Assignment2 report -Rosie Wang 1806394

1. Training dynamics:

```
Epoch 1, Train Loss: 0.457, Train Accuracy: 84.92%, Val Loss: 2.226, Val Accuracy: 50.96%
Epoch 2, Train Loss: 0.389, Train Accuracy: 87.26%, Val Loss: 2.283, Val Accuracy: 51.72%
Epoch 3, Train Loss: 0.378, Train Accuracy: 87.71%, Val Loss: 2.401, Val Accuracy: 50.78%
Epoch 4, Train Loss: 0.368, Train Accuracy: 87.74%, Val Loss: 2.406, Val Accuracy: 51.38%
Epoch 5, Train Loss: 0.382, Train Accuracy: 87.56%, Val Loss: 2.408, Val Accuracy: 51.64%
Epoch 6, Train Loss: 0.351, Train Accuracy: 88.68%, Val Loss: 2.526, Val Accuracy: 50.82%
Epoch 7, Train Loss: 0.333, Train Accuracy: 89.08%, Val Loss: 2.638, Val Accuracy: 50.18%
Epoch 8, Train Loss: 0.293, Train Accuracy: 90.53%, Val Loss: 2.728, Val Accuracy: 49.42%
Epoch 9, Train Loss: 0.318, Train Accuracy: 89.82%, Val Loss: 2.783, Val Accuracy: 49.34%
Epoch 10, Train Loss: 0.264, Train Accuracy: 91.50%, Val Loss: 2.682, Val Accuracy: 50.42%
Epoch 11, Train Loss: 0.245, Train Accuracy: 92.12%, Val Loss: 2.795, Val Accuracy: 50.96%
Epoch 12, Train Loss: 0.251, Train Accuracy: 92.11%, Val Loss: 3.030, Val Accuracy: 49.10%
Finished Training
```

a. Loss and Accuracy dynamics:

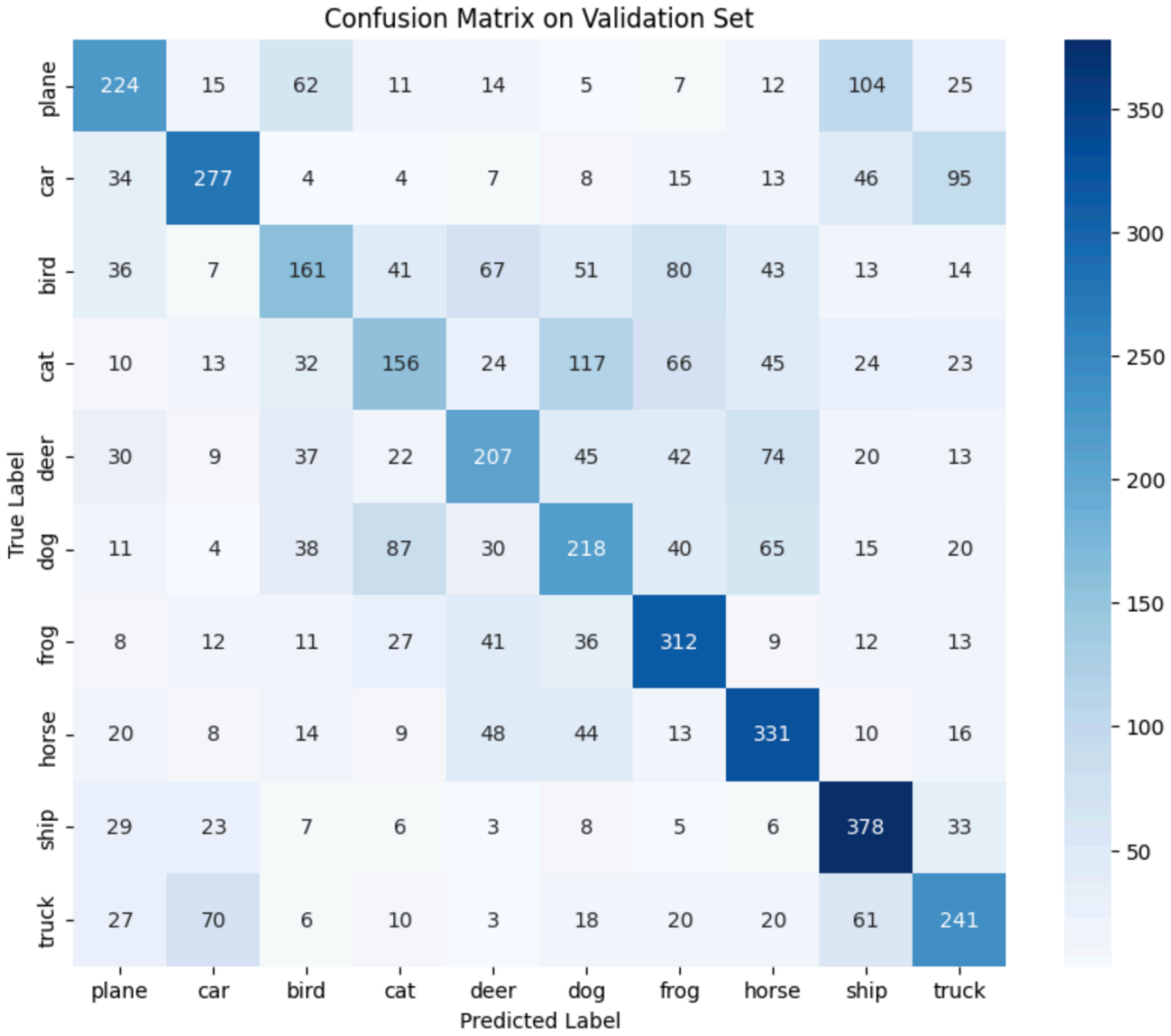
In the 12 epochs, traning loss keeps decreasing from 0.45 to 0.25, not convergent, and it also performs high accuracy rate on the traing accuracy figure, which shows the model performs good on the traning dataset.

In the 12 epochs, validation loss increases at the beginning, and decreased a little bit and keeps increasing, which means it's ***alittle bit overfitting***. And the accuracy is stable near 50%, but the accuracy rate on the validation dataset is much lower than the training dataset, which shows the ***bad generalization*** of this model.

b. Stability:

The accuracy rate on the validation dataset is stable, nearly 50% so it has ***good stability***.

2. Confusion trends:



a. General trends(accuracy):

The diagonal is much brighter, nearly 200ish, so ***the accuracy of the model is medium.***

This model ***has issue with classify ship and plane*** (val:104).

For column deer and column dog, the val is biggger than others so ***the model cannot really learn deer and dog***.

b. Precision / Recall / F1-score:

```
Metrics per class:
plane: Precision=0.4886, Recall=0.5804, F1 Score=0.5305
car: Precision=0.5958, Recall=0.5626, F1 Score=0.5787
bird: Precision=0.3795, Recall=0.3684, F1 Score=0.3739
cat: Precision=0.3313, Recall=0.3235, F1 Score=0.3274
deer: Precision=0.5142, Recall=0.3988, F1 Score=0.4492
dog: Precision=0.3622, Recall=0.4754, F1 Score=0.4111
frog: Precision=0.5314, Recall=0.4927, F1 Score=0.5113
horse: Precision=0.5767, Recall=0.5789, F1 Score=0.5778
ship: Precision=0.6998, Recall=0.6225, F1 Score=0.6589
truck: Precision=0.5168, Recall=0.5168, F1 Score=0.5168
```

The ***F1 score not too high or too low***, which means the dataset is ***balanced***.

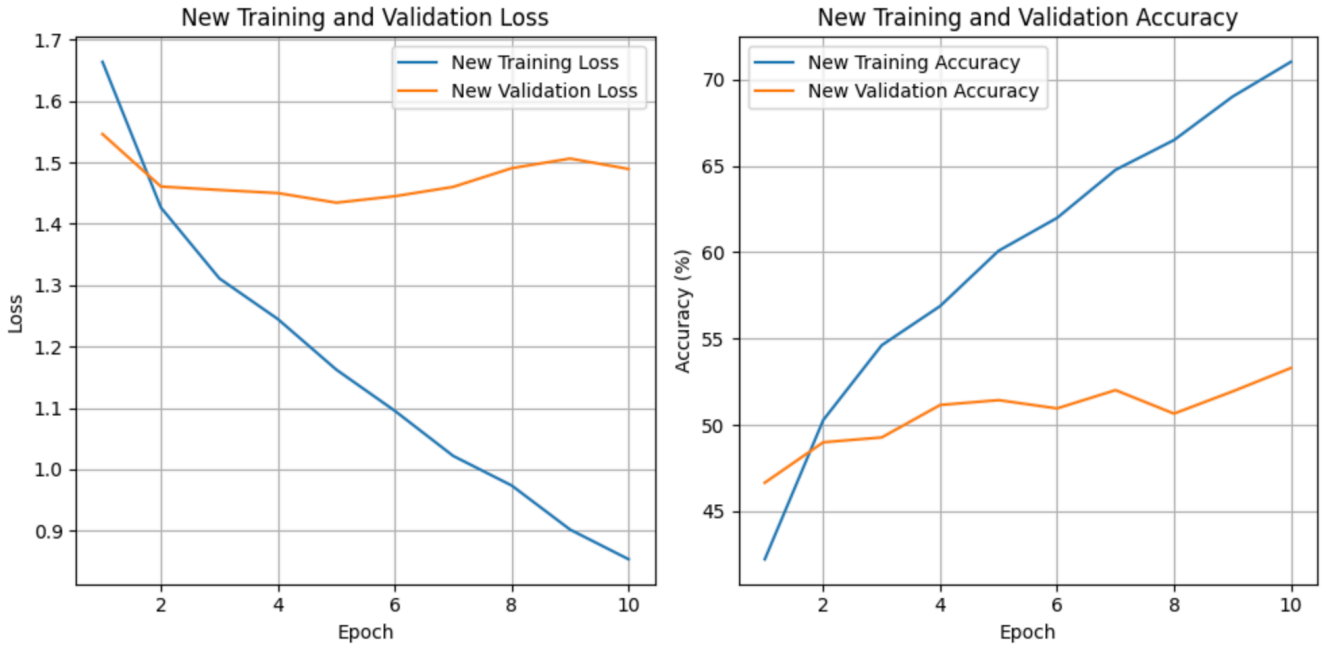
The ***cat has the lowest precision and lowest recall***, which means ***the model has a lot of erros when predicting cat class***, maybe because of ***less pics of cat, or inappropriate features***. For the ***optimization***, we can consider doing ***data augomentation to increase more cat datasets*** and ***increase the features***.

3. Effect of your changes:

a. Changed hyperparams of model:

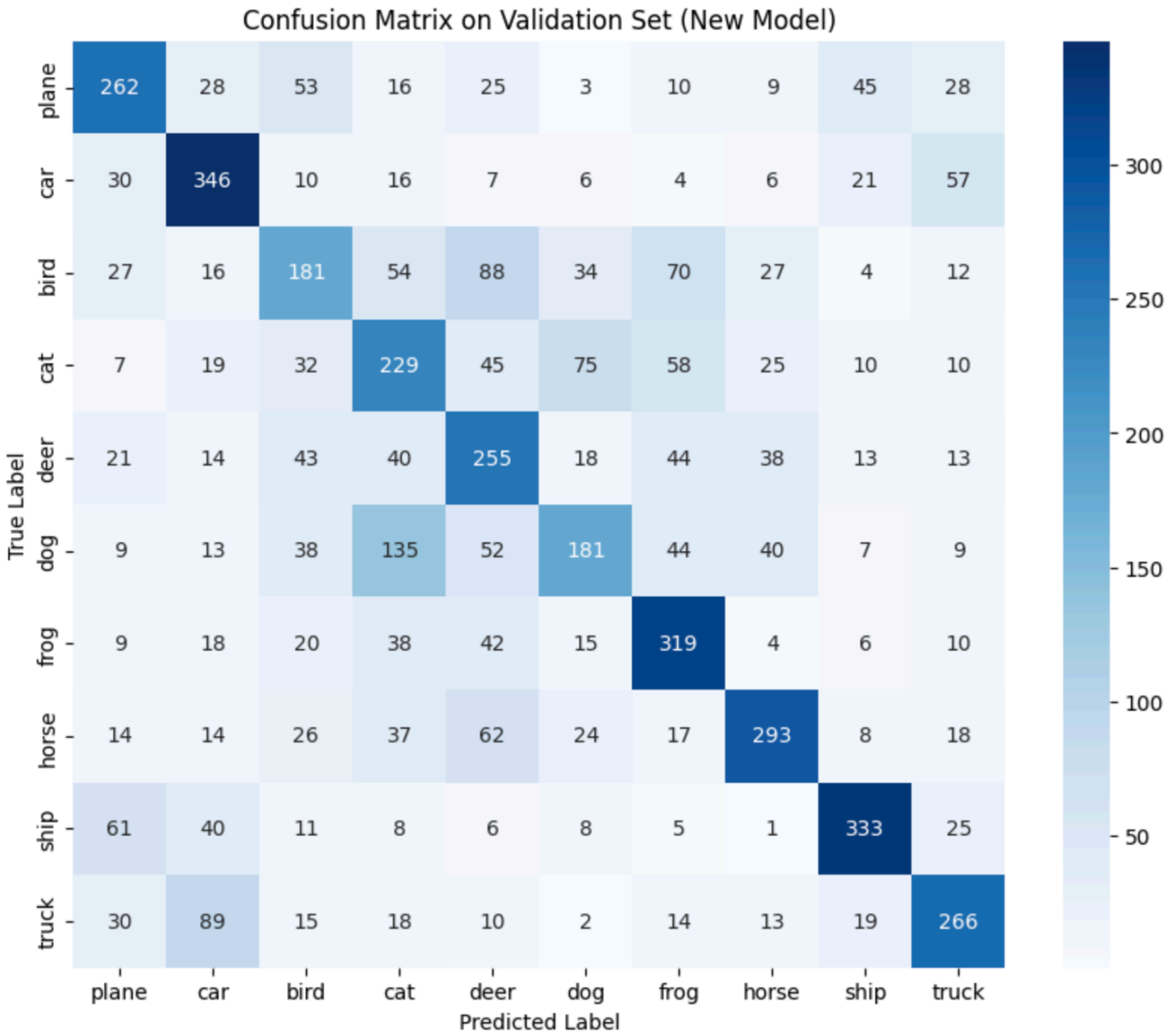
Change ***hidden_size*** from 512 to 1024 and ***learning_rate*** from 0.001 to 0.0005.

b. Performance after change:



After modifying the hyperparams, the loss and accuracy function ***converges more quickly*** compared to the original one. The traing loss decreases more quickly and traning accuracy increases more quickly.

c. confusion matrix after change:



The diagonl is ***brighter*** with more higher val compared to the original one, so we ***imporves the accuracy of this model***.

d. Summary:

We modify the ***hidden_size*** and ***learning_rate*** so that we make a ***faster and more accurate*** model. But the model initialized with FFNN will alwasy ***converges to accuracy=50%***, which is a shortness.