[redacted]

[redacted]

Feature Engineering with Convolutional Neural Networks

First Name, Last Name	Li Sean, Wong
[redacted]	[redacted]
[redacted]	[redacted]

Question 1: CNN Training

For data augmentation on the training set, I decided to use horizontal flip, rotation and brightness augmentation. I used a lower bound of 0.5 and an upper bound of 1.5 for brightness augmentation. This means that the brightness of an image can be reduced to up to half its original brightness or increased to up to 1.5 times its original brightness. For rotation, I used a range of 15 which means that images can be rotated up to 15 degrees. These augmentations were used as CNNs are not invariant to these augmentations (unlike translation and scale augmentation). They also increase the variability of images that are fed into the model, increasing the performance of the model under these different augmented conditions.

The number of epochs I decided to use is 21. This number was obtained by using early stopping, where the patience level is set to 3. If the validation loss does not improve after 3 epochs, it will cause an early stop.

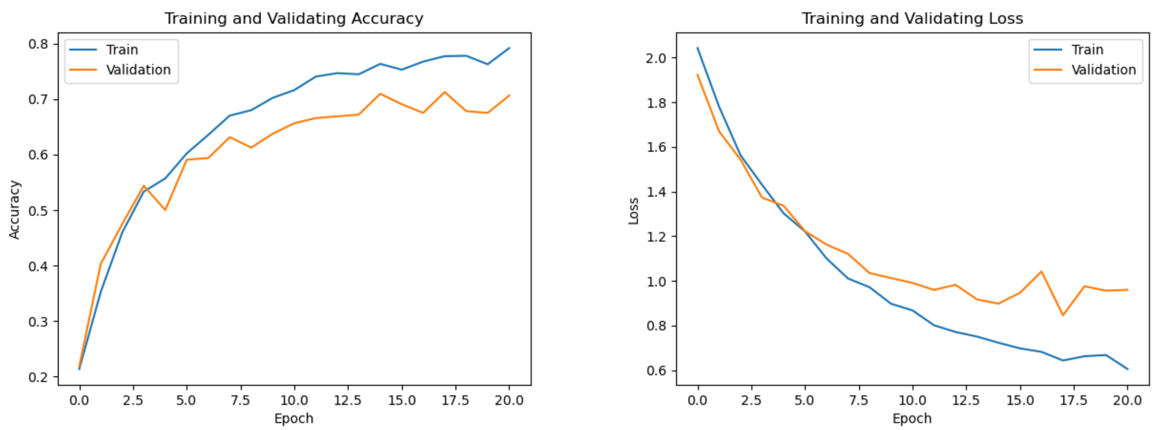

Fig. 1: Training and Validating Accuracy and Loss

The accuracy of my final trained network on the test set is 76.875% (3 d.p).

Question 2: Error Analysis

Class Number	Class Name	Accuracy
0	coast	80%
1	forest	75%
2	highway	82.5%
3	insidecity	67.5%
4	mountain	72.5%
5	opencountry	62.5%
6	street	82.5%
7	tallbuilding	92.5%

Overall Classification Accuracy = 76.875%

Fig. 2: Overall Classification Accuracy and Average Accuracy for Each Class

From figure 2, we can see that my model has the highest accuracy when predicting class 7 (= tallbuilding) and the worst accuracy when predicting class 5 (= opencountry). When looking closer at the classification for class 5, we can see the misclassified images below:

Fig. 3: Misclassified Images for Class 5 (opencountry)

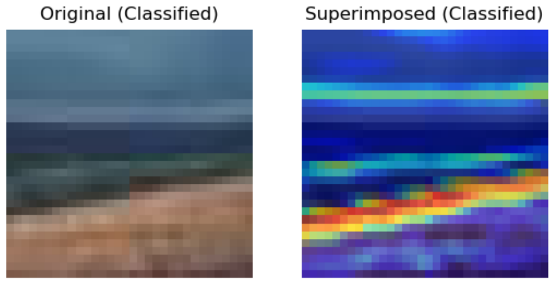

Fig. 4: Correctly Classified Image for Class 5 (opencountry)

Fig. 5: Misclassified Image for Class 5 (opencountry) - Classified as Class 0 (coast)

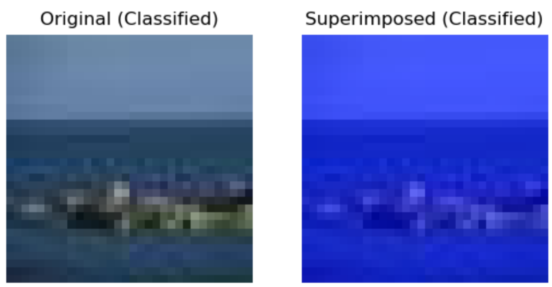

Fig. 6: Correctly Classified Image for Class 0 (coast)

From figure 4, we see that there is a big cluster where the sand meets the ocean. We see that when it was misclassified as “coast”, the clusters from the heatmap of figure 4 are not similar to the one in figure 5. It is more similar to the heatmap from figure 6, where it picks up details from the white area of figure 5 and the rocky area from figure 6. We can also observe that the model is having difficulties when colors across the image are relatively similar (there are only 2 clear distinct colors).

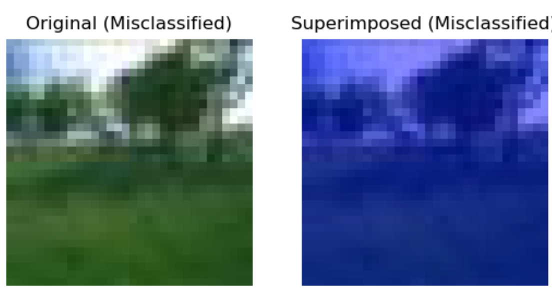

Fig. 7: Misclassified Image for Class 5 (opencountry) - Classified as Class 1 (forest)

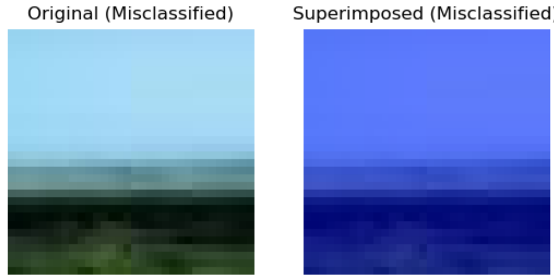

Fig. 8: Misclassified Image for Class 5 (opencountry) - Classified as Class 2 (highway)

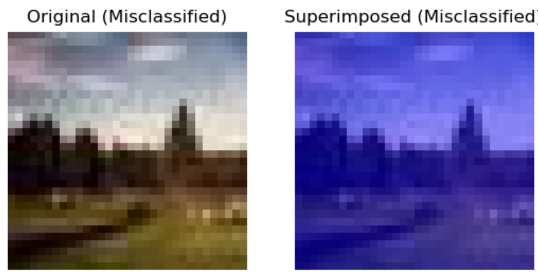

Fig. 9: Misclassified Image for Class 5 (opencountry) - Classified as Class 3 (insidecity)

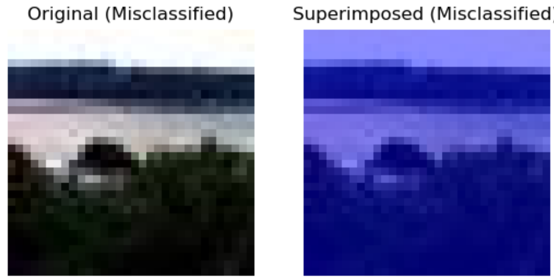

Fig. 10: Misclassified Image for Class 5 (opencountry) - Classified as Class 4 (mountain)

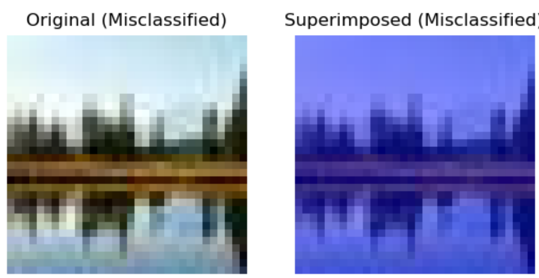

Fig. 11: Misclassified Image for Class 5 (opencountry) - Classified as Class 6 (street)

We can see that across figures 7 to 11 when images have a little color, the model finds it difficult to pick up on details. This could also be due to how the images are also relatively small, giving the model less to work with. Another reason could also be that the model has overfitted to the training set.

Question 3: Error Analysis

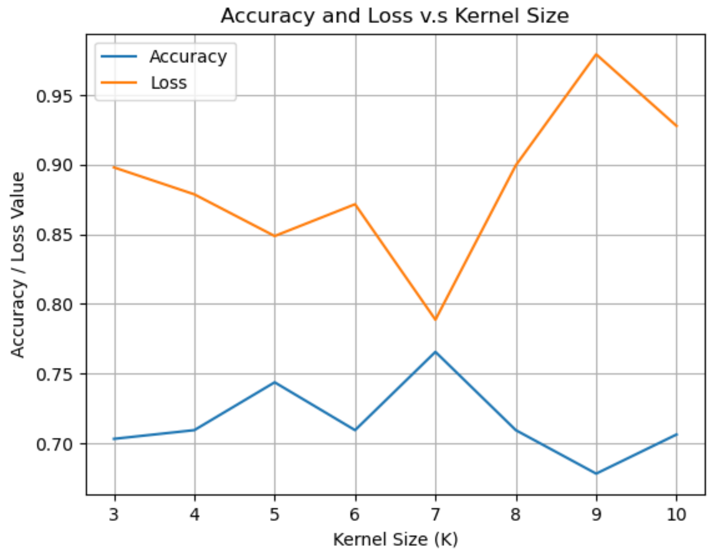

Fig. 12: Accuracy and Loss Value for Different Kernel Sizes

From figure 12, we can observe an increase in classification accuracy from the base kernel size of 3 to the kernel of size 7. After this point, the accuracy decreases and the loss value increases significantly. With a smaller kernel size, the model is able to capture finer details such as edges and/or textures. With larger kernel sizes, the model becomes more complex and is able to capture underlying patterns, which shows in the figure above with the increase in accuracy. However, increasing the kernel size raises the risk of the model overfitting. The model becomes too specialized for the training data and performs poorly when introduced to unseen data. This is most likely the cause of the significant decrease in accuracy after kernel size K = 7.

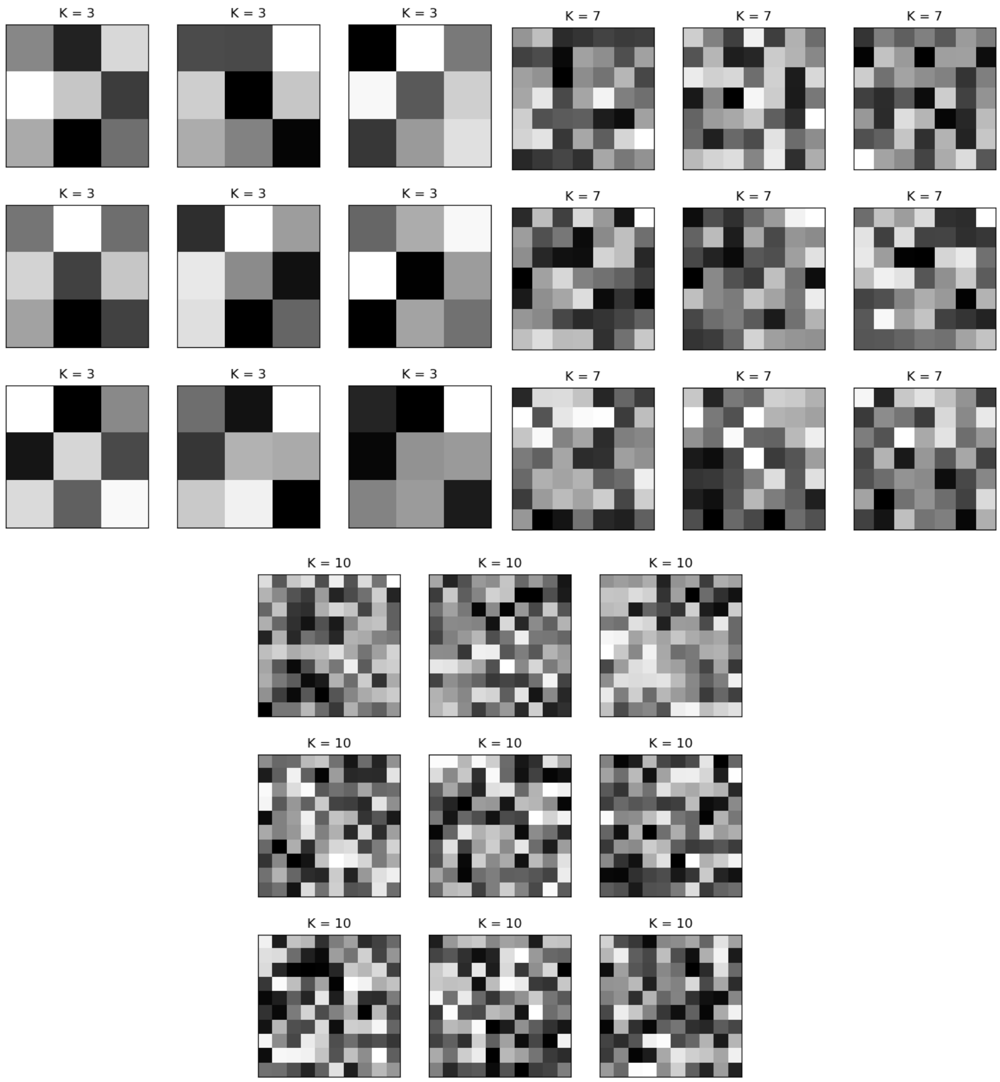

Fig. 13: Kernel Visualization for K = 3, 7 and 10

From figure 13, we can see that for K = 3, the first row of filters are detecting some edges from the left to right (vertical line). With this kernel size, the filters learn smaller local features as mentioned above (edges and/or textures). A disadvantage would be that the filters are too general. This would decrease accuracy as the model could mistakenly identify these patterns in various parts of the image. When looking at K = 7, it is hard for us to understand what the filters are learning. Since the kernel size has increased, the model is learning more global features such as complex patterns. Lastly, looking at K = 10, the model has become more complex. It looks as if it has learned extremely specific features. This causes overfitting, which was made evidently clear from figure 12.