Computer Vision Assignment2 report

Jiaheng Dong
1166436

Part 1. Analysis of CNN training processes and result

By visualizing the size of the training data, there are 1440 images in total for 8 classes, which means there are only 180 images for each class to train the model. In this case, the model might be less able to extract the features of each class and easier to get overfitted. Therefore, the image augmentation is needed to expand the size of the training data. Applying small amounts of variations on the original image does not change the target class but provides a new perspective of capturing the object, which allows the CNN model to generalize better on unseen data.

However, data augmentation techniques should be suited for current dataset, exploiting them could potentially lead the model to predict substandard results. In my implementation, firstly, the random rotation of 20 degrees and fits the blank with nearest pixel value is previously used to prevent if the image is not in the right angle. Secondly, random shift is used in both directions, it overcomes if the object is not in the center of the image. Thirdly, horizontal flip is used to help the model distinguish features if the photo is taken in a symmetric position. Fourthly, the random zoom is used to give more training images which are taken in different distances. The last is the random brightness which is used to expand the training data with different shooting time and different seasons. Besides logically select the augmentation techniques, I used backward elimination based on the overall training accuracy (table 1).

Eliminated augmentation technique	Overall test accuracy
Random shift	0.7536
Random zoom	0.7344
Horizontal reflection	0.7031
Random brighteness	0.7031
Random rotation	0.7156
None	0.7531
All	0.6531

Table 1. overall test accuracy of after elimination

Based on the testing accuracy, all the augmentations have a good performance except the random shift. The main reason could be the testing dataset does not contain many shifted photos, so the prediction accuracy is improved after eliminating the random shift.

Moreover, CNN is prone to overfit for large training epochs or large batch size. In my implementation, batch size is set to 32 which gives a more generalized performance. Besides, 32 is the powers of 2, it highlights that batch size is still a hyperparameter by helping constrain the hyperparameter search space. As for the epochs control, the early stop (figure 1, 2) is used as a callback function to stop the training when the validation loss is not decreasing. The final overall accuracy is 0.7563, loss is 0.7953.

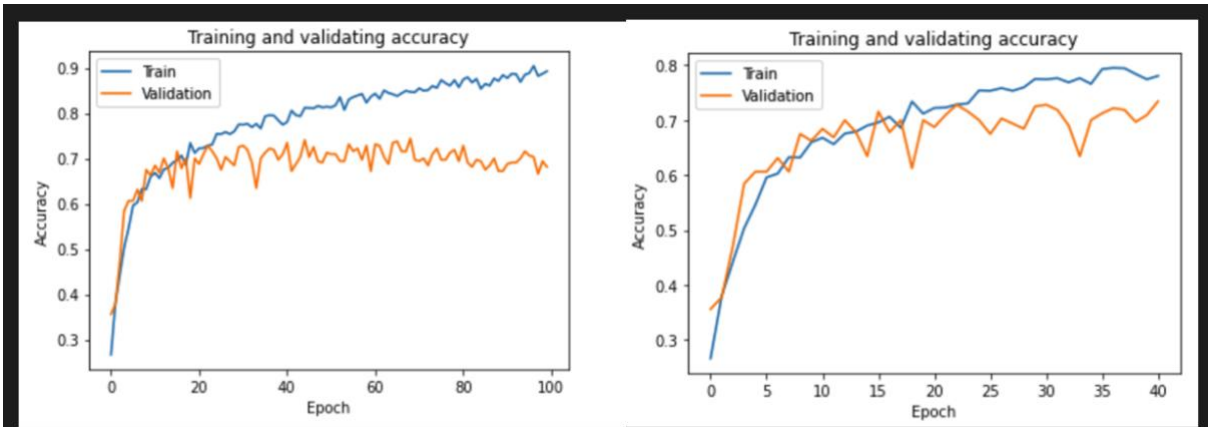

Figure 1. training and validating accuracy before and after using early stop

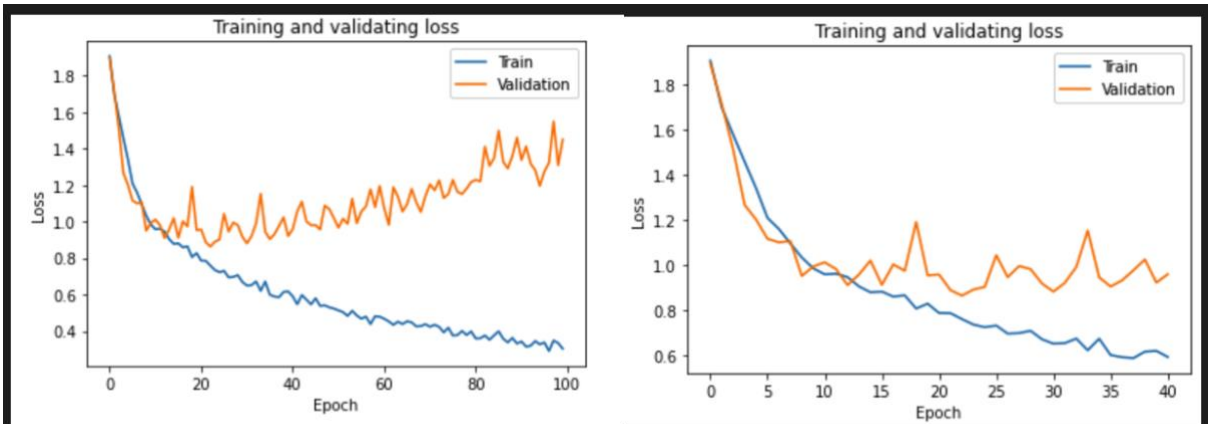

Figure 2. training and validating loss before and after using early stop

Part 2. Error analysis

According to the evaluated scores of each class (figure 3), the model has a moderate ability in predicting “forest”, “mountain” at around 70% accuracy, it has a better prediction in “coast”, “highway”, “opencountry” at 78% accuracy. As for “street” and “tallbuilding”, the model prediction has the highest accuracy at 85% and 97%. Moreover, it has the worst prediction for “insidecity” class at 0.53 recall score.

	precision	recall	f1-score	support
0	0.79	0.78	0.78	40
1	0.91	0.72	0.81	40
2	0.72	0.78	0.75	40
3	0.64	0.53	0.58	40
4	0.74	0.65	0.69	40
5	0.72	0.78	0.75	40
6	0.74	0.85	0.79	40
7	0.80	0.97	0.88	40
accuracy			0.76	320
macro avg	0.76	0.76	0.75	320
weighted avg	0.76	0.76	0.75	320

```
[[31  0  5  0  2  2  0  0]
 [ 0 29  0  4  1  2  3  1]
 [ 0  0 31  5  1  3  0  0]
 [ 1  0  2 21  2  0  6  8]
 [ 3  0  2  1 26  5  2  1]
 [ 4  1  2  0  1 31  1  0]
 [ 0  1  1  2  2  0 34  0]
 [ 0  1  0  0  0  0  0 39]]
```

Figure 3. classification report and confusion matrices by using CNN model on given test data

As for “street” and “tallbuilding” classes, there are three reasons for the model got high accuracy scores. Firstly, the classes themselves have very unique and obvious features, the

correct predicted “street” images (figure 4) all has a straight street in the center of the image which takes a large proportion of the image. The tall building class has angular cuboid structure (figure 5) which takes most of the proportion of the image, and it is unique compared with other classes. Secondly, as for tall building class, the data augmentation is more useful, the light conditions and the zoom range are various in the test data of tall building, so the data augmentation in the training stage gives the CNN model a better generalized performance on unseen images. Thirdly, the extremely high prediction accuracy might be caused by the similarity of the training dataset and test dataset. As for the tall buildings (figure 6), which have strict structure and appearance standers, it is easy to get lots of similar images.

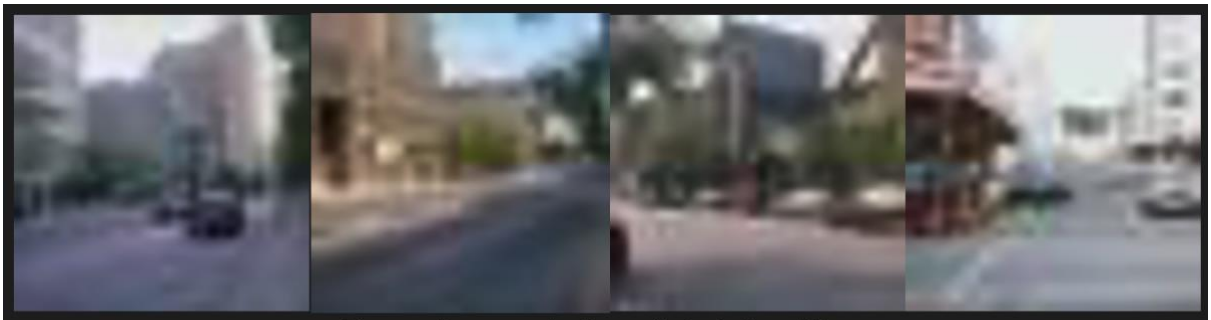

Figure 4. corrected predicted street images

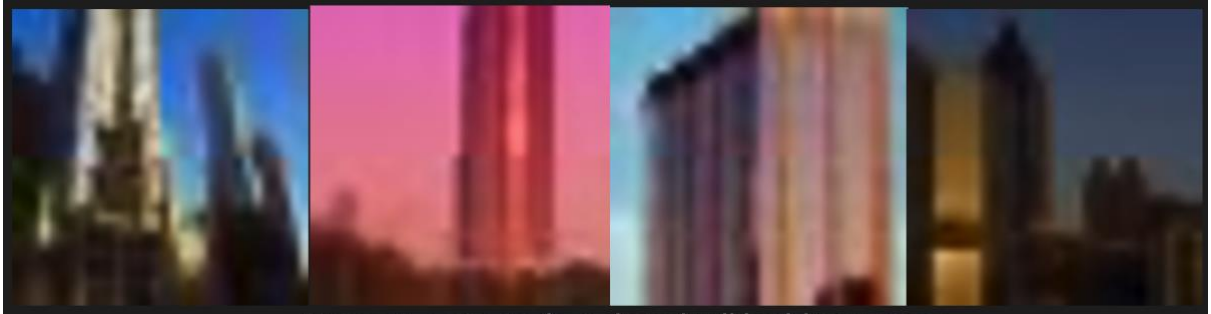

Figure 5. corrected predicted tall building images

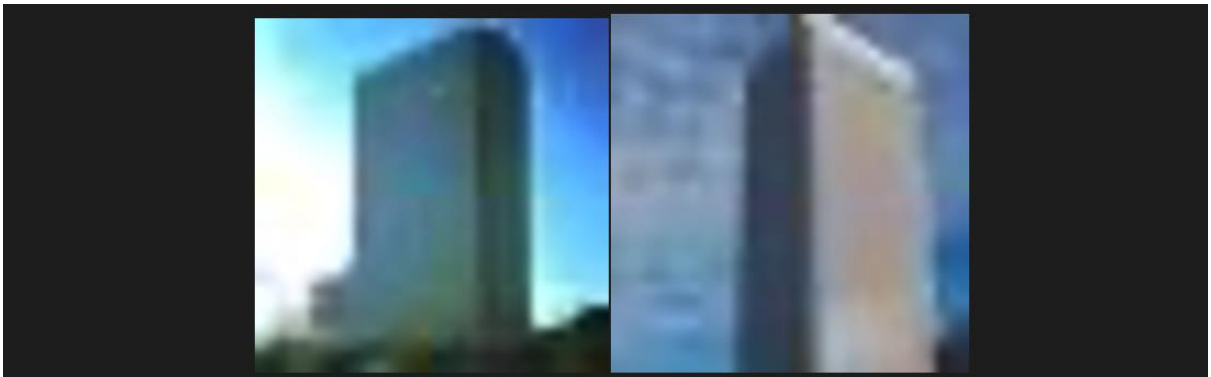

Figure 6. Similar images example of training data (left) and test data (right)

As for the “insidecity” class, the recall, precision and f1 scores are all the lowest, which means the CNN model did not learn enough features to distinguish this class. The main reason could be that the class’s features are not obvious and unique, which might overlap with other classes (figure 7). According to the confusion matrices, the most wrongly predicted results of “inside city” class are “street” and “tall buildings”, which are also the scene inside the city but with more sufficient features.

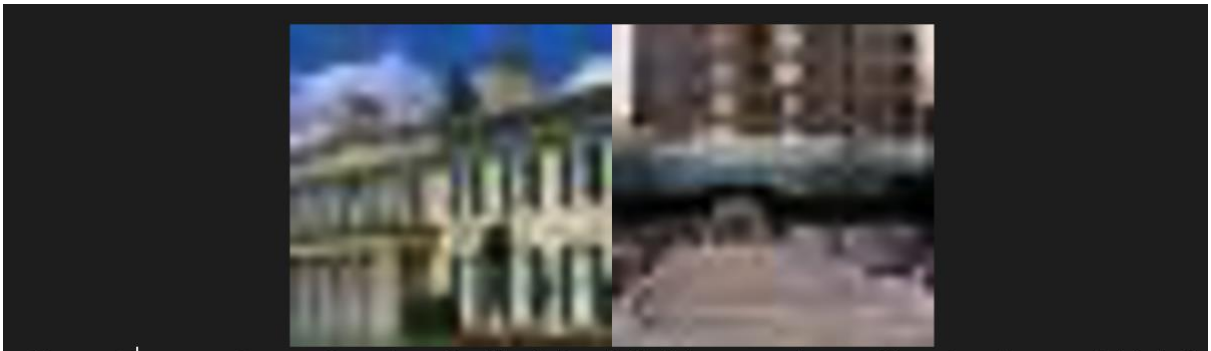

Figure 7. wrongly predicted as “tallbuilding” (left), wrongly predicted as “street” (right)

As for other classes, the evaluating score is reasonable. According to the confusion matrices, the misclassified results are mainly concentrated on one specific class of each class, this is mainly the same reason as the “insidecity” class, there might be some overlapped features. Moreover, it could be the data augmentation problem, some of the augmentation might not suitable for learning the sufficient features of these classes based on the given data.

Part 3. Kernel engineering

Kernel size in CNN model determines the number of learning parameters in the convolutional layer. When the number of convolutional layers decreases to one, the effects of the kernel size changing will be obvious. In my implementation, the kernel size is increasing from 1*1 to 11*11 in odd numbers. The larger size of the kernel size, the more parameters were learnt in the convolutional layer, the accuracy changed as a curve, the extremely small and large will lead to low accuracy scores in predicting the test data (table 2). The training time might be oscillating because of the early stop.

Kernel size	Overall accuracy	Parameters of Con2D layer	Training time
1*1	0.6562	64	65
3*3	0.7625	448	48
5*5	0.7156	1216	45
7*7	0.7469	2368	47
9*9	0.7156	3904	52
11*11	0.6219	5824	56

Table 2. performance of CNN model with different kernel size

As for the smaller sized kernel, each kernel will extract smaller range of pixels which means each kernel could capture detailed information of the class images (figure 8). It might speeds up training (table 2) and makes the model less prone to overfitting. However, if the kernel size is too small, it will not be able to capture global features, especially the images of some class are sparse. As for the larger sized kernel, each kernel has greater receptive field which allows kernel to extract global features (figure 8). However, increasing the kernel size may not improve model performance, it depends on the image size. In our data, the images are all in 32*32, so if a 11*11 kernel size is applied, the CNN model will miss out some important local features. Moreover, it could increase the complexity of model which might lead to overfitting and might slow down the training process (table 2).

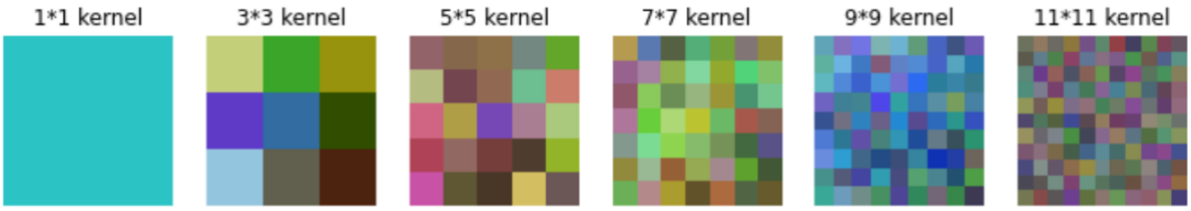

Figure 8. first filter of each model's convolutional layer with RGB color

To balance the kernel size and model performance, the multiple convolutional layers with reasonable small kernels is commonly used. It could not only maintain the ability of capture local features and global features, but also reducing the learning parameters and computation complexity to speeds up the training and prevent from overfitting.