The University of Melbourne
School of Computing and Information Systems
COMP90086 Computer Vision, 2023 Semester 2

Assignment 2: Feature engineering with convolutional neural networks

Due:	7pm, 8 Sept 2023
Submission:	Source code (in Jupyter Notebook) and written responses (as .pdf)
Marks:	The assignment will be marked out of 7 points, and will contribute 7% of your total mark.

In this assignment, you will use CNNs to classify a scene categorization dataset, `scene32`. This dataset is based on a subset of a dataset created by Aude Oliva and Antonio Torralba[1], and includes 8 categories of everyday scenes such as "forest," "coast," and "street." Images have been downsampled to 32 × 32 pixels. We have provided a train/validation/test split with 180/40/40 images per class. **Please use our provided train/validation/test split throughout this assignment.**

1. CNN training [2 pt]

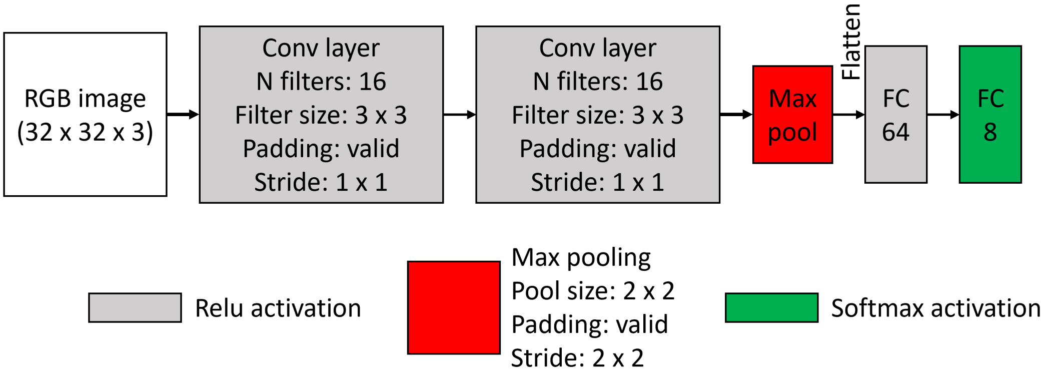
RGB image (32 x 32 x 3)
Conv layer N filters: 16 Filter size: 3 x 3 Padding: valid Stride: 1 x 1
Conv layer N filters: 16 Filter size: 3 x 3 Padding: valid Stride: 1 x 1
Max pool
Flatten
FC 64
FC 8
Relu activation
Max pooling Pool size: 2 x 2 Padding: valid Stride: 2 x 2
Softmax activation

Figure 1: Network diagram

Implement the CNN architecture shown above in Figure 1. Use ReLU activation functions for all layers except the final layer, which should use the Softmax activation function. Use the Adam optimiser and `categorical_crossentropy` loss.

Train this on the `scene32` dataset. Your training process should include some form of data augmentation – it is up to you to decide which augmentations are suitable for this dataset.

In your written report, briefly explain and justify your implementation (including your data augmentation scheme, and any other implementation choices you made, such as the number of epochs to train or when to stop training). Your report should include plots of the training and validation loss/accuracy over training epochs. State the accuracy of your final trained network on the test set.

[1] Downloaded from `https://people.csail.mit.edu/torralba/code/spatialenvelope/`

2. Error analysis [2 pt]

Evaluate your network from Question 1 on the scene32 test set. (Note: whatever data augmentations you added for Question 1 *should not* be used on the test set.) In your written report, present the overall classification accuracy and the average accuracy for each of the 8 classes. Explain the performance of the CNN model, using example images from the test set to illustrate your discussion. What classes/images were difficult for this model, and why?

3. Kernel engineering [3 pt]

Modify the network from Question 1 so that instead of two convolutional layers, there is only one convolutional layer with kernel size $K \times K$. Keep all other convolutional layer parameters the same.

Train and test this modified network on the scene32 dataset using various values for the kernel size K. How does classification performance change as you increase kernel size and why? In your written report, include visualisations of the kernels learned by the convolutional layer to support your discussion.

Submission

You should make two submissions on the LMS: your code and a short written report explaining your method and results. The response to each question should be no more than 400 words.

Submission will be made via the Canvas LMS. Please submit your code and written report separately under the **Assignment 2: Code** and the **Assignment 2: Report** links on Canvas.

- Your **code** submission should include the Jupyter Notebook (please use the provided template) with your code and any additional files we will need to run your code, if any (do not include the scene32 dataset).
- Your written **report** should be a .pdf with your answers to each of the questions. The report should address the questions posed in this assignment and include any images, diagrams, or tables required by the question.

Evaluation

Your submission will be marked on the correctness of your code/method, including the quality and efficiency of your code and the appropriateness of design decisions. You should use built-in Python functions where appropriate and use descriptive variable names. Your written report should clearly explain your approach and any experimentation used to produce your results, and include all of the specific outputs required by the question (e.g., images, diagrams, tables, or responses to sub-questions).

Late submission

The submission mechanism will stay open for one week after the submission deadline. Late submissions will be penalised at 10% of the total possible mark per 24-hour period after the original deadline. Submissions will be closed 7 days (168 hours) after the published assignment deadline, and no further submissions will be accepted after this point.

Updates to the assignment specifications

If any changes or clarifications are made to the project specification, these will be posted on the LMS.

Academic misconduct

You are welcome — indeed encouraged — to collaborate with your peers in terms of the conceptualisation and framing of the problem. For example, we encourage you to discuss what the assignment specification is asking you to do, or what you would need to implement to be able to respond to a question.

However, sharing materials — for example, showing other students your code or colluding in writing responses to questions — or plagiarising existing code or material will be considered cheating. Your submission must be your own original, individual work. We will invoke University's Academic Misconduct policy (`http://academichonesty.unimelb.edu.au/policy.html`) where inappropriate levels of plagiarism or collusion are deemed to have taken place.