Project Report

Ron Shmoilov
40247674

Model Architecture and Training

Decision Tree

Changes:

I tested out the depth of the Decision tree by changing the max depth to 30 and then to 70.

Training methodology:

Max depth: 50 (changed for the depth changes in the variants)
Criterion: Gini

MLP

Changes:

I tested multi-layer perceptrons with bigger hidden layers. My second Linear layer goes from 512 to 1024 (instead of to 512) which is considerably bigger.I also tested it with a few layers removed. I had only 2 layers. The training methodology stayed the same for all tests.

Training methodology:

Number of epochs: 1000
Learning rate: 0.003
Loss function: Cross Entropy Loss
Optimizer: SGD with 0.9 momentum

Optimization algorithms:

SGD optimizer (as stated to be used in the instructions)

CNN:

Changes:

Experimented with less convolutional layers. Instead of having 8 as stated in the project, I had 5 . I also experimented with more layers having 9 instead of 8.

I experimented with smaller and bigger kernel sizes (both 2,2 and 5x5). I used the smaller convolutional network for these experiments since the training would be quicker (the one with 5 layers)

Training methodology:

Number of epochs: 15
Learning rate: 0.001 (but dynamic using a scheduler)
Loss function: Cross Entropy Loss
Optimizer: SGD with 0.9 momentum
Model in channels: 1

Optimization techniques:

Used a scheduler for the learning rate since I realized that it gave better results
In the initial transformation for the data download I added some extra rotation to help increase the randomness of the data so that there is less chances of overfitting.

Evaluation

Metrics:

Naïve Bayes:

	precision	recall	f1-score	support
0	0.73	0.79	0.76	100
1	0.95	0.90	0.92	100
2	0.76	0.64	0.70	100
3	0.66	0.81	0.73	100
4	0.73	0.73	0.73	100
5	0.80	0.71	0.75	100
6	0.77	0.84	0.80	100
7	0.84	0.76	0.80	100
8	0.85	0.88	0.86	100
9	0.91	0.88	0.89	100
accuracy			0.79	
macro avg	0.80	0.79	0.79	
weighted avg	0.80	0.79	0.79	

MLP:

	precision	recall	f1-score	support

0	0.90	0.95	0.93	100
1	0.98	0.95	0.96	100
2	0.88	0.89	0.89	100
3	0.82	0.85	0.83	100
4	0.91	0.89	0.90	100
5	0.88	0.85	0.86	100
6	0.95	0.94	0.94	100
7	0.92	0.93	0.93	100
8	0.98	0.95	0.96	100
9	0.96	0.97	0.97	100
accuracy			0.92	
macro avg	0.92	0.92	0.92	
weighted avg	0.92	0.92	0.92	

MLP More hidden Layers:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	100
1	0.99	0.97	0.98	100
2	0.98	0.99	0.99	100
3	0.94	0.98	0.96	100
4	0.97	0.98	0.98	100
5	1.00	0.98	0.99	100
6	0.99	0.97	0.98	100
7	1.00	0.99	0.99	100
8	1.00	0.99	0.99	100
9	0.99	0.99	0.99	100
accuracy			0.98	
macro avg	0.98	0.98	0.98	
weighted avg	0.98	0.98	0.98	

MLP Removed Layers:

	precision	recall	f1-score	support
0	0.86	0.92	0.89	100
1	0.99	0.95	0.97	100
2	0.83	0.82	0.82	100

3	0.81	0.81	0.81	100
4	0.86	0.87	0.87	100
5	0.82	0.80	0.81	100
6	0.89	0.90	0.90	100
7	0.92	0.89	0.90	100
8	0.94	0.94	0.94	100
9	0.95	0.96	0.96	100
accuracy			0.89	
macro avg	0.89	0.89	0.89	
weighted avg	0.89	0.89	0.89	

Decision Tree:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	100
1	1.00	1.00	1.00	100
2	1.00	0.99	0.99	100
3	0.98	0.99	0.99	100
4	0.99	1.00	1.00	100
5	0.99	0.98	0.98	100
6	1.00	1.00	1.00	100
7	1.00	1.00	1.00	100
8	1.00	1.00	1.00	100
9	1.00	1.00	1.00	100
accuracy			1.00	
macro avg	1.00	1.00	1.00	
weighted avg	1.00	1.00	1.00	

Bigger Depth

	precision	recall	f1-score	support
0	1.00	1.00	1.00	100
1	0.99	1.00	1.00	100
2	1.00	0.99	0.99	100
3	1.00	0.99	0.99	100
4	0.99	1.00	1.00	100
5	0.99	1.00	1.00	100
6	1.00	1.00	1.00	100
7	1.00	1.00	1.00	100
8	1.00	1.00	1.00	100
9	1.00	0.99	0.99	100

accuracy			1.00	1000
macro avg	1.00	1.00	1.00	1000
weighted avg	1.00	1.00	1.00	1000

Smaller Depth:

	precision	recall	f1-score	support
0	0.99	1.00	1.00	100
1	1.00	0.98	0.99	100
2	1.00	0.99	0.99	100
3	0.99	0.99	0.99	100
4	1.00	1.00	1.00	100
5	0.99	0.99	0.99	100
6	0.99	1.00	1.00	100
7	1.00	1.00	1.00	100
8	0.99	0.99	0.99	100
9	0.99	1.00	1.00	100

accuracy			0.99	1000
macro avg	0.99	0.99	0.99	1000
weighted avg	0.99	0.99	0.99	1000

CNN:

	precision	recall	f1-score	support
0	0.76	0.75	0.75	100
1	0.91	0.98	0.94	100
2	0.75	0.70	0.73	100
3	0.66	0.60	0.63	100
4	0.75	0.72	0.73	100
5	0.75	0.77	0.76	100
6	0.89	0.89	0.89	100
7	0.86	0.88	0.87	100
8	0.86	0.93	0.89	100
9	0.91	0.91	0.91	100

accuracy		0.81	
macro avg	0.81	0.81	0.81
weighted avg	0.81	0.81	0.81

VGG Small:

	precision	recall	f1-score	support
0	0.69	0.70	0.69	100
1	0.83	0.94	0.88	100
2	0.63	0.54	0.58	100
3	0.62	0.47	0.53	100
4	0.65	0.65	0.65	100
5	0.64	0.71	0.67	100
6	0.82	0.86	0.84	100
7	0.80	0.86	0.83	100
8	0.82	0.85	0.83	100
9	0.79	0.75	0.77	100
accuracy			0.73	1000
macro avg	0.73	0.73	0.73	1000
weighted avg	0.73	0.73	0.73	1000

VGG Big:

	precision	recall	f1-score	support
0	0.83	0.76	0.79	100
1	0.92	0.93	0.93	100
2	0.79	0.70	0.74	100
3	0.67	0.54	0.60	100
4	0.71	0.78	0.74	100
5	0.69	0.84	0.76	100
6	0.84	0.87	0.86	100
7	0.86	0.88	0.87	100
8	0.86	0.91	0.88	100
9	0.92	0.87	0.89	100
accuracy			0.81	1000
macro avg	0.81	0.81	0.81	1000
weighted avg	0.81	0.81	0.81	1000

VGG Small Kernel:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	100
1	0.10	0.91	0.18	100
2	0.20	0.01	0.02	100
3	0.00	0.00	0.00	100
4	0.00	0.00	0.00	100
5	0.24	0.04	0.07	100

6	0.00	0.00	0.00	100
7	0.00	0.00	0.00	100
8	0.21	0.12	0.15	100
9	0.00	0.00	0.00	100
accuracy			0.11	1000
macro avg	0.07	0.11	0.04	1000
weighted avg	0.07	0.11	0.04	1000

VGG Big Kernel:

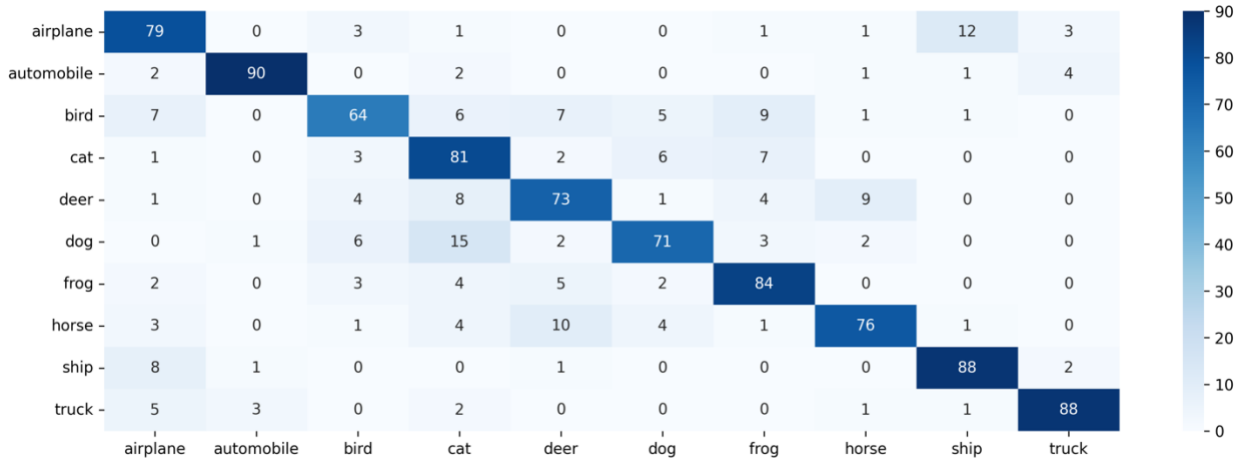
	precision	recall	f1-score	support
0	0.18	0.16	0.17	100
1	0.11	0.21	0.15	100
2	0.00	0.00	0.00	100
3	0.16	0.13	s 0.14	100
4	0.19	0.16	0.17	100
5	0.00	0.00	0.00	100
6	0.11	0.31	0.17	100
7	0.00	0.00	0.00	100
8	0.13	0.05	0.07	100
9	0.17	0.42	0.24	100
accuracy			0.14	1000
macro avg	0.11	0.14	0.11	1000
weighted avg	0.11	0.14	0.11	1000

Performance:

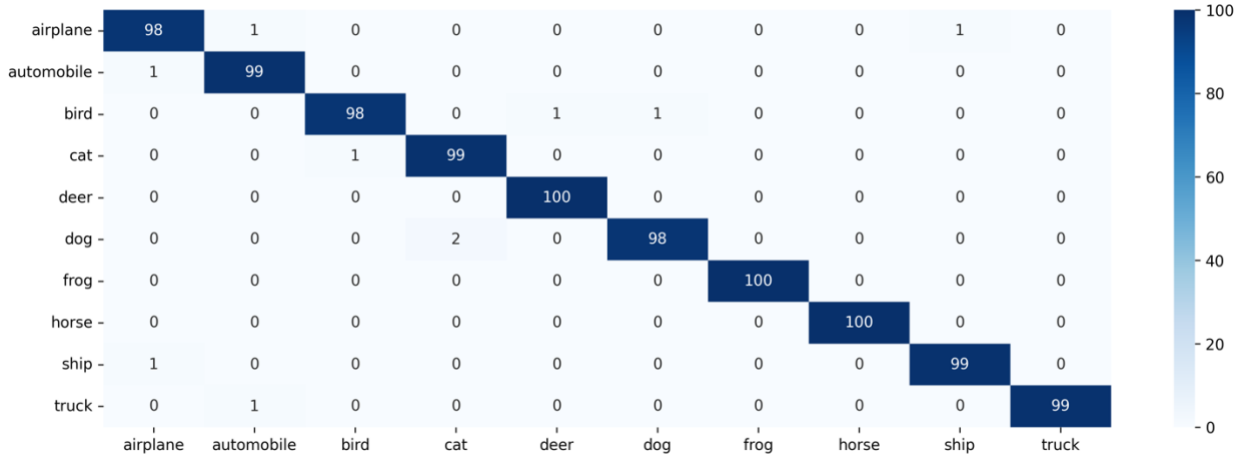
Decision tree had by far the best performance all around. It seems as thought it can correctly choose the correct splitting criterion and do so very efficiently.

Confusion matrices

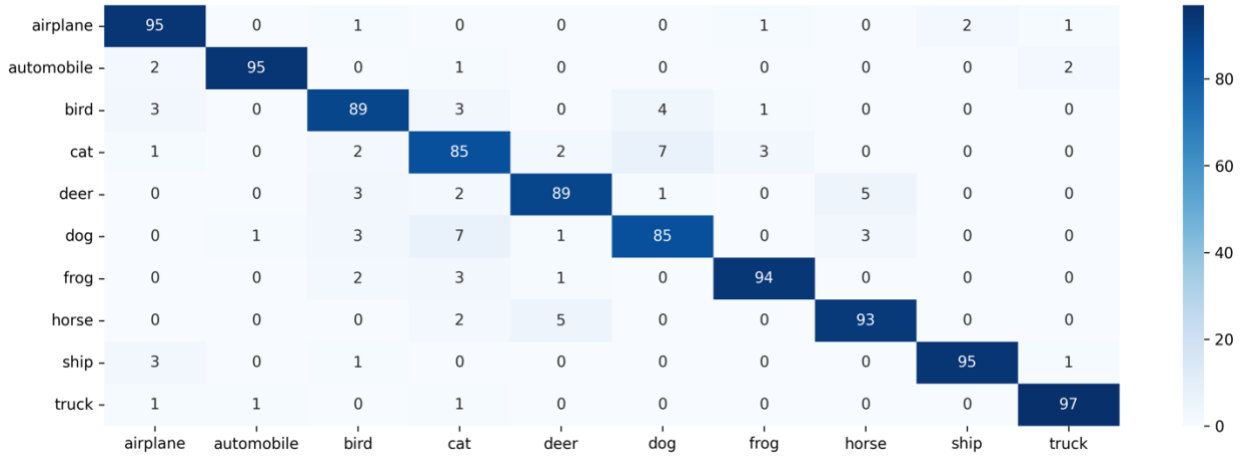
Naïve Bayes:
Sklearn:



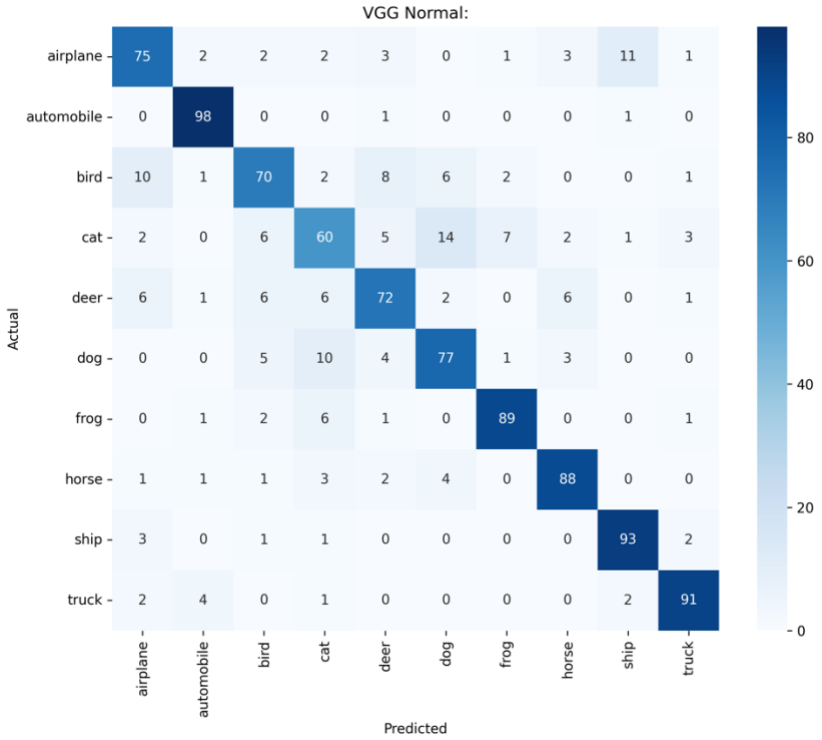
Decision Tree:



MLP:



CNN:



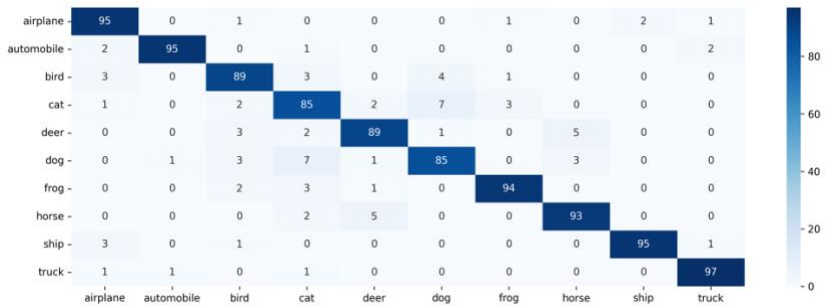
Misclassification:

The VGG smaller and bigger kernel size massively misclassified the data. This can be due to how much harder it is to generalize data with a smaller kernel since it misses the bigger picture). And how a bigger kernel overgeneralizes.
Depth:

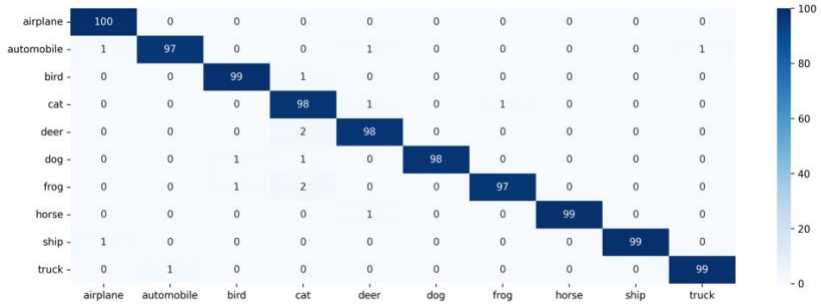
Layer size and Kernel size:

More depth in the CNN seems to have overfit the data and has performed slightly worse than the initial normal CNN. Same for the MLP. Bigger hidden layers have resulted in the MLP producing stronger results.

Normal CNN Results:



Increased hidden layers:



Best and Worst performance

The best performance goes to the MLP with increased hidden layers and the Decision tree. CNN seems to be a bit too complex for the dataset provided and take too much time to train. Naïve Bayes didn't do poorly but Decision trees are capability of distinguishing small details much better than Naïve Bayes.