COMP90086 – Computer Vision – Assignment 2

Arya Araban – 1439683 – aaraban@student.unimelb.edu.au

---

1. CNN Training:

I implemented the CNN architecture exactly as described in the assignment specification. For data augmentation, I used rotations, width/height shifts, horizontal flips, and zooms. Rotations help prevent overfitting to upright orientations, while shifts introduce robustness to position. Flips augment the dataset size, and zooms induce invariance to scale. This augments the diversity of data to improve generalization.

To further prevent overfitting, early stopping with a patience of 4 epochs was implemented. This mechanism halts the training if the validation accuracy does not improve for 4 consecutive epochs, after which the weights are restored to the best-performing state achieved so far.

The accuracy and loss plots presented in Figure 1 indicate successful model training using augmented training data, with no notable signs of overfitting. The model was initially configured to train for a total of 30 epochs. However, due to early stopping, training concluded on the 15th epoch while the weights obtained at the 11th epoch were restored. This approach resulted in a final validation accuracy of 70.1%, highlighting a rather strong performance.

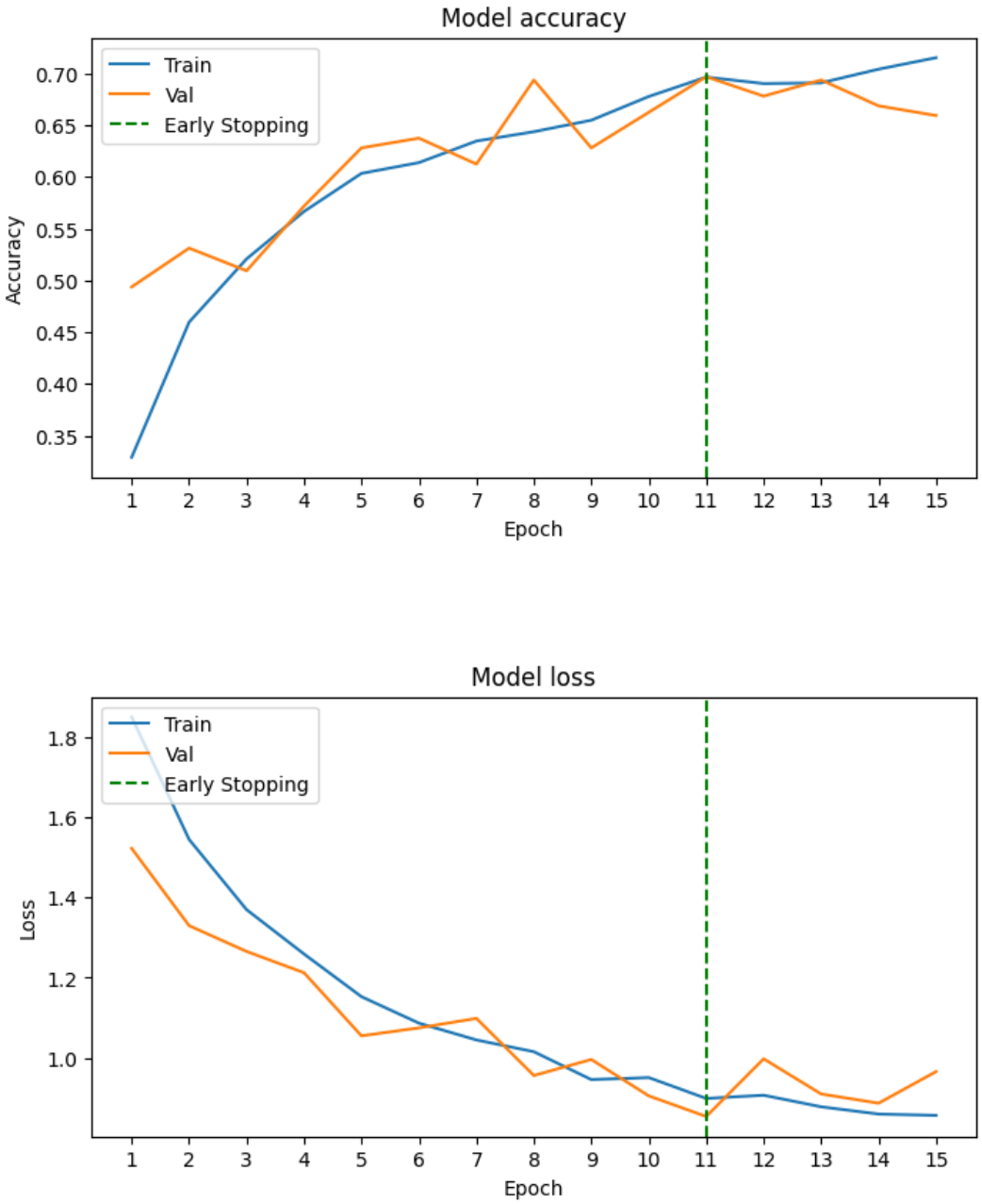

Figure 1 - Training and Validation Accuracy/Loss

To achieve higher accuracy, it may be important to employ effective regularization techniques and consider expanding the dataset. Nevertheless, our current implementation provides a strong basis for scene classification.

2. Error Analysis:

The CNN model achieved an overall test accuracy of 76.56% on the scene32 dataset. Additionally, we compute per-class accuracy as the percentage of ground truth images in each class that were correctly classified. Specific accuracy values for each category can be found in Figure 2.

```
Overall accuracy: 76.56%
Class 0 (coast) accuracy: 70.00%
Class 1 (forest) accuracy: 80.00%
Class 2 (highway) accuracy: 77.50%
Class 3 (insidecity) accuracy: 75.00%
Class 4 (mountain) accuracy: 62.50%
Class 5 (opencountry) accuracy: 65.00%
Class 6 (street) accuracy: 90.00%
Class 7 (tallbuilding) accuracy: 92.50%
```

Figure 2 - Overall & Per-Class Accuracies on the scene32 Dataset

The model demonstrated strong performance on classes such as tallbuilding and street, which possess distinct structural features that can be effectively captured by the convolutional filters. For instance, Figure 3 showcases a range of skyscraper clusters from the test set, exhibiting clear straight lines and geometric patterns that are advantageous for the model's learning process.

Figure 3 - Example of Skyscraper Clusters images in the Test Set

However, the model encountered challenges with categories that represent more natural environments, such as the mountain category. Figure 4 exemplifies a range of mountain images from the dataset, which as can be seen, lack prominent edges or recognizable shapes. This absence of distinctive features makes it difficult for the model to accurately differentiate mountain scenes from other outdoor scenes.

Figure 4 - Example of Skyscraper Clusters images in the Test Set

Furthermore, the classification of the opencountry and coast classes also posed a challenge, likely due to their reliance on holistic scene layout and color features rather than textures, as shown in Figure 5.

Figure 5 - Example of Open Country images in the Test Set

To enhance the overall performance of the model, it is likely necessary to incorporate larger and more diverse training data specifically targeting the challenging classes. This would enable the model to learn robust filters that can better capture the unique characteristics of these classes. Additionally, strategies such as incorporating an early pooling layer to capture scene layout could prove beneficial in expanding the range of features extracted by the model. While the CNN exhibits promise, further refinement is needed to effectively handle the diverse range of outdoor categories encountered.

3. Kernel Engineering:

I implemented the modified CNN architecture described in the assignment specification. I trained the model for various kernel sizes [3, 5, 7, 9] over a period of 5 epochs on the non-augmented training set. The decision to limit the training to 5 epochs was made to obtain a preliminary understanding of how the models compare to each other. The number of kernels in each model, regardless of the kernel size, has been fixed at a constant value of 16.

The following table shows test accuracy of the model across varying kernel sizes:

Kernel Size	3x3	5x5	7x7	9x9
Accuracy	60.31%	68.44%	64.69%	59.06%

As observed in the provided table, the results demonstrate a mixed pattern of performance improvement and decline with increasing kernel sizes. Across various tests, I have observed that the shift from a 3x3 kernel to a 5x5 kernel generally shows an improvement in accuracy, suggesting the potential benefits of capturing more spatial information and enhanced feature extraction. However, further increasing the kernel size to 7x7 and 9x9 leads to a decrease in accuracy. This suggests that the advantages of larger kernel sizes, such as increased receptive field and improved translation invariance, may be offset by factors like increased computational complexity, loss of local detail, and reduced spatial resolution. The complex interplay of these factors can make it challenging to determine a clear-cut pattern, indicating the need for careful consideration and experimentation when selecting an optimal kernel size.

Figure 6 displays the visualization of the average kernel (averaging across the values of the 16 kernels) from each of the various different kernel sizes. Since the input image has 3 channels (representing RGB), the kernels also have a channel depth of 3. To improve the visualization, the weight values of the average kernel for each kernel size can be rescaled between 0 and 255. This rescaling allows for a better understanding of the relative magnitudes of the weights within each kernel, as the values are mapped to the full range of grayscale values, where 0 represents black and 255 represents white.

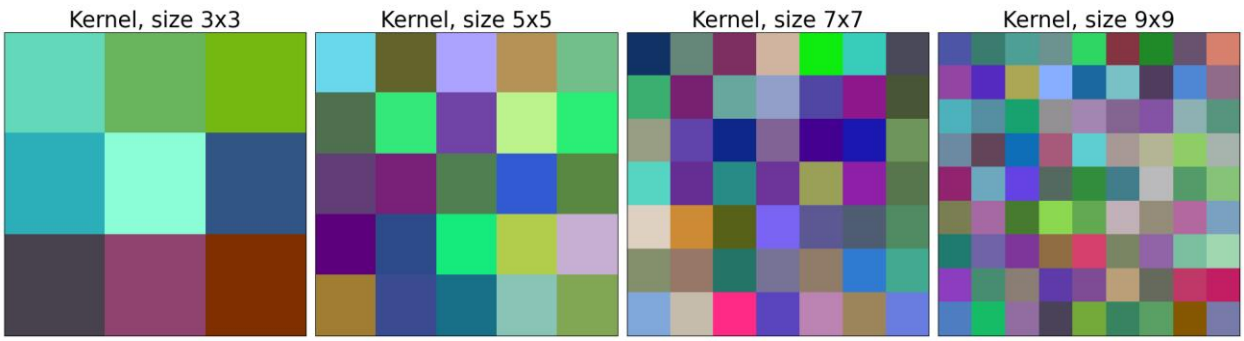

Figure 6 - Visualization of Average Kernel for Different Kernel Sizes

In the analysis of the kernel images displayed in Figure 6, it becomes apparent that it is challenging to discern any clear patterns or distinguishable features across the different filter sizes. This lack of distinguishability can be attributed to the complexity and variability of the learned weights within each kernel. As the kernel size increases, the number of parameters and the level of detail captured by the filters also increase. This complexity makes it difficult to visually interpret the specific characteristics represented by each kernel.

To address this challenge and gain a better understanding of the different filter sizes, we can directly apply these convolutions to one of the test images. By convolving the image with each kernel, we can observe the resulting convolved images and identify the visual effects produced by each kernel, as seen in Figure 7.

Figure 7 – Result of Convolving Average kernel of Different Sizes on Test Image

The analysis of the kernel images and the resulting convolved images in 7 reveals that as the kernel size increases, the convolutions produce more blurred and abstract shapes in the images, indicating a higher level of abstraction and potential for capturing hidden information. Conversely, smaller kernel sizes tend to emphasize edges and local details. These findings suggest a trade-off between capturing fine-grained features and extracting higher-level abstract representations. As seen previously, in my tests, the 5x5 kernel appears to strike a favorable equilibrium between these two aspects.