FIT5221 – Assignment 3

This assignment focuses on practicing advanced model architectures and their applications in semantic segmentation. The objective of this assignment is to get a deeper understanding and hands-on practice of two models: Fully Convolutional Network (FCN) and Vision Transformer (ViT).

Available: 06-May-2024
Submission due: 11.55 PM, 30-May-2024

Instructions:

1. All code should be in Python (>=3.7.x). You should write appropriate comments through the code.
2. Late submission penalty: 2 marks per day.
3. Submission is to be made only on Moodle via Moodle Assignment Submission.
4. You must maintain academic integrity. Plagiarism cases will be dealt in accordance with the Monash policy.

Submission:

You need to submit **a single ZIP file**, named **A3_<YourMonashID>.zip** (e.g. A3_12345678.zip). The ZIP file should contain:

- Two Jupyter notebooks with answers to the tasks of the assignment and any extra files to complete your assignment. You should name them following the format in the provided notebook files.
- A report file, named **Report_A3_<StudentName_MonashID>.pdf**
- The required checkpoint files.

Task 1 – Build a baseline FCN model for semantic segmentation (5 marks)

In this task, you are expected to build a semantic segmentation system with FCN architectures. This system will be trained and evaluated on the PASCAL VOC 2012 dataset. This dataset consists of 20 object classes (+1 for background) and is divided into a training set with 1464 images and a validation set with 1449 images. We provide the code to load and process this dataset in the notebook file named "**A3_FCN_StudentName_ID.ipynb**".

Build a **baseline FCN model**:

Layer	**Hyperparameter**	**Output shape**	**Param #**
Input	/	(224, 224, 3)	0
EfficientNetB0 as a backbone	/	(7, 7, 1280)	4,049,571
Conv2D	No. Kernels: 21 Kernel size: 1x1 Stride: 1 Padding: “valid” Activation: ReLU	(7, 7, 21)	26,901
TransposeConv2D	No. Kernels: 21 Kernel size: 64x64 Stride: 32 Padding: “same”	(224, 224, 21)	1,806,336
Total number of params			5,882,808

You are required to:

- **Print out the summary** of this model
- **Train and evaluate** this model on the PASCAL VOC 2012 dataset.
- **Plot the accuracy, loss value, and MeanIoU score** of both the training set and the validation set across epochs.
- **Write a report**, in which you should mention the optimizer, the loss function, and the hyperparameters used to train the model.

Note:

- For the backbone EfficientNetB0, you could use the model from the Keras library and change its parameters (such as “include_top”) according to your needs. You might want to use the backbone with pre-trained weights on ImageNet for better performance. You should check the reference for more details.

- If you want to implement the backbone EfficientNetB0 from scratch instead of using the library, make sure the number of parameters and output shape match the description above.

- For the MeanIoU score, we provided the "*metric_mean_iou(model, dataset)*" function in the notebook. You should call this function to evaluate the training set and validation set after each training epoch. We suggest you use the *callback* function of TensorFlow.

- We achieved 0.71 MeanIoU on the training set and 0.30 MeanIoU on the validation set with a pre-trained backbone on ImageNet. These scores are for your information only. You are not expected to achieve the same numbers on this task.

Reference

- EfficientNetB0 model from Keras library: https://keras.io/api/applications/efficientnet/

Task 2: Improve the baseline FCN model (9 marks)

In this task, you are expected to improve the performance of the baseline FCN model described in Task 1 by using multi-scale features.

- You need to implement at least one approach from papers you read or your ideas. For any approaches you applied to the baseline FCN, make sure you use the same backbone as the baseline model in Task 1.

- The number of parameters of your model in this task should not be more than 20 million. Marks might be deducted if it is more than 20 million.

- Your report should provide a **description, references** for each approach you implement, and the **performance** of your model. The code for the **implementation** should be submitted along with a checkpoint.

Mark allocation for this task:

- Implementation and report: **6 marks**
- MeanIoU on the validation set:
  - `MeanIoU ≥ 0.37` **3 marks**
  - `0.37 > MeanIoU ≥ 0.34` **2 marks**
  - `0.34 > MeanIoU ≥ 0.31` **1 mark**

Hint: You may want to take a look at some reference papers for ideas about multi-scale features.

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", MICCAI, 2015.

[3] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path Refinement Networks for High-Resolution Semantic Segmentation", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[4] T. -Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

Task 3: Implement the MultiHeadAttention layer from scratch (4 marks)

In this task, you are provided a codebase that implements a part of ViT model. You are required to implement the MultiHeadAttention module on your own, plug it into the ViT model, and train it on the CIFAR-100 dataset.

The codebase is provided in a notebook named "**A3_VIT_StudentName_ID.ipynb**" with a set of hyperparameters. You need to train the ViT model with this set of hyperparameters using your implemented MultiHeadAttention module. The necessary condition to get full marks is that the accuracy on the test set of CIFAR-100 should be **at least 45%**.

You are also required to:

- Print out the model summary
- Print out the training progress of your best model in a notebook cell's output.
- Plot the accuracy and loss value for both the train set and test set across epochs.
- Write a report explaining key steps of your implementation.

Reference

[1] Custom layers with TensorFlow. (link)

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. "Attention Is All You Need". The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS), 2017.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". The International Conference on Learning Representations (ICLR), 2021.

Task 4: Run a baseline ViT model (2 marks)

Using the code for the ViT model in task 3, you are required to conduct experiments to evaluate the impact of hyperparameters **patch_size**, **num_heads**, **projection_dim**, and **transformer_layers** of the ViT model on the train and test accuracies. You need to report:

1. Your findings from the experiments by adjusting the mentioned hyperparameters.
2. The hyperparameter values, train and test accuracies of the model with the highest test accuracy among all models you report.

– END OF ASSIGNMENT –