Computer Vision and Artificial intelligence Coursework
Part 3

Part 3: Coursework Description:
You will need to submit the following files:

- A PDF file of the report: part3report.PDF (I will check this file for marking)
- A Notebook of the code: part3coursework.ipynb *(I will only check this file if I need to verify and find evidence of the claim in the report as well as plagiarism)*

Main Tasks
Implement a Convolutional Neural Network (CNN) on the CIFAR-10 dataset and solve the following three problems using pytorch. *[tentative mark distribution is given in the brackets]*:

1. For the training, use **early stopping** and **save the model** that produces the **best validation results**. (you will need to use a training set and a validation set) [*Marks 10*: 5+3+2]
2. Show the performance of your designed CNN **with** and **without batch normalisation**. Use a convergence graph to show the difference in their performances [*Marks 10*: 5+5]
3. Visualise the Convolutional Features / Filters. This could be done by using imshow or similar methods. Show how filter features change over different layers over a test image. [*Marks 20*: 10+10]

Design of CNN Architecture
You can design any type of CNN architecture that could have a minimum of three Conv layers and 3 FC layers. Between Conv layers, you will need to use MaxPool layers. For the batch normalisation question (question 2), you will need to add batch normalisation layers as well.
VGG16 may take a long time to run on some laptops or may not be feasible to run on CPU. Therefore, if your computation permits, such as having GPU, you can use a standard VGG16 as your network. There is no harm in using simple, small CNN architecture satisfying the above constraints.

CNN training hyperparameter
There are many user-defined hyperparameters, such as batch size, learning rate, type of optimisation, training epochs, early stopping patience, number of filters, filter size, etc. You will need to set these parameters based on your choice. If you would like to optimise them, then you can follow the following research article:
Taylor, R., Martino, I., & Nicosia, G. Ojha, V. (2021, November). Sensitivity analysis for deep learning: ranking hyper-parameter influence. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 512-516). IEEE.

Training Dataset
You can reshape the dataset size or normalise it. This is optional.

Feature Visualisation
You must visualise both the raw trained filters and filter features on a test input of your choice.

Plagiarism
What will not constitute plagiarism: if you do your own research and develop/code your own notebook, you will be fine. If you do your own code, you may have different choices of hyperparameters, different types of architecture, sizes of filters, normalisation of datasets, and so on. Therefore, it is easy to do your own coding by researching topics. But passing on the entire code from an external source is plagiarism, including your friend'sfriend's code. You can also cite articles that you have researched to solve this coursework. It is perfectly fine the piece or article you have studied will have similar code snippets, etc, as long as you cite them.

Marks
Part 1, Part 2, and Part 3 coursework roughly have the distribution of 30%, 30%, and 40%.

Computer Vision and Artificial intelligence Coursework
Part 3

Report Format
Answer each question (Task) as follows: (get an inspiration from Practical 2 and Practical 3 write that have provided.) **Page limit. Max 6 pages.**

1. For the training, use **early stopping** and **save the model** that produces the **best validation results**. (you will need to use a training set and a validation set) [*Marks 10*: 5+3+2]

Answer:
You will need to show in code snippet from how your model look like.
Mention the values of all hyperparameters you have defined yourself for the best possible results.
You need to show using a convergence graph that indicate divergence between training and validation loss over a predefined number of epochs.

2. Show the performance of your designed CNN **with** and **without batch normalisation**. Use a convergence graph to show the difference in their performances [*Marks 10*: 5+5]

Answer:
You will need to show in code snippet from how your model look like after batch normalization.
Answer to question 1 already have model without Batch normalization. Therefore, no need to repeat that. Only show code snippet with Batch norm. You can avoid using Early Stopping (that not an important for answering this question)
Mention the values of all hyperparameters you have defined yourself for the best possible results.
You need to show using a convergence graph that indicate performance comparing with/without batch norm losses.

3. Visualise the Convolutional Features / Filters. This could be done by using imshow or similar methods. Show how filter features change over different layers over a test image. [*Marks 20*: 10+10]

You will need to show in code snippet from how your model look like (you can use any of the model from question 1 and question2).
Show the images of filters/kernels that you generated.
Show the test images, the filters after using test images.