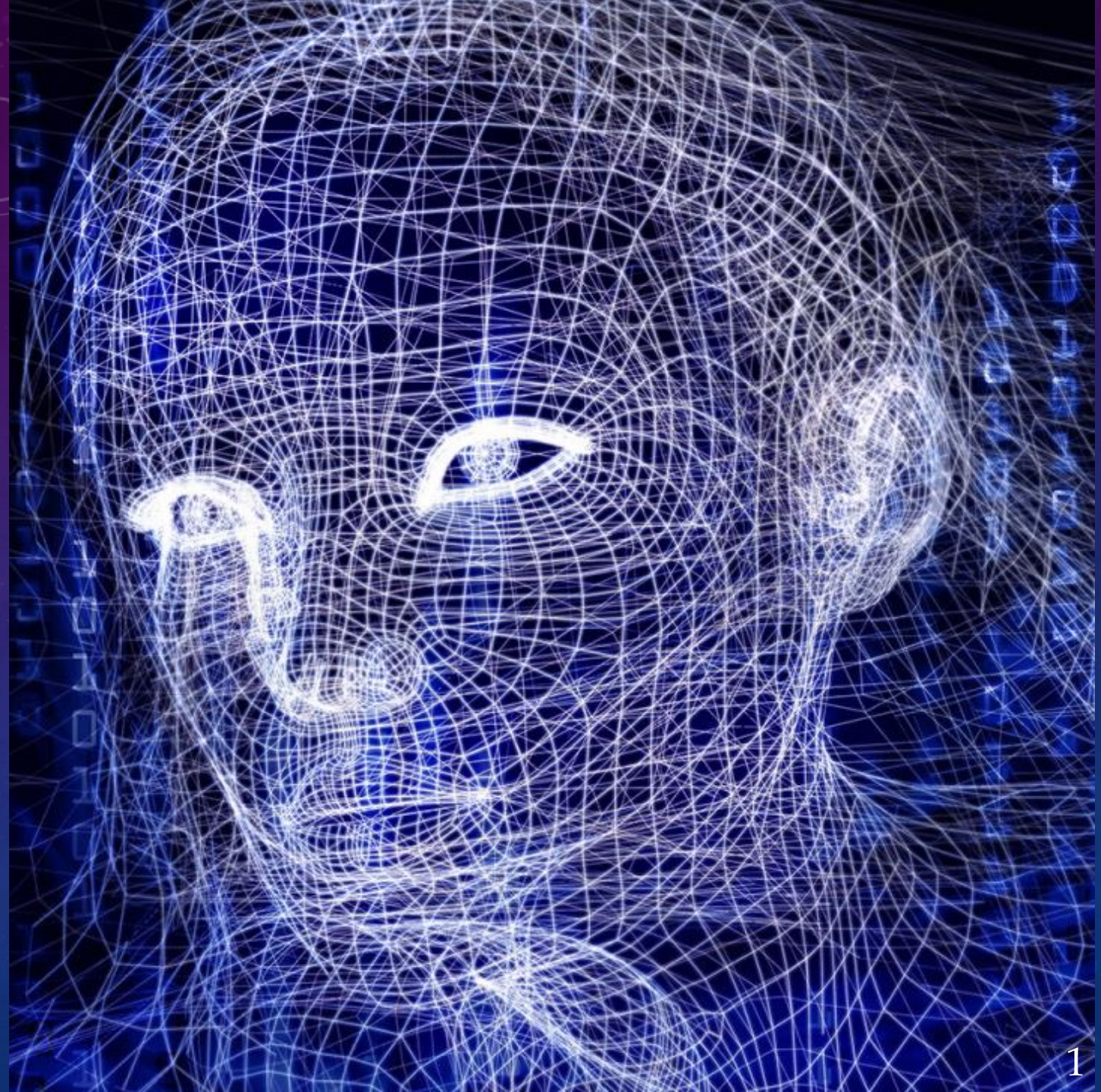
DIGITAL SURVEILLANCE SYSTEMS AND APPLICATION

CH 3_EXERCISE

徐繼聖

Gee-Sern Jison Hsu

National Taiwan University of Science and Technology



Example 3.1 Calculate Parameters

- Given a input image sized in 32×32×1, i.e., height × *width* × *channel*.
- With a convolutional kernel at 5×5×6 in "C1" layer, please compute the number of the training parameters while convolving the input image.
- In this case, we can know how many parameters need to be updated during training.

Layer Name	Input W×H×D	Kernel W×H×D/S	Output W×H×D	Params
C1: conv2d	32×32×1	5×5×6	28×28×6	1×5×5×6+6=156 weights biases
S2: pool/2	28×28×6	2×2/2	14×14×6	0
C3: conv2d	14×14×6	5×5×16	10×10×16	6×5×5×16+16 =2,416
S4: pool/2	10×10×16	2×2/2	5×5×16	0
C5: conv2d	5×5×16	5×5×120	1×1×120	16×5×5×120+120 =48,120
F6: conv2d	1×1×120	1×1×84	1×1×84	120×1×1×84+84 =10,164
F7: conv2d	1×1×84	1×1×10	1×1×10	84×1×1×10+10 =850
			Total	61,706

Exercise 3.1 Calculate Parameters

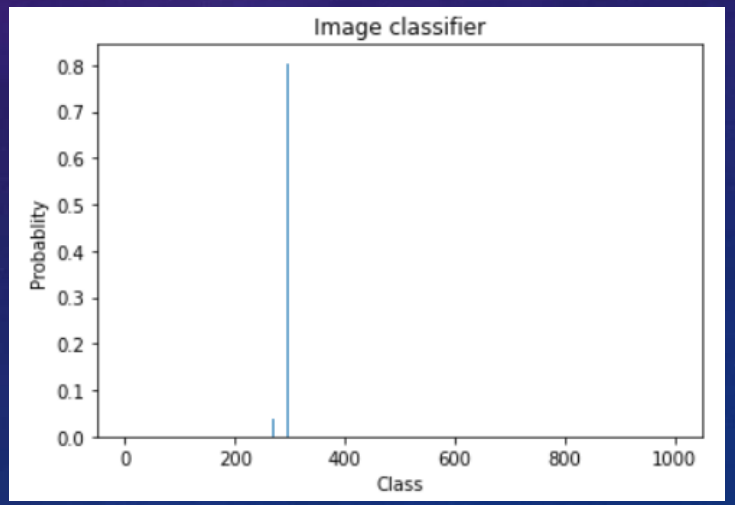
- Given a input image sized in 32×32×3, i.e., $width \times height \times channel$.
- With a convolutional kernel at 7×7×6 in "C1" layer, please compute the output size and the number of the training parameters while convolving the input image.
- Please detail how you compute the output size and the number of training parameters in your uploaded file.

Layer Name	Input $\boldsymbol{W \times H \times C}$	Kernel $\boldsymbol{W \times H \times C}$	Output $\boldsymbol{W \times H \times C}$	Params
C1:con2d	32× 32 × 3	7× 7 × 6	?	?

Example 3.2 Use VGGNet pretrained on ImageNet

- Please download the “3-2_VGGNet_ImageNet.zip” from the Moodle and unzip it.
- Follow the instruction in “How_to_Use_Colab.pdf” to upload the .ipynb file to Colab.
- Upload the “3-2_VGGNet_ImageNet.ipynb” and “imagenet1000_clsidx_to_labels.txt” to the Google Colab.
- Compare the probability of the images downloaded from Internet.

Original image: ice_bear.jpg



Probability of the classes

```
TOP_1
Probablity:0.804244875907898
Predicted: 'ice bear

TOP_2
Probablity:0.14214567840099335
Predicted: 'Arctic fox

TOP_3
Probablity:0.03769978880882263
Predicted: 'white wolf
```

Predicted class : ice bear

Example 3.2 Use VGGNet pretrained on ImageNet

```
self.pretrained_model = models.vgg16(pretrained=True)
self.pretrained_model.eval()
```

Use the VGG16 pretrained model

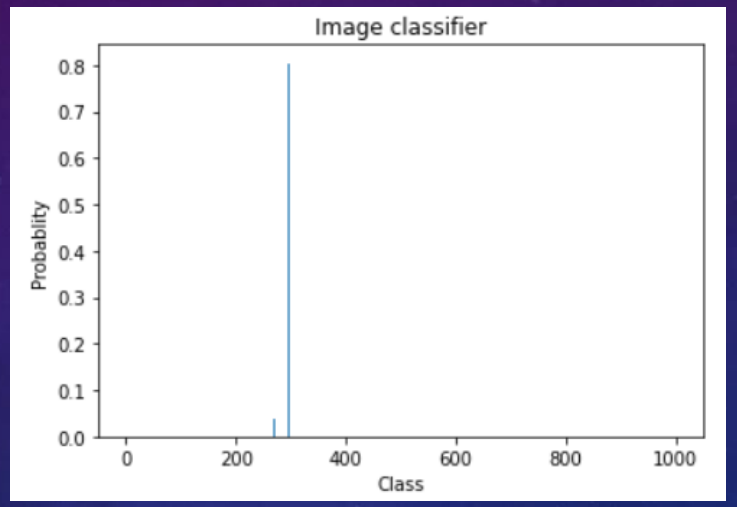
```
if __name__=='__main__':
    # get class
    c = {}
    with open("imagenet1000_clsidx_to_labels.txt") as f:
        for line in f:
            (key, val) = line.split(":")
            c[int(key)] = val.split(",")[0]
    # Define image path
    myClass=Pretrained_VGGNet('./ice_bear.jpg')
    print(myClass.pretrained_model)
    myClass.predict()
```

Load the 1000 class labels

Define Image path

Example 3.2 Use VGGNet pretrained on ImageNet

Original image: ice_bear.jpg



Probability of the classes

```
TOP_1
Probablity:0.804244875907898
Predicted: 'ice bear

TOP_2
Probablity:0.14214567840099335
Predicted: 'Arctic fox

TOP_3
Probablity:0.03769978880882263
Predicted: 'white wolf
```

Predicted class : ice bear

Exercise 3.2 Use VGGNet pretrained on ImageNet

- Please download the "3-2_VGGNet_ImageNet.zip" on the Moodle and choose your own images from Internet.
- Upload the "3-2_VGGNet_ImageNet.ipynb" and "imagenet1000_clsidx_to_labels.txt" to the Google Colab.
- Compare the probability of the images that contain multi classes and different variations (pose, occlusion).
- Please write down the results, codes, and your observations in MS Word to the Moodle.

Example 3.3 Train VGGNet on CIFAR100

- Please download the "3-3_VGGNet_CIFAR100.zip" from the Moodle and unzip it.
- Upload the "3-3_VGGNet_CIFAR100.ipynb" to the Google Colab.
- Use the VGG-16 model pretrained on ImageNet to train the CIFAR-100 dataset with the following parameters: input size = 32 (color image), batch size = 64, learning rate=0.001.
- Please search the images with following categories: bird, cat and dog.
- Given these images as input to your trained model, what are the probabilities in the output layer.

CIFAR 100 dataset contains 60000 images and consists of 100 class

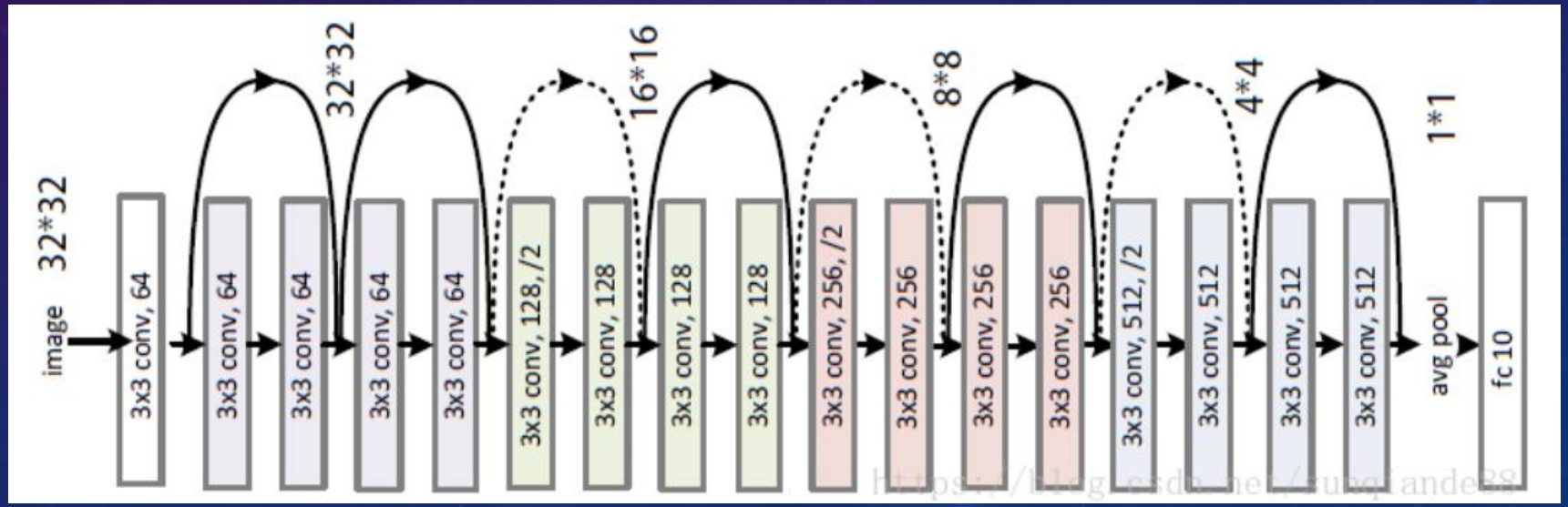
Exercise 3.3 Train VGGNet on CIFAR100

- Please download the "3-3_VGGNet_CIFAR100.zip" from the Moodle and unzip it.
- Upload the "3-3_VGGNet_CIFAR100.ipynb" to the Google Colab, please use the pre-trained model and re-train it on the CIFAR100 dataset.
- Please search the images with following categories: bird, cat and dog.
- Given these images as the input, please compare the predicted probabilities made by the pre-trained model, and your re-trained model.
- Please write down the results, codes, and your observations in MS Word to the Moodle.

CIFAR 100 dataset contains 60000 images and consists of 100 class

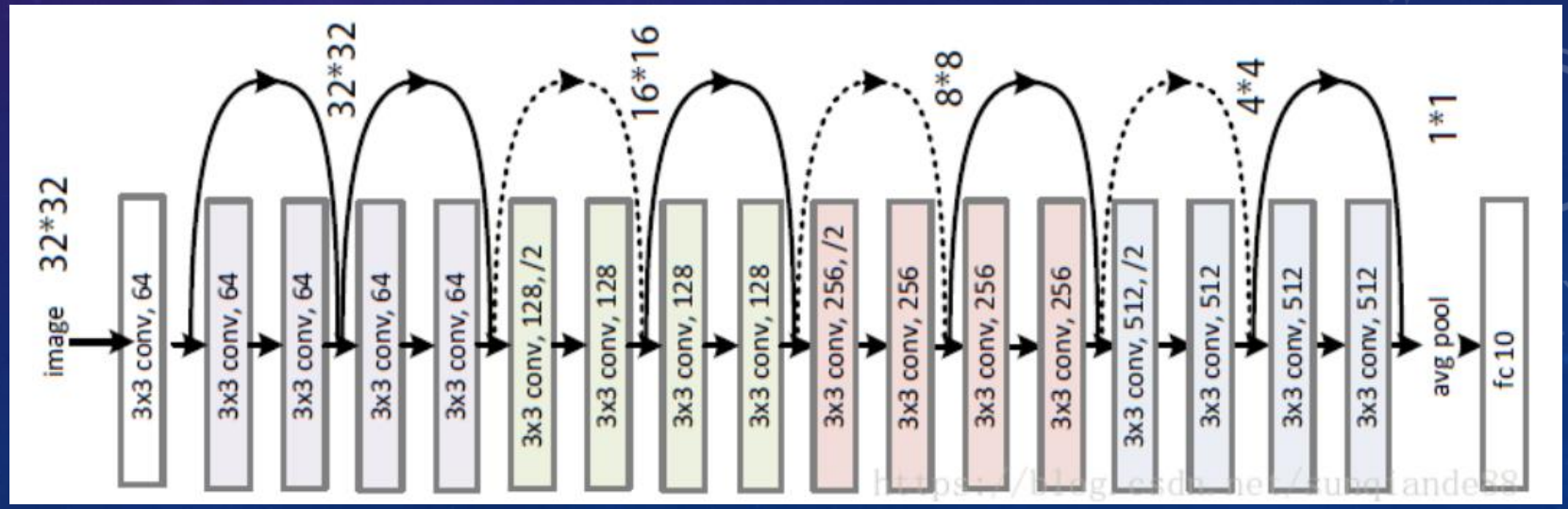
Example 3.4 Train ResNet on CIFAR-100

- Please download the "3-4_ResNet_CIFAR100.zip" from the Moodle and unzip it.
- Upload the "3-4_ResNet_CIFAR100.ipynb" to the Google Colab.
- Use the ResNet model pretrained on ImageNet to train the CIFAR-100 dataset with the following parameters: input size = 32 (gray image), batch size = 64, learning rate=0.001.
- Please search the images with following categories: bird, cat and dog.
- Given these images as input to your trained model, what are the probabilities in the output layer.



Exercise 3.4 Train ResNet on CIFAR-100

- Please download the "3-4_ResNet_CIFAR100.zip" from the Moodle and unzip it.
- Upload the "3-4_ResNet_CIFAR100.ipynb" to the Google Colab, please use the pre-trained model and re-train it on the CIFAR100 dataset.
- Please search the images with following categories: bird, cat and dog.
- Compare the pretrained model with trained model by you. Given these images as input to model, what are the probabilities in the output layer.
- Compare two results made by Exercise 3.3 with the results in Exercise 3.4.
- Please write down the results, codes, and your observations in MS Word to the Moodle.



Example 3.5 Feature Map Visualization

- Please download the “3-5_Feature_map_visualization.zip” from the Moodle, which is built on the VGG-16 pretrained on the ImageNet.
- Upload the “3-5_Feature_map_visualization.ipynb” and “imagenet1000_clsidx_to_labels.txt” to the Google Colab.
- Choose your own images from Internet.
- Compare the feature maps that extract from layer 5 and observe the size and dimension of the feature maps.

Original image: cat.jpg

Feature map: cat_feature_5.jpg

Example 3.5 Feature Map Visualization

```
if __name__=='__main__':
    # Define image path and select the layer
    myClass=FeatureVisualization('./cat.jpg',5)
    print (myClass.pretrained_model)

    myClass.save_feature_to_img()
```

```
class FeatureVisualization():
    def __init__(self,img_path,selected_layer):
        self.img_path=img_path
        self.selected_layer=selected_layer
        # Load pretrained model
        self.pretrained_model =
        models.vgg16(pretrained=True).features
```

```
Sequential(
  (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (1): ReLU(inplace=True)
  (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (3): ReLU(inplace=True)
  (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (6): ReLU(inplace=True)
  (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (8): ReLU(inplace=True)
  (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (11): ReLU(inplace=True)
  (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (13): ReLU(inplace=True)
  (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (15): ReLU(inplace=True)
  (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (18): ReLU(inplace=True)
  (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (20): ReLU(inplace=True)
  (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (22): ReLU(inplace=True)
  (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (25): ReLU(inplace=True)
  (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (27): ReLU(inplace=True)
  (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (29): ReLU(inplace=True)
  (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
)
```

Example 3.5 Feature Map Visualization

```
def save_feature_to_img(self):
    #to numpy
    feature=self.get_single_feature()
    feature=feature.data.numpy()

    #use sigmoid to [0,1]
    feature= 1.0/(1+np.exp(-1*feature))

    # to [0,255]
    feature=np.round(feature*255)
    print(feature[0])

    #Save the feature map
    save_name = './cat_feat_' + str(self.selected_layer) + '.jpg'
    cv2.imwrite(save_name, feature)
```

```
def get_single_feature(self):
    #Get the feature map
    features=self.get_feature()
    print("features:{}".format(features.shape))
    feature=features[:,0,:,:]
    print("feature:{}".format(feature.shape))
    feature=feature.view(feature.shape[1],feature.shape[2])
    print("features:{}".format(feature.shape))

    return feature
```

```
def get_feature(self):
    #Image  preprocessing
    input=self.process_image()
    print("input.shape:{}".format(input.shape))
    x=input
    for index,layer in enumerate(self.pretrained_model):
        x=layer(x)
        if (index == self.selected_layer):
            return x
```

Define function that we can use.
For more detail about python functions,
please refer to the following link:
https://www.tutorialspoint.com/python/python_functions.htm

Example 3.5 Feature Map Visualization

Original image: cat.jpg

Feature map: cat_feature_5.jpg

Exercise 3.5 Feature Map Visualization

- Please download the “3-5_Feature_map_visualization.zip” from the Moodle, which is built on the VGG-16 trained on the ImageNet.
- Upload the “3-5_Feature_map_visualization.ipynb” and “imagenet1000_clsidx_to_labels.txt” to the Google Colab.
- Choose your own images from Internet.
- Compare the feature maps that extract from layer 8 and observe the size and dimension of the feature maps.
- Please write down results and your codes in MS Word to the Moodle.