Dr. Tolga Soyata *© George Mason University, Department of ECE* *tsoyata@gmu.edu*

ECE655 (Advanced GPU Programming and Deep Learning) Project 6 (15 points)

Deadline: Nov 30, 2025 11:59PM. 2 point penalty per each late day

- In this project, you will be using your knowledge from Lectures 8-10.
- You will use the EfficientNet-B3 model.
- Document your results in a report, written in LaTeX (Overleaf).
- Name your code **`P6A.py`**, **`P6B.py`**, … just like the previous projects.
- Submit your report and code files under **Project 6 Submission Area**, *as a single ZIP file*, which must include:
    - Your Python source files and other necessary files for me to generate your results.
    - Your Report PDF and source files (the entire Overleaf directory)

PROJECT 6 DESCRIPTION:

You will use the popular EfficientNet-B3 model and the CIFAR100 dataset. Both of them are a part of PyTorch (torchvision).
CIFAR100 dataset has 20 super-classes and 100 sub-classes (5 under each super-class).

Note: **Test accuracy** refers to the same thing as **validation accuracy**.

PART A (3 points):

Get the EfficientNet-B3 model; determine where its layers are, i.e., featurizer and classifier. Unplug and replace the appropriate layer in the classifier to re-train the model. Freeze the rest. This is exactly what we did in the class.

Train this new "modified EfficientNet-B3" to classify CIFAR100 images for 3 epochs.

Do not apply preprocessing in this part.

- Report your training time (per epoch) without preprocessing.
- Once the training is complete, report your training and test accuracy (after 3 epochs). You do not need to plot the losses. Only the Top-1 accuracies need to be reported.

PART B (4 points):

Apply the pre-processing technique we learned in the class to preprocess CIFAR100 images and save the resulting tensors to disk. Using these preprocessed tensors, this time train the model for 30 epochs. Inside your training loop, collect relevant accuracies and save them into a Python variable; you will need them to plot them later.

- Report your training time (per epoch) <u>with preprocessing</u>.
- Plot the training and test **Top-1 accuracy** against epoch.
- Plot the test **Top-5 accuracy** against epoch.

```
X axis = epoch
Y axis = Training Top-1 accuracy (blue)
         Test     Top-1 accuracy (red)
         Test     Top-5 accuracy (green)
```

PART C (4 points):

In this part, you will do something close to what the human brain does.
The human left and right hemisphere process the same visual input with less accuracy (right) and more accuracy (left). The former is faster and is used to react quickly to events (e.g.., a tiger attacking you) and the latter is slower but more accurate for analytical processing (e.g., calculating a precise math result).

You will use this methodology to classify the CIFAR-100 images from two different aspects, using the same features generated by EfficientNet-B3.

Freeze your model, unplug the last layer of the classifier and plug in two different <u>parallel</u> final (FC: fully-connected) layers.

- First FC final layer will output 20 classes, which are the super-classes of CIFAR-100
- Second FC final layer will output 100 classes, which are the sub-classes of CIFAR-100
- Notice that you can use the preprocessed tensors from PART B. You will be independently training the two new final layers using the same PP tensors.
- Call these two new FC layers SuperFinal and SubFinal.
- Using the PP tensors, run 30 epochs on SuperFinal and 30 epochs on SubFinal. They will be two completely independent training sessions.
- Plot your Top-1 training and test accuracies for SuperFinal and Sub-Final on two separate plots.

PART D (4 points):

Plug these two trained parallel layers back into EfficientNet-B3 and call this new model BrainNet.
BrainNet now has all of the frozen stuff, which feeds the data into two parallel final layers, which output two separate outputs (first 20 outputs, next 100 outputs).

Find 100 completely separate images from a different source (cannot be CIFAR).
They must be images that somehow have similar sub-classes and super-classes as CIFAR-100. Otherwise BrainNet will produce garbage results.

Evaluate these 100 images through your BrainNet. Of course, you must properly resize them and transpose them.

For each image BrainNet will produce a super-class as the output of SuperFinal.
It will also produce a sub-class as the output of SubFinal.
In other words, BrainNet will produce two independent classification outputs for each image.

Report your results with a 5-column table, where the columns are:

1) Original label of the image
2) Super-class reported by BrainNet
3) Super-class confidence (at what percentage did it report it)
4) Sub-class reported by BrainNet
5) Sub-class confidence (at what percentage did it report it)

In your report, clearly describe where you got these additional images.