Assessment 2: Neural networks for image classification

Generative AI tools are restricted for this assessment task

In this assessment, you can use the following generative artificial intelligence (AI) only- [insert names of AI tools, or types of tools (e.g. image generators/text generators)]. Any use of generative AI must be appropriately acknowledged (see Learn HQ)

After you have read this information, head over to the Assessment 2 Q&A discussion in Ed to ask any questions and see what your peers are saying about this assessment.

Assessment overview

In this assessment, you will do a coding assignment and then answer a series of questions designed to test your understanding of the concepts covered in modules 3 and 4. There are 4 steps that you will complete using Jupyter Notebook and five questions you will need to answer.

This assessment supports unit learning outcomes 3 and 4.

Assessment details

Part 1: Coding assignment (20 marks)

Please organise your code as a Jupyter Notebook .ipynb file.

Objective:

Build an image classification system with CNNs using Tensorflow. The system will be trained and tested using the CIFAR-100 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
		(32, 32,	

Input	/	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 normalisation
- 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.)
- Normalisation layers

You may want to look at the following TensorFlow references:

- Working with preprocessing layers (TensorFlow 2022).
- tf.keras.layers.Normalization (TensorFlow 2022).
- Data augmentation (TensorFlow 2022).
- tf.keras.layers.BatchNormalization (TensorFlow 2022).
- Normalizations (TensorFlow 2022).

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** combination 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.

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.

Note: The runtime of the code for **Task 1** and **Task 2** should be less than **30** minutes with the tutor's machine. You can use Google Colab if necessary.

Appendix - Build your environment.

# Windows (CUDA)
Prepare the WSL2 with CUDA: CUDA on WSL User Guide (NVIDA 2023).
Then follow the instructions for Linux (CUDA).

# Windows (CPU)

```
conda create --name ito5221 python=3.10 ipykernel scikit-image
conda activate ito5221
pip install tensorflow==2.14.0 tensorflow-hub keras==2.14.0
python -m ipykernel install --user --name ito5221
```

# MacOS (CPU)

```
conda create --name ito5221 python=3.10 ipykernel scikit-image
conda activate ito5221
pip install tensorflow==2.14.0 tensorflow_hub keras==2.14.0
python -m ipykernel install --user --name ito5221
```

# Linux (CUDA)

```
conda create --name ito5221 python=3.10 tensorflow-gpu=2.14.0 cudatoolkit=11.8 tensorflow-hub keras==2.14.0 ipykernel scikit-image -c conda-forge
conda activate ito5221
python -m ipykernel install --user --name ito5221
```

# Linux (CPU)

```
conda create --name ito5221 python=3.10 tensorflow=2.14.0 tensorflow-hub keras ipykernel scikit-image -c conda-forge
conda activate ito5221
python -m ipykernel install --user --name ito5221
```

# Google Colab

You don’t need to do anything as everything is out-of-box.

Part 2: Questions (10 marks)

Each question has a maximum word count indicated. Keep your answers concise and appropriate to the question. See individual questions for word guides.

Question 1

Why are activation functions in neural networks necessary?**[1 mark]**

(maximum of 100 words)

Question 2

Please answer following questions related to CNNs:

1. Why do CNNs perform better in image classification tasks than fully connected neural networks? **[0.5 marks]**
   (maximum of 100 words)
2. What's the functionality for fully connected layers and convolution layers? **[0.5 marks]**
   (maximum of 100 words)

Question 3

Please list at least 2 methods to reduce overfitting. Write your answer below.**[1 mark]**

(maximum of 100 words)

Question 4

For object detection, compared with R-CNN, what is the advantage of Fast R-CNN? **[2 marks]**

(maximum of 100 words)

Question 5

You are designing a CNN-based semantic segmentation system, which takes into input an RGB image and generates an output, where each input image's pixel is given a label. The data is satellite images of landscapes such as forests, dessert, and lakes. The four classes of pixels are trees, water, land, and buildings. In total there are 1000 images, out of which 600 can be used for training. The input resolution is 512x512 and the dataset is class balanced. Please answer the following with respect to your system design.

- 3.1. What’s your idea about designing the model architecture for this task? You can use text or drawing scratch to describe briefly.
  (maximum of 1000 words)
- 3.2. On testing the system, it is observed that for some images the pixels on the boundary of buildings and trees are getting confused with each other. How can this phenomenon be improved? In other words how to disambiguate tree pixels from building pixels and vice-versa at the boundary between buildings and trees in an image? **[2 marks]**
  (maximum of 100 words)

Supporting resources

- Math and equations (Jupyter Book, 2023).
- Guidelines for writing in mathematics (PDF 312 KB).
- Citing and referencing: APA 7th (Monash University Library, 2022).

Monash University Library has further information about referencing on their Citing and referencing page.

Submission details overview

Your submission should consist of at least 1 file (i.e. the .pdf file):

- A text-based .pdf file named: FamilyName_StudentId_ITO5221_assignment2.pdf
- The .ipynb Jupyter Notebook file will be named as: FamilyName_StudentId_ITO5221_assignment2.ipynb
    - The Jupyter Notebook .ipynb files are only used for coding questions. The description and output of the code will be answered both in the .pdf and the .ipynb files.
- Submit a single zip file named: FamilyName_StudentId_ITO5221_assignment2.zip. The zip file should contain all related files including Jupyter notebook, report and any other related files.
- Submission is to be made in Moodle.

The text-based .pdf file will undergo a similarity check by Turnitin at the time you submit to Moodle. Please allow a 24-hour turnaround for an originality report to be generated on your PDF.

Submission status

Your assessment MUST show a status of '**Submitted for**

grading' before it will be marked.

If your submission still shows a status of '**Draft (not submitted)**', it will **not** be assessed and will incur late penalties after the due date/time.

Other Notes and Instructions:

1. All code should be in Python (>=3.7.x). You should write appropriate comments through the code. You can use Tensorflow 2.6 and other necessary libraries. 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. Late submission penalty: 3 marks per day.
3. Please do not post part of a proposed partial solution to a forum or other public locations.
4. Appropriate references need to be included in the report.
5. Your .pdf should typically cross-reference any corresponding answer in your Python .py files. Without clear cross-reference between .pdf and .py, it is possible that any such exercise will be awarded 0 marks.
6. All of your submitted work should be and must be in machine-readable form, and none of your submitted work should be hand-written – with all cases of handwritten work possibly resulting in 0 marks.
7. Your solution can be based on the topics from any module (modules 1-4), unless specified.

8. Please make sure your .ipynb file is less than/equal 10MB and .pdf file is less than/equal 10MB.
9. An incorrect filename will incur a 10% penalty.
10. Submitting the Python (.py) file rather than the Jupyter Notebook (.ipynb) file as the program entrance will incure a 10% penalty. Using .py Python modules for import is allowed.

Assessment criteria

Your work will be assessed using the following marking guide:

Assessment 2 marking guide

Item	Marks
Part 1 - Task 1	5
Part 1 - Task 2	12
Part 1 - Task 3	3
Part 2 • Marks are awarded for correct answers.	10
Total	30

References

Jupyter Book. (2023). Math and equations. *Jupyter Book*. https://jupyterbook.org/content/math.html

Monash University Library. (2022). *Citing and Referencing: APA 7th*. https://guides.lib.monash.edu/citing-referencing/apa7th

TensorFlow. (2022). Working with preprocessing layers. *TensorFlow Core*. https://www.tensorflow.org/guide/keras/preprocessing_layers

TensorFlow. (2022). tf.keras.layers.Normalization. *TensorFlow Core*. https://www.tensorflow.org/api_docs/python/tf/keras/layers/Normalization

TensorFlow. (2022). Data augmentation. *TensorFlow Core*. https://www.tensorflow.org/tutorials/images/data_augmentation

TensorFlow. (2022). tf.keras.layers.BatchNormalization. *TensorFlow Core*. https://www.tensorflow.org/api_docs/python/tf/keras/layers/BatchNormalization

TensorFlow. (2022). Normalizations. *TensorFlow Core*. https://www.tensorflow.org/addons/tutorials/layers_normalizations

Assessment declaration and statement of authorship

By submitting my assessments I declare that:

- This is an original piece of work and no part has been completed by any other person.
- I have read and understood the guidelines on the Assessment and academic integrity policy, and no part of this work has been copied or paraphrased from any other source except where this has been clearly acknowledged in the body of the assessment and included in the reference list.
- I have retained a copy of this assessment in the event of it becoming lost or damaged.

I agree and acknowledge that:

- I have read and understood the declaration and statement of authorship.
- I accept that use of my Monash Online account to electronically submit this assessment constitutes my agreement to the Monash Online Assessment Declaration.
- If I do not agree to the Monash Online Assessment Declaration in this context, the outcome of my assessment may not be valid for assessment purposes and may not be included in my aggregate score for this unit.
- I am aware that it is not acceptable to resubmit the same piece of work (in part or as whole) for multiple assessments without permission from the Unit Coordinator.

Before submitting your assessment, please refer to the Academic integrity Monash Library modules to ensure the integrity of your submission.

Further information relating to the penalties for plagiarism, which range from a formal caution to expulsion from the University, is contained on the Academic integrity and what plagiarism, collusion and cheating mean page.