

1. In this question, you have to compare the performance on a binary classification task of the following models:

- VGG (1 block)
- VGG (3 blocks)
- VGG (3 blocks) with data augmentation
- Transfer learning using VGG16 or VGG19 with tuning all layers (including tuning convolution layers)
- Transfer learning using VGG16 or VGG19 with tuning only final MLP layers (excluding convolution layers)

[Refer this article](<https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/>) You do not need to write your own code (but it might be easier if you do so!). You can reuse the code from the post. Or, you could roll out your own implementation. Either way, you should be able to explain your code during the viva.

You need to create the dataset on your own based on your first names of any two members from the group. For instance if the first name of the team members are: **Vaidehi** and **Raghav**, they can choose a dataset of their liking based on any names, place, animal or thing. As examples:

- Vulture v/s Rat
- Vada Pav v/s Roti

You can refer to [resource 1](<https://python.plainenglish.io/how-to-automatically-download-bulk-images-for-your-dataset-using-python-f1effba7a03>) or [resource 2](<https://github.com/JorgePoblete/DuckDuckGoImages>) or plainly download 100 images of both classes (total 200 images). Of these 100 images of each class, we will use 80 for training and 20 for testing. You get 1 mark for dataset creation [1 mark]

Create a table with models as rows and the following columns [2.5 marks (0.5 marks for each model)]

- Training time
- Training loss
- Training accuracy
- Testing accuracy
- Number of model parameters

We will now be using Tensorboard for visualising network performance. You are suggested to refer to:

- [PyTorch + Tensorboard](https://www.youtube.com/watch?v=RLqsxWaQdHE)
- [Tensorflow + Tensorboard](https://www.youtube.com/watch?v=k7KfYXXrOj0)

Use Tensorboard to log the following and present screenshots/images
[1 mark]

Scalars

- Training loss v/s iterations (and not epochs)
- Training accuracy v/s iterations (and not epochs)
- Testing accuracy v/s iterations (and not epochs)

Images

- Show all images from the test set and their predictions

Now you have to present various insights. For instance, you should discuss the following: [2 marks (0.5 marks for each question)]

- Are the results as expected? Why or why not?
- Does data augmentation help? Why or why not?
- Does it matter how many epochs you fine-tune the model? Why or why not?
- Are there any particular images that the model is confused about? Why or why not?

Now, create an MLP model with parameters comparable to VGG16 and compare your performance with the other models in the table. You can choose the distribution of the number of neurons and the number of layers. What can you conclude? [0.5 marks]

2. In this question you will implement and compare the following different KNN variants (we will discuss the a) and b) part of this in the upcoming tutorial)

- a. LSH (1.5 marks)
- b. KD-tree (1.5 marks)
- c. Naive version of KNN (0.5 marks)

Vary dataset size N , number of dimensions D to do training and testing time and memory comparison for finding K nearest neighbours. (1 mark)

Now, in a 2d randomly generated dataset visually show how many of the K closest neighbours appx. K NN methods miss out due to their approximate nature.

Also show the partitions in the 2d space. (0.5 marks)