Assignment 3 - Convolutional Neural Networks (CNNs) for Image Classification (20 marks)

Available: 29-April-2022.

Due: 11.55PM, 20-May-2022.

Instructions:

1. All code should be in Python (>=3.7.x). You should write appropriate comments through the code.
2. You can use *Tensorflow* 2.6 and other necessary libraries.
3. Late submission penalty follows Monash policy.
4. Submission is to be made only on Moodle.
5. Plagiarism cases will be dealt following Monash policy.

Submission: You need to submit a single zip file containing the **Jupyter Notebook** file, other resources used by this Jupyter Notebook, and a report (**Report.pdf**). The Jupyter Notebook file and the report must clearly identify both your full name and student ID. The zip file should be named "A3_<YourMonashID>.zip" (e.g. A3_12345678.zip).

Objective:

Build an image classification system with CNNs using *Tensorflow*. The system will be trained and tested using the CIFAR100 dataset. This CIFAR-100 dataset consists of 100 classes with a total of 60,000 images, where 50,000 images are used for training and 10,000 images are used for testing. The dataset can be accessed using *Tensorflow*.

```
cifar100 = tf.keras.datasets.cifar100
(train_images, train_labels), (test_images, test_labels) = cifar100.load_data()
```

Tasks:

Task 1 - Baseline CNN model (5 Marks)

Build the following architecture using Tensorflow, and display the model summary.

Layer	Hyperparameters	Output Shape	No. of Parameters
Input	/	(32, 32, 3)	0
Conv2D	No. Kernels: 16 Kernel size: 7x7 Stride: 1 Padding: 0 Activation: ReLU	(26, 26, 16)	2368

MaxPooling2D	Pool size: 2 Stride: 2	(13, 13, 16)	0
Conv2D	No. Kernels: 32 Kernel size: 5x5 Stride: 1 Padding: 0 Activation: ReLU	(9, 9, 32)	12832
MaxPooling2D	Pool size: 2 Stride: 2	(4, 4, 32)	0
Flatten	/	(512)	0
Dense	Neuron: 128 Activation: ReLU	(128)	65664
Dense	Neuron: 100 Activation: No	(100)	12900
Total number of parameters			93764

Train and evaluate this model on the dataset. Display the **top-1** accuracy.

Hint: The top-1 accuracy on the test set of the baseline model trained by the tutor is 0.3144.

Task 2 - Improve the baseline model (12 marks)

You will need to improve the baseline model. You will need to consider all the following:

- Data normalization
- Data augmentation
- Modifications of the baseline model (e.g., change filter size, increase the number of filters, add more layers, residual blocks, inception modules, etc.)
- Normalization layers

  You may want to look at the below references:

  https://www.tensorflow.org/guide/keras/preprocessing_layers

  https://www.tensorflow.org/api_docs/python/tf/keras/layers/Normalization

  https://www.tensorflow.org/tutorials/images/data_augmentation

  https://www.tensorflow.org/api_docs/python/tf/keras/layers/BatchNormalization

  https://www.tensorflow.org/addons/tutorials/layers_normalizations

For this task, your report should provide a short description of **each approach** you use, the accuracy of each approach, and the best accuracy you can achieve with your improved model. You should try **any** combinations of approaches for the best accuracy. Present your findings from your experiments.

For this task, you should submit the code of your **best model**.

Task 3 - Explore image classification (3 marks)

In addition to the above approaches, in this task you implement other approaches to improve accuracy on the test set. They can be ideas from papers you read or your own ideas. One potential approach is to use attention mechanisms.

For this task, your report should provide a **description** of each approach you implement and the **performance** of your model. The code for the **implementation** should be submitted.

More assignment instructions:

1. No change is going to be made to the code at our end. Make sure that your code works, when the zip file is unzipped.
2. The runtime of the code for **Task 1 and Task 2** should be less than **30** minutes with the tutor's machine (Quadro A6000). You can use Google Colab if necessary.