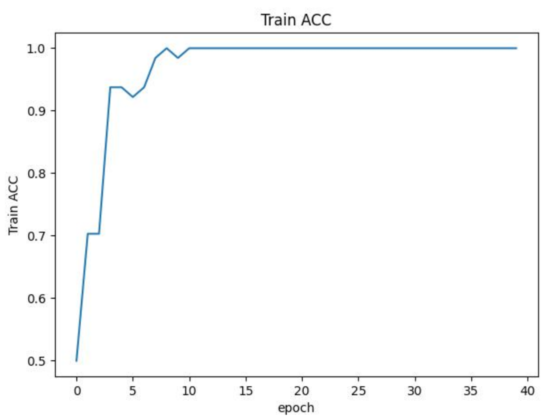
COMP5623 Coursework on Image Classification and Visualizations with Convolutional Neural Networks – ImageNet10

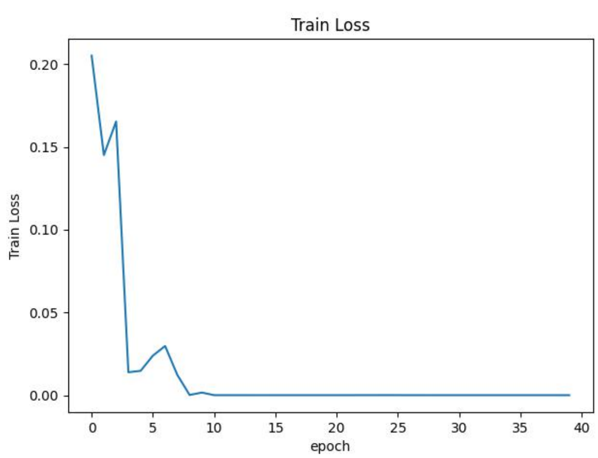
Name	**YANGTIAN YI**
Student username & ID	**sc20yy**

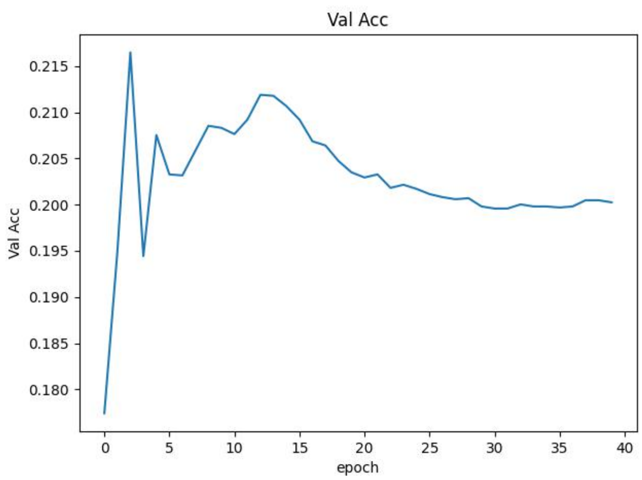
QUESTION I [55 marks]

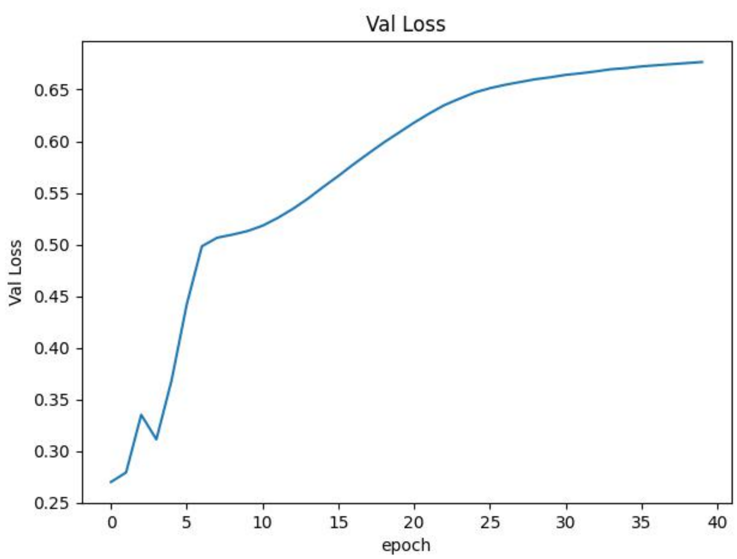
1.1 Single-batch training [16 marks]

1.1.1. Display graph 1.1.1 (training & validation loss over training epochs) and briefly explain what is happening and why. [4 marks]







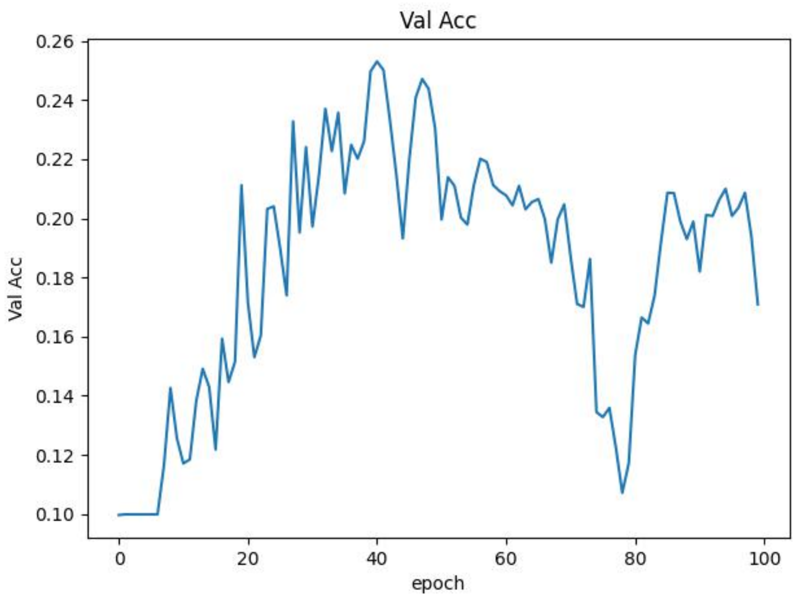


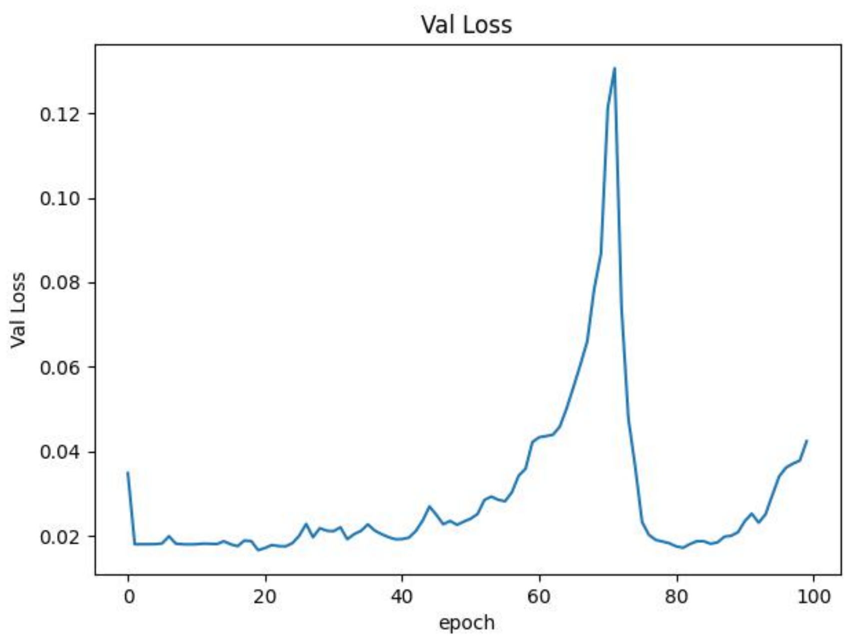
Since only one batch of data is used to train the model, the target accuracy is quickly achieved.

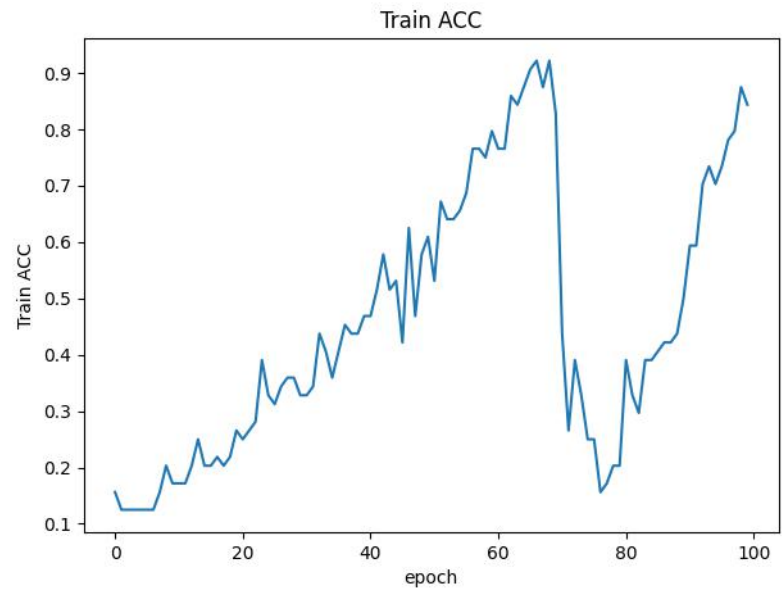
The validation set loss rises significantly after the maximum accuracy is reached

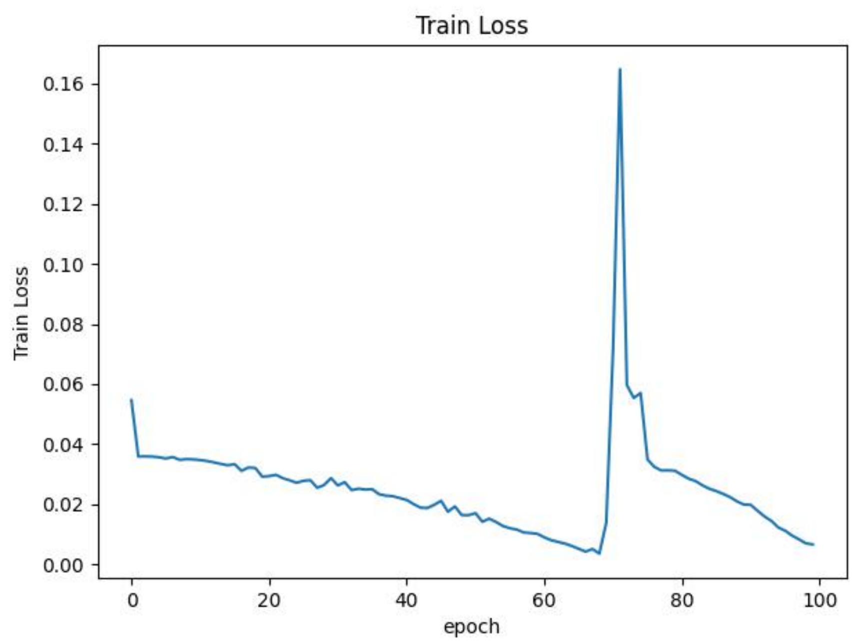
Inconsistent data distribution in the training and validation sets, or the training set is too small and does not contain all cases in the validation set, i.e. overfitting

1.1.2 Display graph 1.1.2 (training & validation loss over training epochs, with modified architecture) and explain how and why it shows that the model is overfitting the training batch. [8 marks]









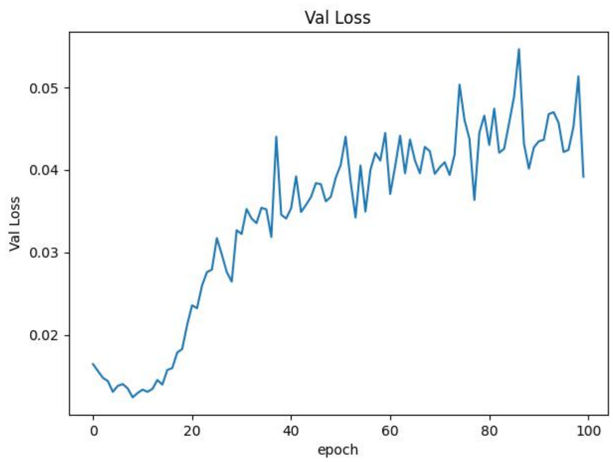
As the training sample increases, the training error increases and the test error decreases, but the training error is much smaller than the test error

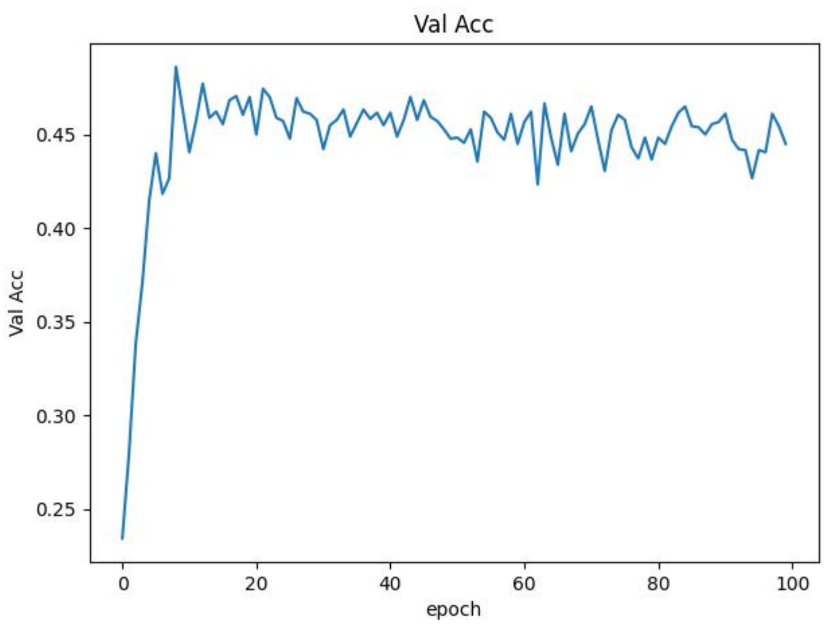
1.1.3 Fill in table 1.1.3 (your adjusted architecture after single-batch training), adding rows and columns as necessary. [4 marks]

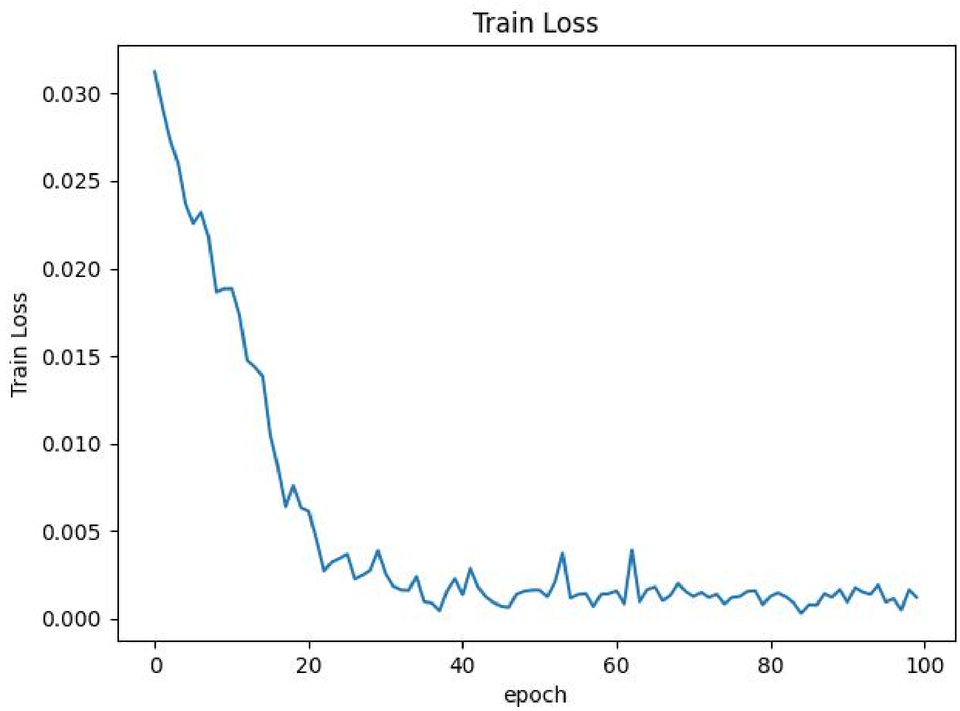
Input channels	Output channels	Layer type	Kernel size
3	64	nn.Conv2d	11
64	192	nn.Conv2d	5
192	384	nn.Conv2d	3
384	256	nn.Conv2d	3
256	256	nn.Conv2d	3

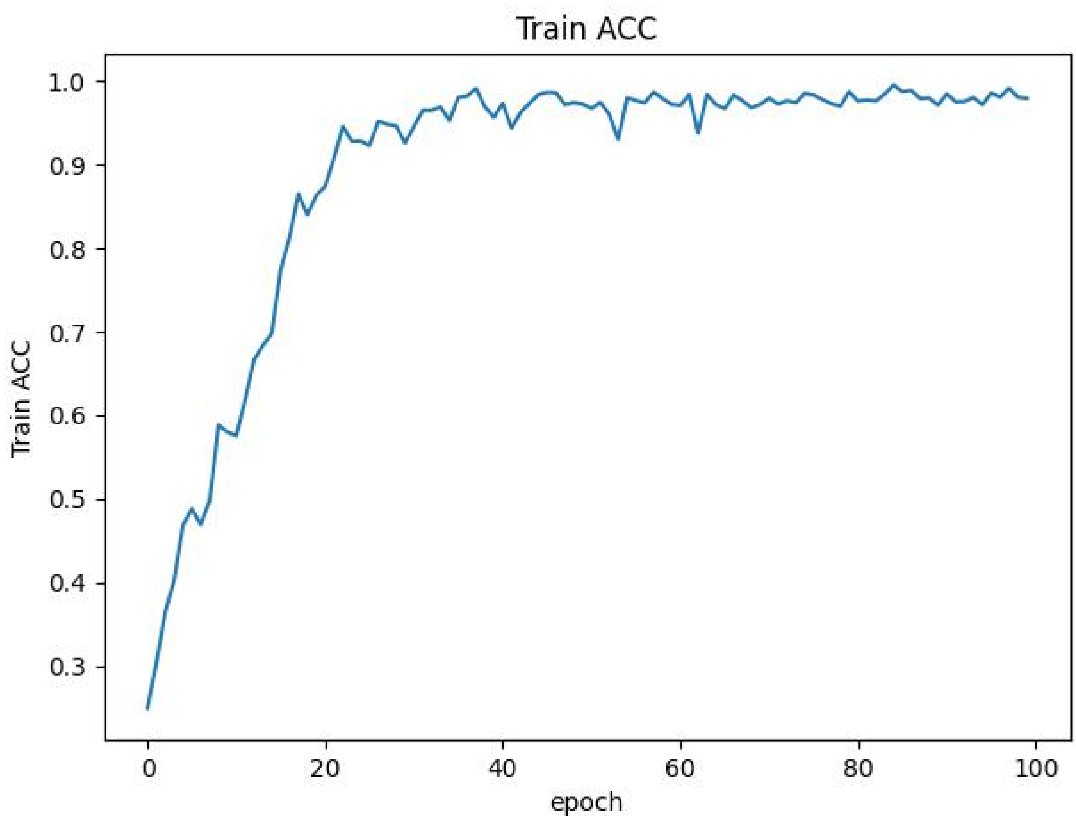
1.2 Fine-tuning on full dataset [18 marks]

1.2.1 Display graph 1.2.1 and indicate what the optimal number of training epochs is and why. [4 marks]






Train Loss
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Train Loss
0
20
40
60
80
100
epoch


Train ACC
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Train ACC
0
20
40
60
80
100
epoch

I think the optimal number of training sessions should be between 60 and 80, as the graph shows that the model is not yet overfitting and the correctness of both the training and validation sets is relatively high, with a relatively low loss.

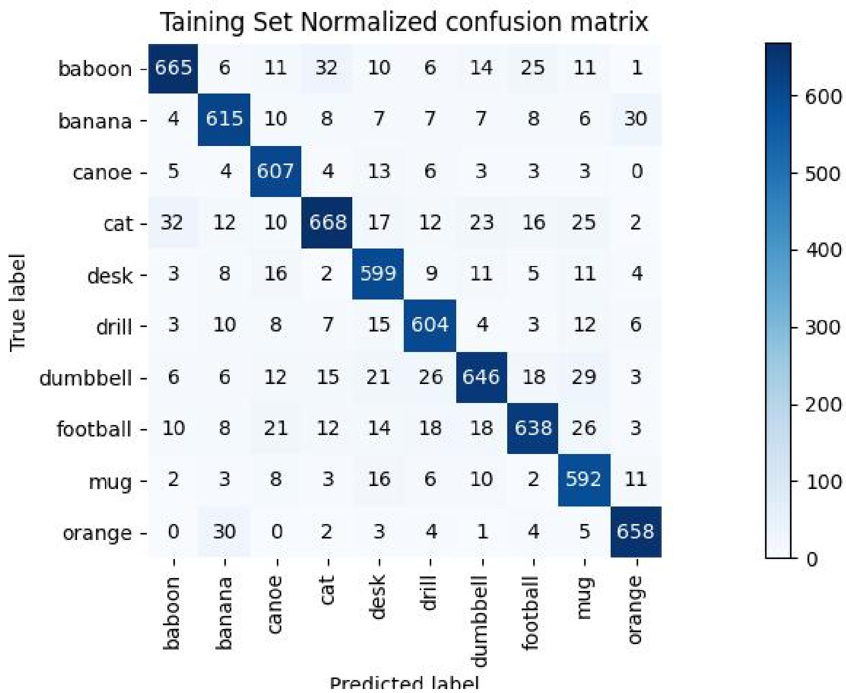
1.2.2 Describe in detail your fine-tuning process on the complete dataset, including any adjustments you made to the network or training process to increase prediction accuracy. Explain why these adjustments increased accuracy. [10 marks]

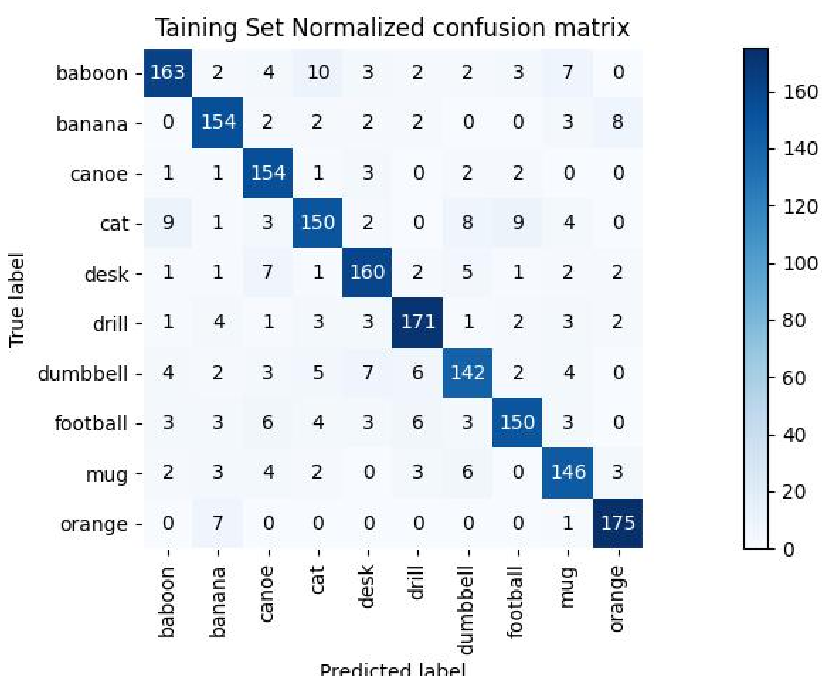
I am using 4 loops to define the model, one including nn.Conv2d, nn.ReLU(), nn.MaxPool2d.

And I used nn.dropout(0.5) to mitigate the overfitting problem more effectively and to regularise the model.

I also used L2 regularisation on the optimiser to mitigate the overfitting problem.

1.2.3 Display two confusion matrices 1.2.3 (one each for complete validation set and complete training set) for your final trained model and interpret what is shown. [4 marks]





From the confusion matrix, it can be seen that the model has excellent recognition, with decreasing values as you look diagonally to the sides.

At the same time you can calculate the correct rate, adding up the values on the diagonal and dividing by the total number of images 9000, you can conclude that the accuracy rate is at 69.9%

1.3 Evaluation and code [21 marks]

1.3.1 Please include *[my_student_username]_test_preds.csv* with your final submission. [8 marks]

1.3.2 Please submit all relevant code you wrote for Question I in Python file *[my_student_username]_q1.py*. No need to include the config or ImageNet10 files. [13 marks]

No response needed here.

QUESTION II [45 marks]

2.1 Preparing the pre-trained network [20 marks]

2.1.1 Read through the provided template code for the AlexNet model *explore.py*. What exactly is being loaded in line 59? [2 marks]

```
Downloading: "https://download.pytorch.org/models/alexnet-owt-4df8aa71.pth" to C:\Users\11571/.cache\torch\hub\checkpoints\alexnet-owt-4df8aa71.pth
100%|██████████| 233M/233M [00:52<00:00, 4.68MB/s]
AlexNet(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(11, 11), stride=(4, 4), padding=(2, 2))
    (1): ReLU(inplace=True)
    (2): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(64, 192, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))
    (4): ReLU(inplace=True)
    (5): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(192, 384, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU(inplace=True)
    (8): Conv2d(384, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): ReLU(inplace=True)
    (10): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): MaxPool2d(kernel_size=3, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(6, 6))
  (classifier): Sequential(
    (0): Dropout(p=0.5, inplace=False)
    (1): Linear(in_features=9216, out_features=4096, bias=True)
    (2): ReLU(inplace=True)
    (3): Dropout(p=0.5, inplace=False)
    (4): Linear(in_features=4096, out_features=4096, bias=True)
    (5): ReLU(inplace=True)
    (6): Linear(in_features=4096, out_features=1000, bias=True)
```

2.1.2 Write the code in *explore.py* after line 50 to read in the image specified in the variable `args.image_path` and pass it through a single forward pass of the pre-trained AlexNet model. [5 marks]

2.1.3 Fill in function `extract_filter()` after line 84 extracting the filters from a given layer of the pre-trained AlexNet. [4 marks]

2.1.4 Fill in function `extract_feature_maps()` after line 105 extracting the feature maps from the convolutional layers of the pre-trained AlexNet. [6 marks]

Please submit all your Question II code in a Python file *[my_student_username]_explore.py*.

No response needed here.

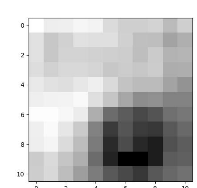
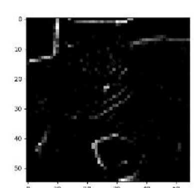
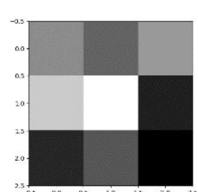
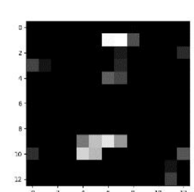
2.1.5 Describe in words, not code, how you ensure that your filters and feature maps are pairs; that the feature maps you extract correspond to the given filter. [3 marks]

Take the first convolution kernel and the corresponding feature map via the pytorch hook mechanism

2.2 Visualizations [25 marks]

2.2.1 For three input images of different classes, show three pairs of filters and corresponding feature maps, each from a different layer in AlexNet. Indicate which layers you chose. For each pair, briefly explain what the filter is doing (for example: horizontal edge detection*)* which should be confirmed by the corresponding feature map. [15 marks]

Image #1, class: ______cat_____

	Filter	Feature map	Brief explanation
Early layer			Edge information, edge detection, beveled edge detection
Intermediate layer			Advanced semantic convolution for abstracting underlying features

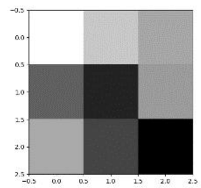
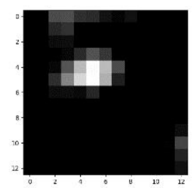
Deep layer			Decision-level convolution for making decisions

Image #2, class: _____banana______

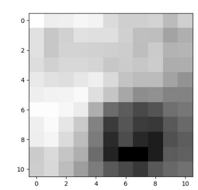
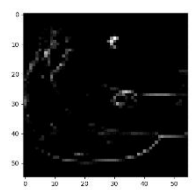
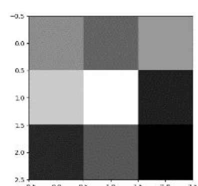
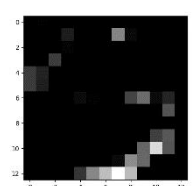
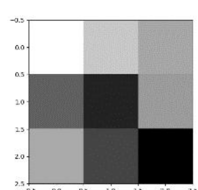
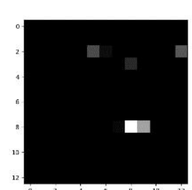
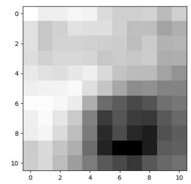
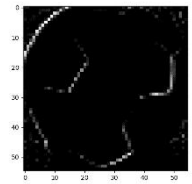
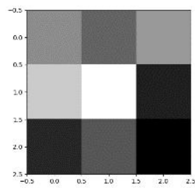
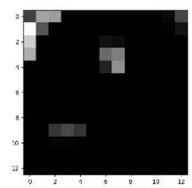
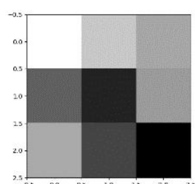
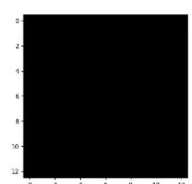
	Filter	Feature map	Brief explanation
Early layer			Edge information, edge detection, beveled edge detection
Intermediate layer			Advanced semantic convolution for abstracting underlying features
Deep layer			Decision-level convolution for making decisions

Image #3, class: ____football_______

	Filter	Feature map	Brief explanation
Early layer			Edge information, edge detection, beveled edge detection

			
Intermediate layer			Advanced semantic convolution for abstracting underlying features
Deep layer			Decision-level convolution for making decisions

2.2.2 Comment on how the filters and feature maps change with depth into the network. [5 marks]

As the depth of the network deepens, the convolutional kernel becomes more and more abstract and the feature semantics become more and more advanced

Marks reserved for overall quality of report. [5 marks]

No response needed here.