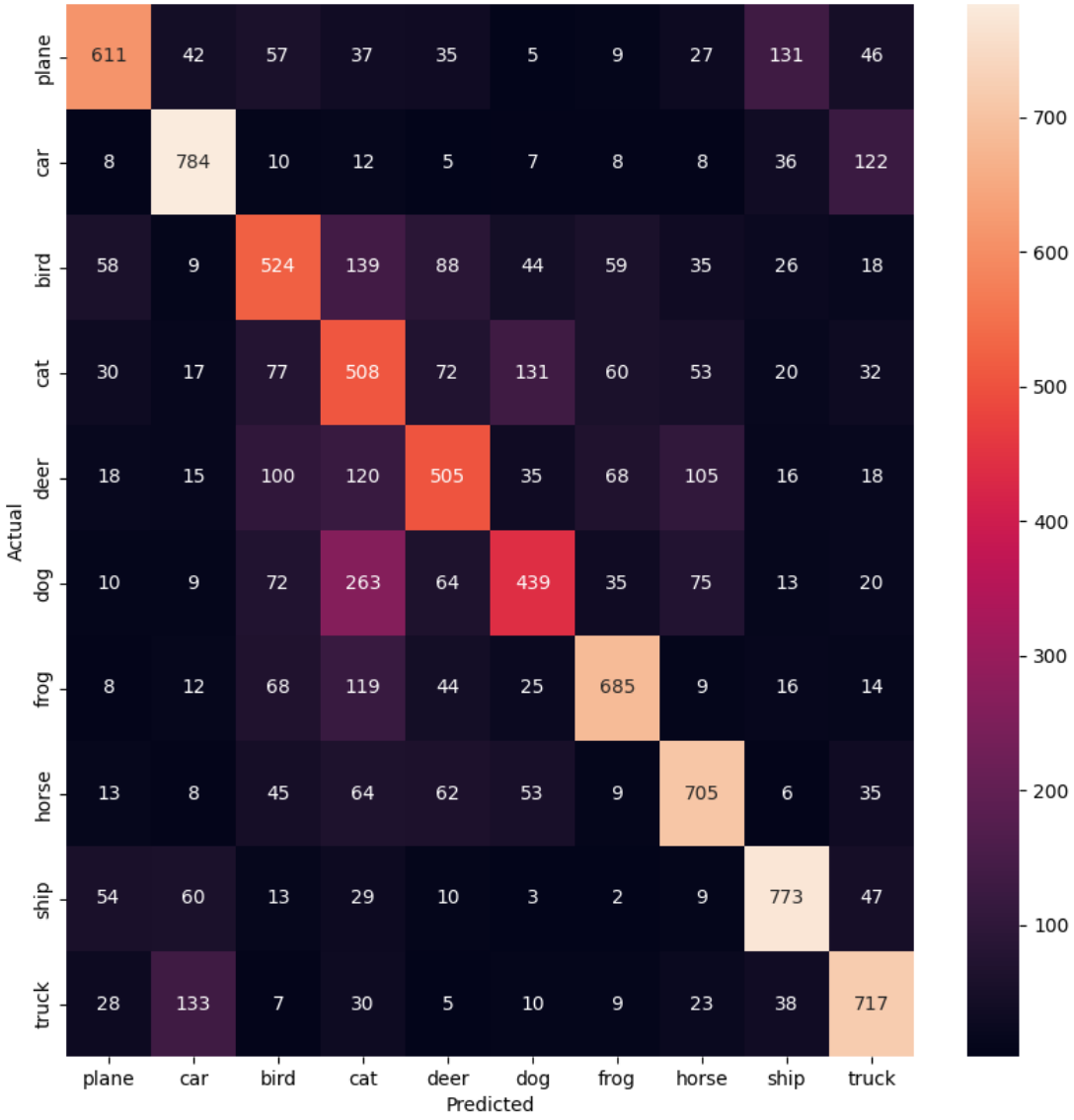
CMP_SC 8690: Computer Vision
Homework 3A: Image Classification
By: Mikey Joyce
Due: 3/14/2024

Abstract:

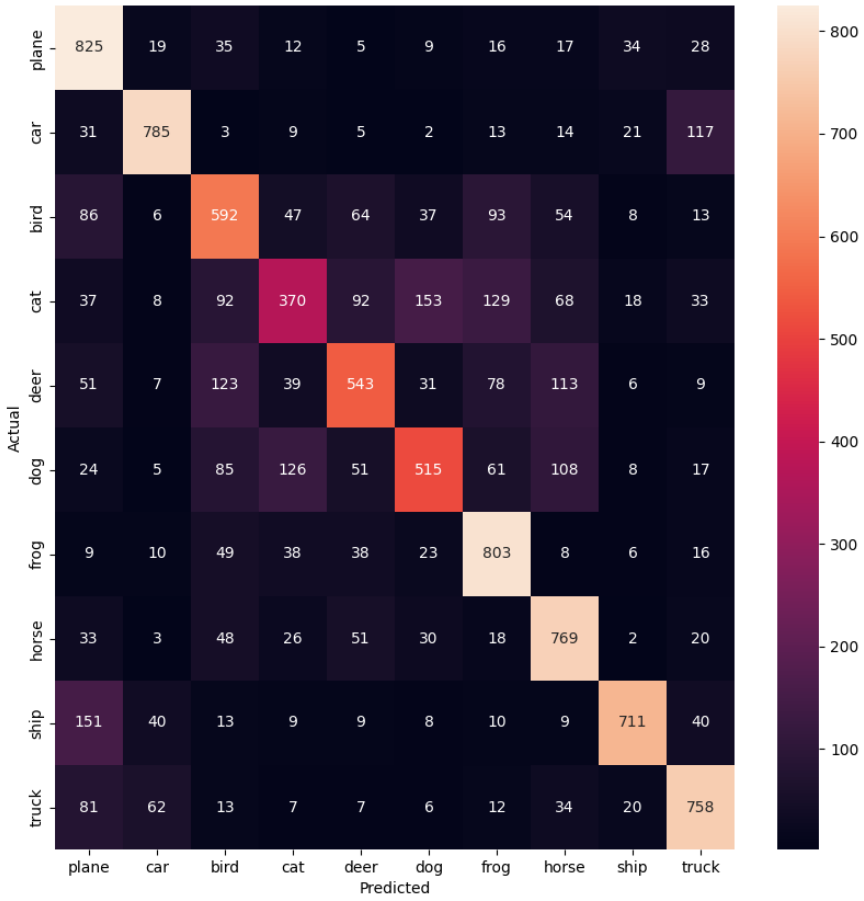
The goal of this assignment is to familiarize ourselves with the PyTorch library, a neural network programming library made by Meta. It is also to familiarize ourselves with convolutional neural networks (CNNs) a bit, however that isn't the main focus as a lot of the starting code was already given, only a few tweaks were really required to program the second network. The skills gained from this assignment will be valuable in the future as deep learning is an important part of the computer vision field. One item I would like to note is I decided to train on 10 epochs instead of only 4.

Experiments and Results:

(10 Epochs)	Network 1	Network 2
Plane	61.1%	82.5%
Car	78.4%	78.5%
Bird	52.4%	59.2%
Cat	50.8%	37.0%
Deer	50.5%	54.3%
Dog	43.9%	51.5%
Frog	68.5%	80.3%
Horse	70.5%	76.9%
Ship	77.3%	71.1%
Truck	71.7%	75.8%
Average	62%	66%
Training Running Time	637.73 seconds	640.78 seconds
Testing Running Time	9.07 seconds	8.21 seconds



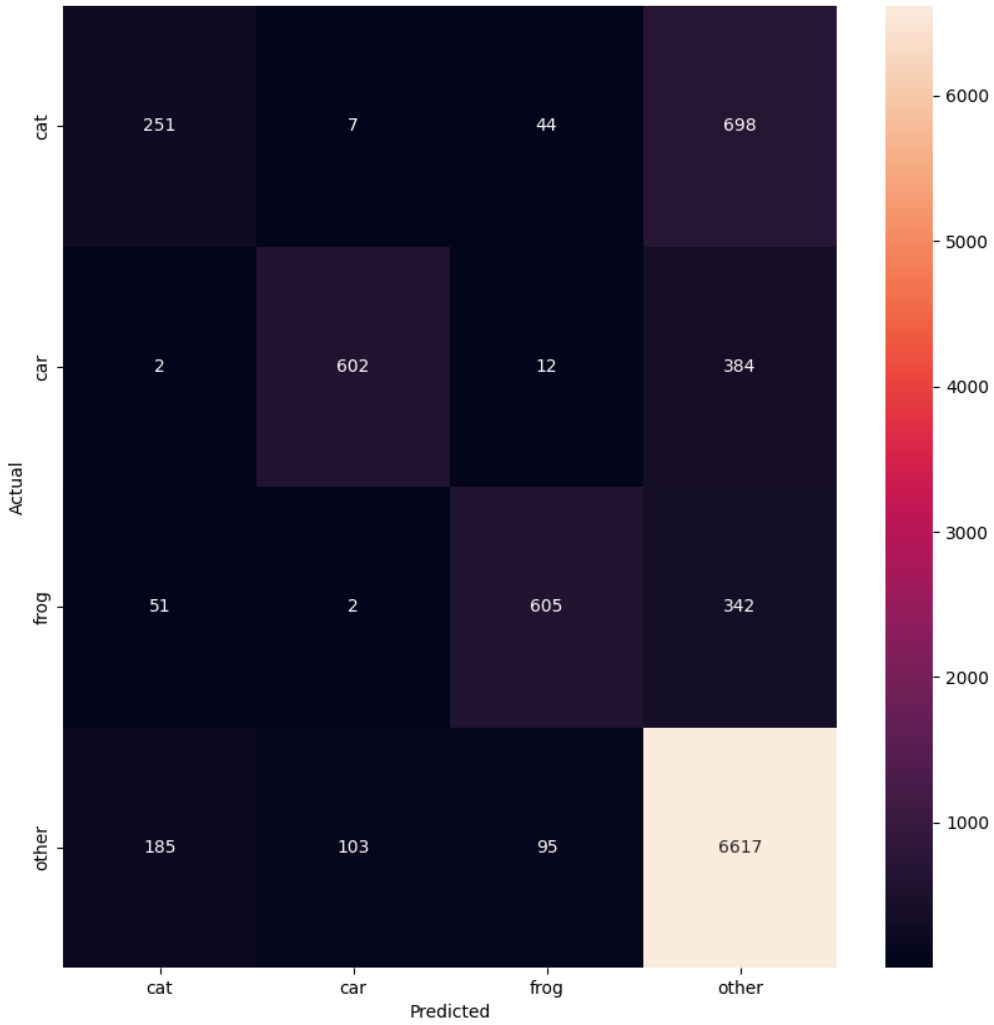
Network #1 Confusion Matrix



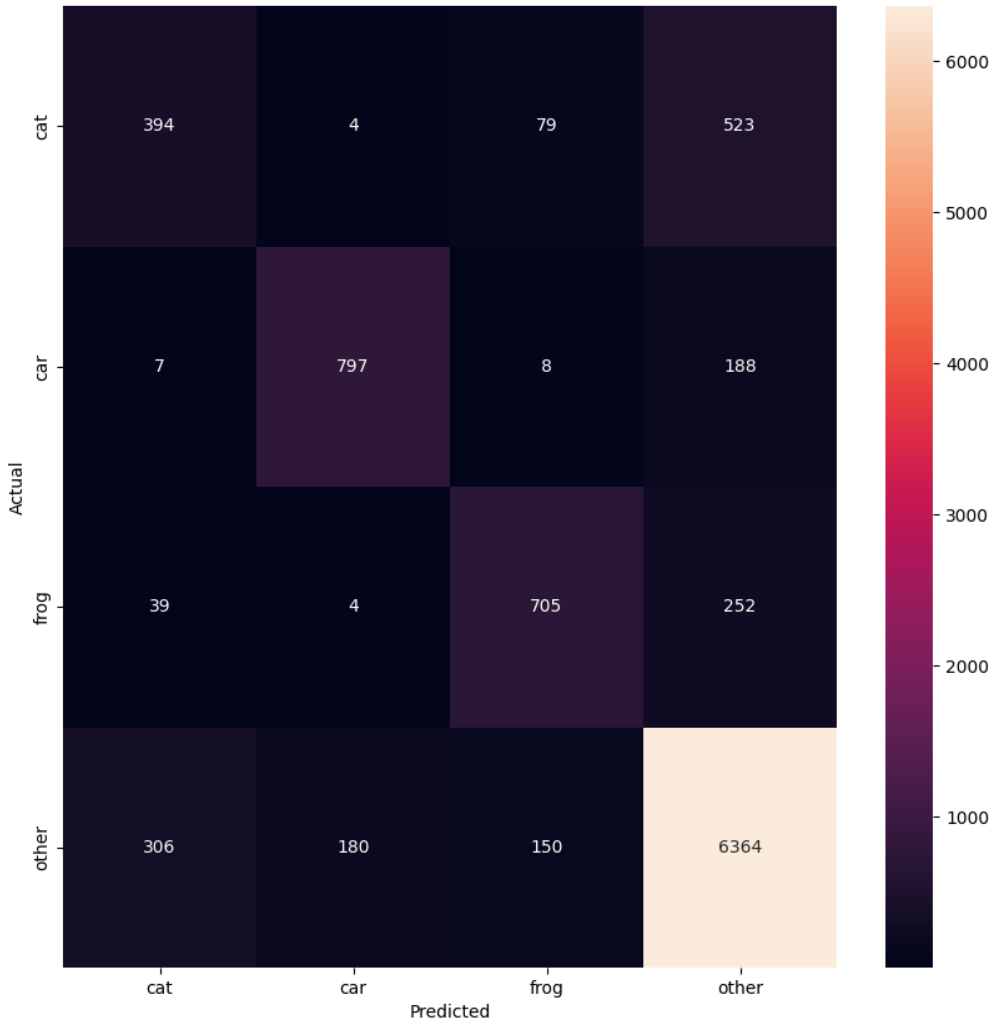
Network #2 Confusion Matrix; The class with the worst accuracy is the class 'Cat' it got confused with the 'Dog' class most of the time

Network #2 Wrong Preds v. Labels: (left to right)
Pred: Car, Label: Ship; Pred: Deer, Label: Frog; Pred: Truck, Label: Car; Pred: Truck, Label: Car; Pred: Bird, Label: Plane

(10 Epochs)	Network 1	Network 2
Cat	25.1%	39.4%
Car	60.2%	79.7%
Frog	60.5%	70.5%
Other	94.5%	90.9%
Average	80%	82%
Training Running Time	617.45 seconds	636.29 seconds
Testing Running Time	8.51 seconds	8.87 seconds



Network 1: four class confusion matrix



Network 2: four class confusion matrix

Conclusion:

Here we will discuss the results of the simple convolutional neural networks that were programmed in this assignment. We were asked to copy the CNN from the PyTorch tutorial and make sure we could get it to work properly on the CIFAR10 dataset. After this, a second network was defined in the assignment, and we were asked to modify the initial network to fit the definition of the second network. After this

we were asked to modify both of the networks to do predictions on the following classes: Cat, Car, Frog, and Other. All the classes not included should be grouped into the Other class. I was able to successfully implement every network and train the networks over 10 epochs utilizing Google Colab resources. The first network had not the greatest results, much of this is probably due to utilizing a small amount of epochs, if we trained for more epochs, I believe better accuracies would be obtained. The network got an average accuracy of 62% with the dog class being the worst class at 43.9% accuracy. The second network obtained a little bit better of an accuracy of 66% however the worst class was the cat class with an accuracy of 37%. It was interesting to see the worst class of the "better" model had a lower accuracy, what was not surprising was that the worst class was dog and then changed to cat. If you look at the confusion matrices the networks seem to have trouble deciphering between a dog and cat, it "thinks" they look somewhat similar. For this second network, a table of 5 test images were provided. Each of these images were predictions that the network got wrong, and I am going to try and attempt to give reasons why the network might have gotten them wrong. The first image is of a boat, but it decided to predict car. I think it is possible it got this wrong because many images of the boat have a background with water, however this image does not have much of a background at all, so it probably conflated that to the images of cars that are in a showroom and don't have much of a background either. The second one that it got wrong was the frog, I think it may have gotten this one wrong because the frogs color is brown similar to a deer, with a green background (a deer's natural habitat). I was not expecting it to get this one wrong. The next one it predicted is a truck but if you look it is actually a car with the trunk open. If I had to guess, I think the trunk being open gave the car a more elongated shape, similar to a truck, which is what tricked the model. The next picture is of a car, but it predicted truck. To be honest I think the model kind of got unlucky, because when I look at it, it also looks like a truck. In reality it is just a beefy SUV that looks similar to a truck. Part of the problem is that the model didn't have access to all of the data here, if we had a side shot of the vehicle, we could clearly see there is no truck bed, but instead there is a trunk, like a car has. The last one it predicted a bird, but it actually is a military plane. I think the main reason it predicted a bird is because many of the pictures of the birds have the wings and the blue sky as a background, similar to what is seen in the picture. However, this is clearly a plane. Lastly, we can discuss the last two models which were classifying four classes instead of ten. It is no surprise that

these models performed the best on the other class. This is because the data distribution was not even, it had much more training data related to the other class, which makes it easier to predict which images are part of the other class. What was interesting though, is that it got really bad scores for the cat class. My hypothesis for this is because many of the classes that the first couple models confused the cat with are contained within the other class. Since the model is skewed towards the other class, it would rather predict that the cat is part of the other class instead of the cat class.

References:

- Libraries and tools: PyCharm, OpenCV, NumPy, Matplotlib, Seaborn, PyTorch, Google Drive, Google Colab, Preview.
- https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html