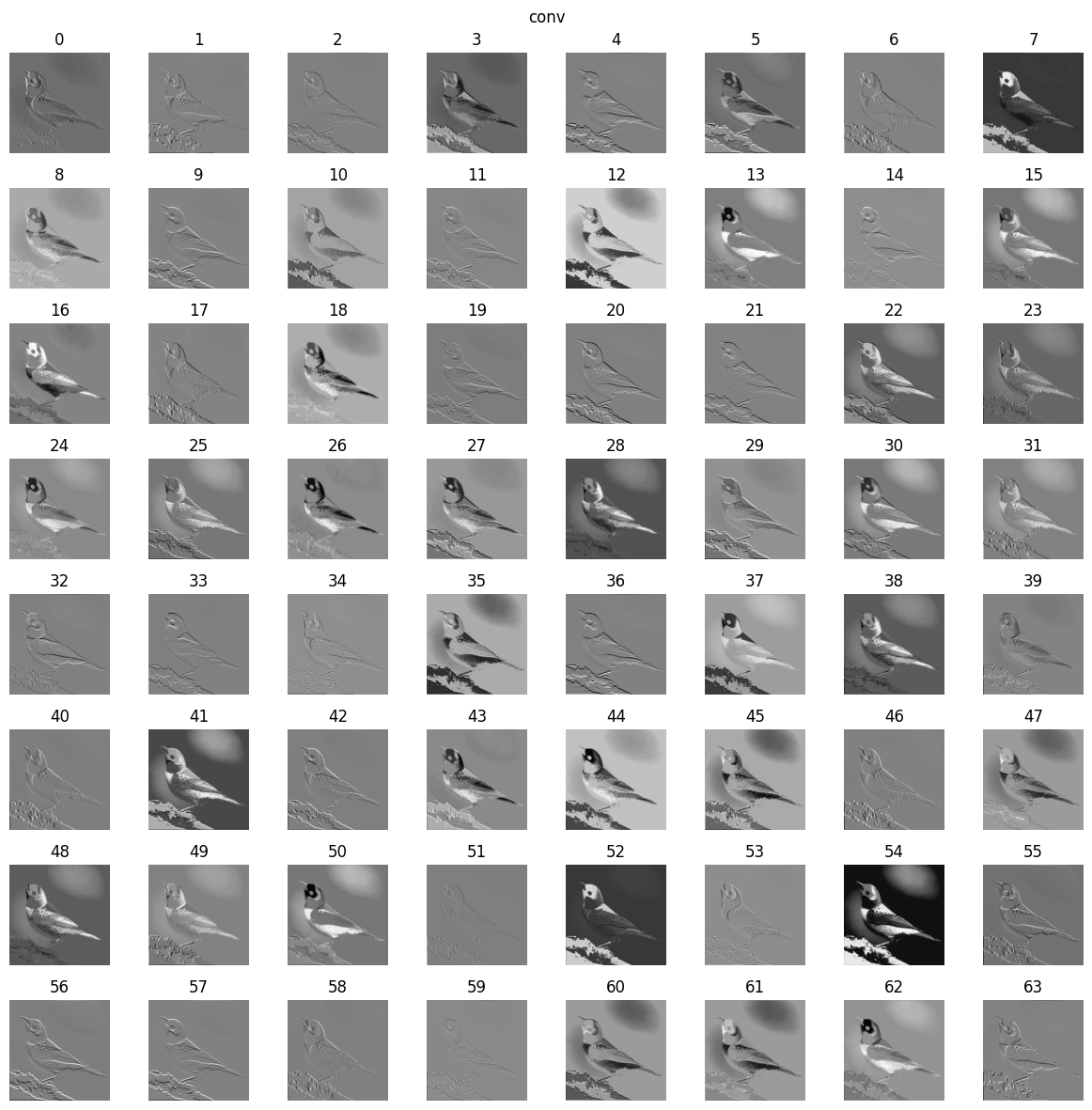
A computer screen with text on it

Description automatically generated

สร้าง function convolution2d() โดยจะทำการดึงขนาดของ img และ kernel แล้วทำการ zero padding ในกรณีที่มี Input ใน parameter padding โดยจะตั้ง default เป็น 0 ต่อจากนั้นคำนวณความสูงและความกว้างของ output ตามขนาด img, kernel, padding, และ stride เพื่อนำมาสร้าง output array ที่เป็น array 0 ขนาดตามที่คำนวณไว้ ก่อนจะ loop ไล่ดึง region มาแล้ว plot ลง output array



ทำการดึงค่า weights และ biases ของ convolution layer แรกแล้วแปลงภาพจาก numpy array เป็น pytorch tensor (Channel First) เพื่อให้ convolution ได้ ก่อนจะ loop ตามจำนวน output channels ใน convolution layer โดยการเรียก weights.shape[0] ในกรณีนี้เป็น 64 โดยแต่ละ output channel ก็จะต้อง loop อีก 3 ครั้งเนื่องจากมี 3 kernels สำหรับ RGB โดยจะนำ kernel ที่ได้มาใส่ใน function convolution2d() แล้วนำผลลัพท์มาเก็บใน feature map แล้วแสดงผลเมื่อ loop เสร็จได้ผลลัพท์ดังรูปด้านล่าง



A computer screen shot of a program code

Description automatically generated

ในการทำ ReLU activation ก็จะทำเหมือนกับการ convolution ด้านบนโดยการใช้ loop และ function convolution2d() เหมือนเดิมเพียงแต่จะต้องเพิ่ม feature\_map = np.maximum(feature\_map, 0) เพื่อให้ค่าใน feature\_map ที่ติดลบกลายเป็น 0 จะได้ผลลัพท์ดังนี้

A grid of squares with numbers

Description automatically generated