Image Classification Using Various Models

Author(s): Alvi Matin - 40250537, Rahath Ahmed - 40252092

Department of Gina Cody School of Engineering and Computer Science, University of Concordia

COMP 472: Introduction to Artificial Intelligence

Professor: Kefaya Qaddoum

November 25, 2024

https://github.com/Meltcus/COMP472_Alvi_Rath

Contents

Gaussian Naive Bayes (GNB)

Main Model

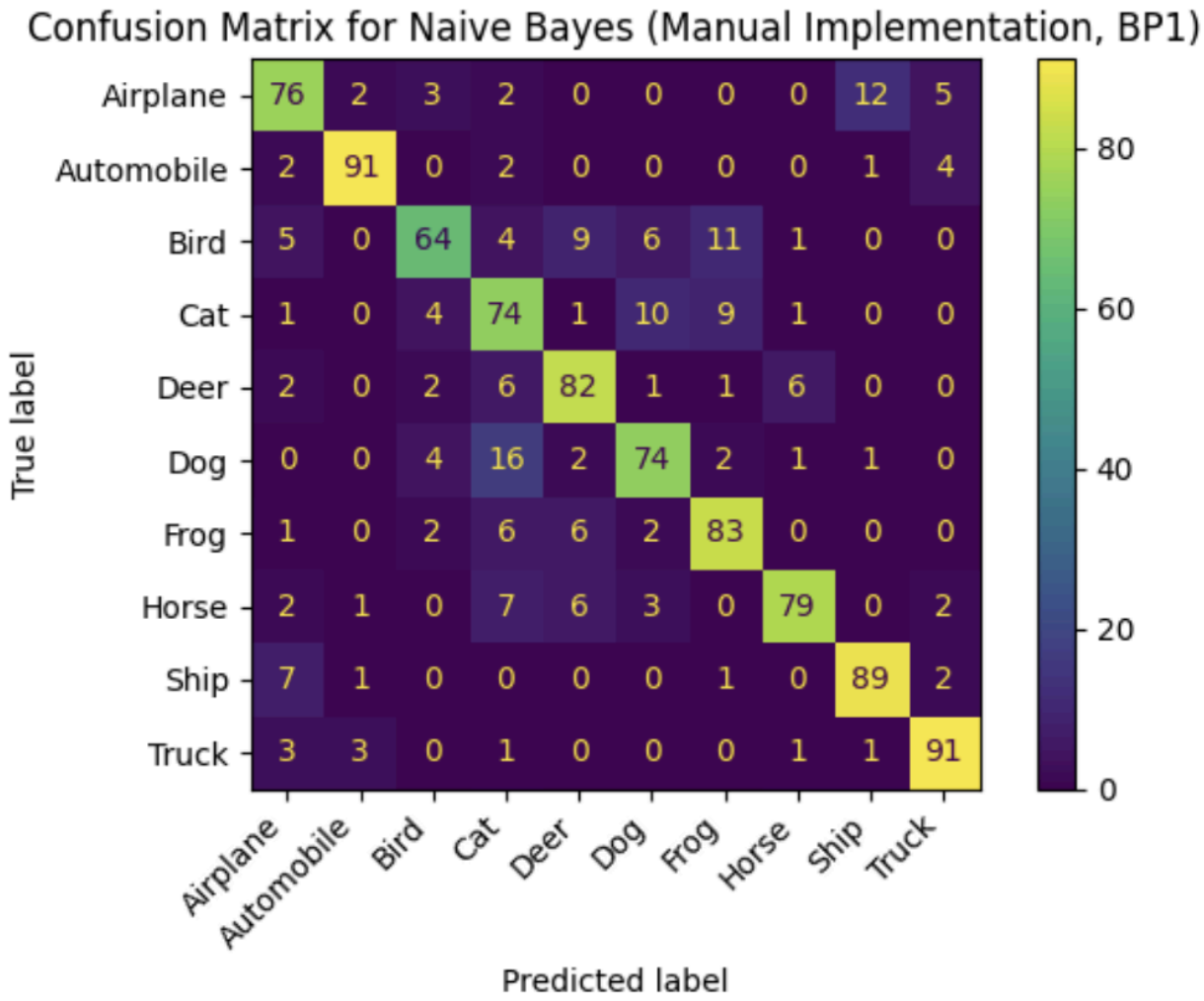
The Gaussian Naive Bayes model is a probabilistic model and does not involve traditional iterative training processes. As such, the training approach of the model calculates the mean and variance of each feature for every class in the dataset during the training phase. These statistics are then used to compute the probability of each class for a given test example based on Bayes' theorem. There are no epochs as it is not applicable in GNB, the required parameters are computed in a single pass over the data. Learning rates are also not applicable as there are no weights involved. The loss function used is the joint log-likelihood, log p(X, Y). A notable hyperparameter in GNB is the smoothing parameter to avoid zero probabilities; in our model, we add an extremely tiny number to our variance to avoid division by zero.

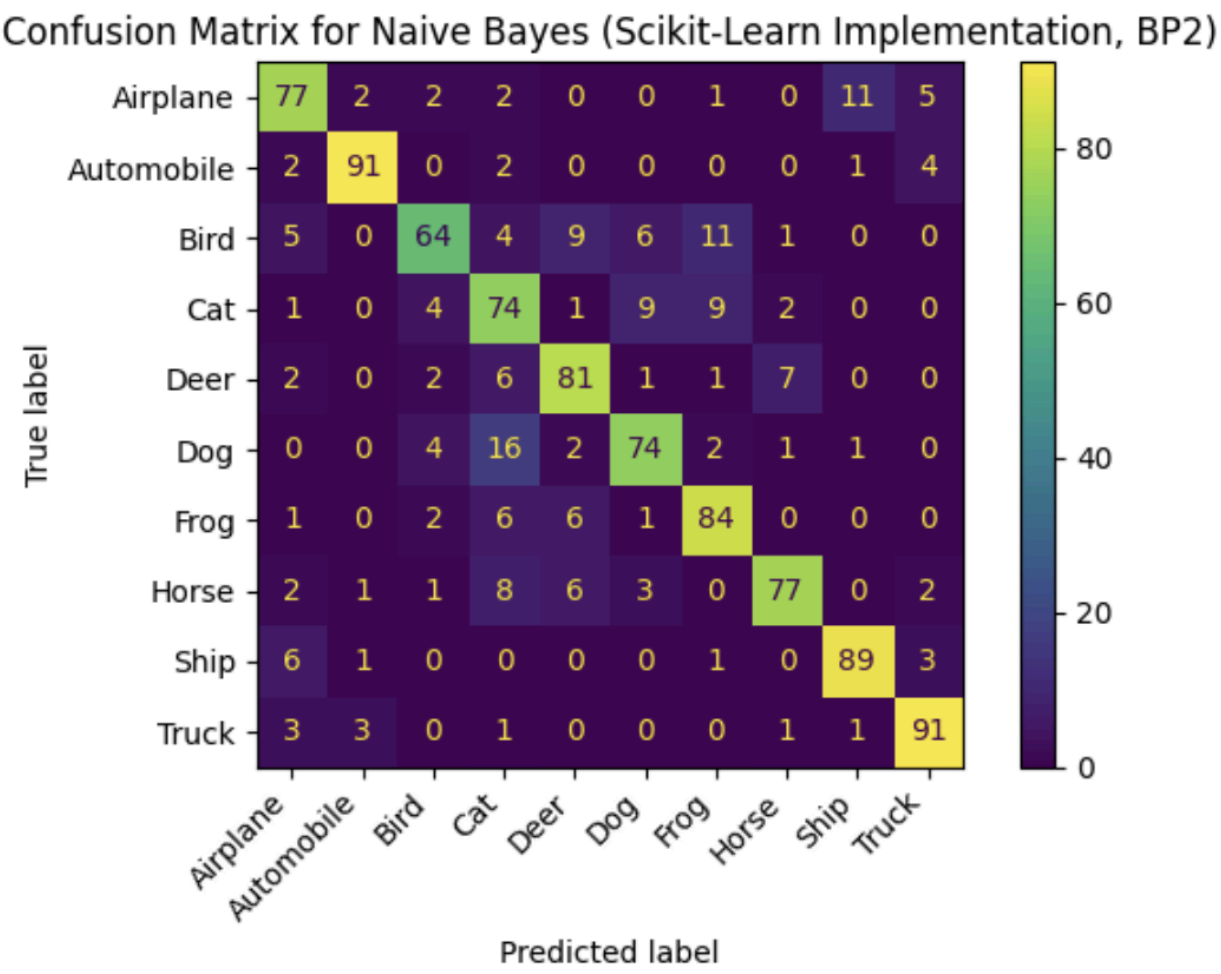
Variants

Scikit's Gaussian Naive Bayes classifier under-the-hood implementation is abstracted away. There does not seem to be a huge difference between ours and Scikit's implementation. We can deduce our implementations are very similar. They both have extremely similar confusion matrices and classification reports.

Evaluation

Confusion Matrices:




Confusion Matrix for Naive Bayes (Scikit-Learn Implementation, BP2)
Airplane 77 2 2 2 0 0 1 0 11 5
Automobile 2 91 0 2 0 0 0 0 1 4
Bird 5 0 64 4 9 6 11 1 0 0
Cat 1 0 4 74 1 9 9 2 0 0
Deer 2 0 2 6 81 1 1 7 0 0
Dog 0 0 4 16 2 74 2 1 1 0
Frog 1 0 2 6 6 1 84 0 0 0
Horse 2 1 1 8 6 3 0 77 0 2
Ship 6 1 0 0 0 0 1 0 89 3
Truck 3 3 0 1 0 0 0 1 1 91
True label
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
Predicted label
80
60
40
20
0

Classification Reports:

```
Classification Report for Naive Bayes (Manual Implementation, BP1):
              precision    recall  f1-score   support

    Airplane       0.77      0.76      0.76       100
  Automobile       0.93      0.91      0.92       100
        Bird       0.81      0.64      0.72       100
         Cat       0.63      0.74      0.68       100
        Deer       0.77      0.82      0.80       100
         Dog       0.77      0.74      0.76       100
        Frog       0.78      0.83      0.80       100
       Horse       0.89      0.79      0.84       100
        Ship       0.86      0.89      0.87       100
       Truck       0.88      0.91      0.89       100

    accuracy                           0.80      1000
   macro avg       0.81      0.80      0.80      1000
weighted avg       0.81      0.80      0.80      1000

Accuracy for BP1 (Manual Implementation): 80.30%
```

```
Classification Report for Naive Bayes (Scikit-Learn Implementation, BP2):
              precision    recall  f1-score   support

    Airplane       0.78      0.77      0.77       100
  Automobile       0.93      0.91      0.92       100
        Bird       0.81      0.64      0.72       100
         Cat       0.62      0.74      0.68       100
        Deer       0.77      0.81      0.79       100
         Dog       0.79      0.74      0.76       100
        Frog       0.77      0.84      0.80       100
       Horse       0.87      0.77      0.81       100
        Ship       0.86      0.89      0.88       100
       Truck       0.87      0.91      0.89       100

    accuracy                           0.80      1000
   macro avg       0.81      0.80      0.80      1000
weighted avg       0.81      0.80      0.80      1000

Accuracy for BP2 (Scikit-Learn Implementation): 80.20%
```

Decision Tree (DT)

Main Model

The Decision Tree (DT) model is a supervised learning model and does not use traditional iterative training processes. This means the tree is built recursively based on the feature and threshold that minimizes Gini impurity. It is inherently a greedy approach due to selecting the lowest weighted impurity from all possible features and thresholds. There are no epochs as it is constructed recursively until a stopping criterion is reached. No learning rate is used as no gradient-based optimization is being used. The model used the Gini impurity as its loss function. The hyperparameters are all related to the structure of the tree such as max_depth, min_samples_split, and min_samples_leaf which are also the stopping criteria.

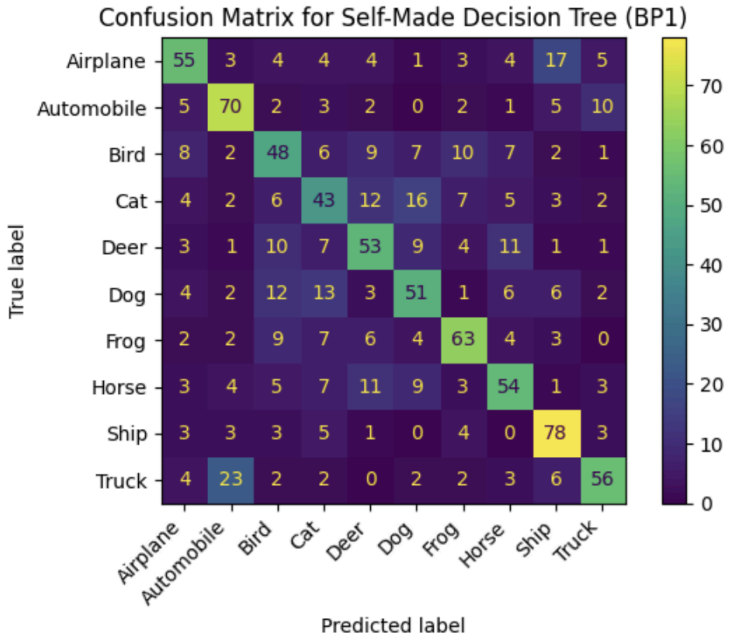
Variants

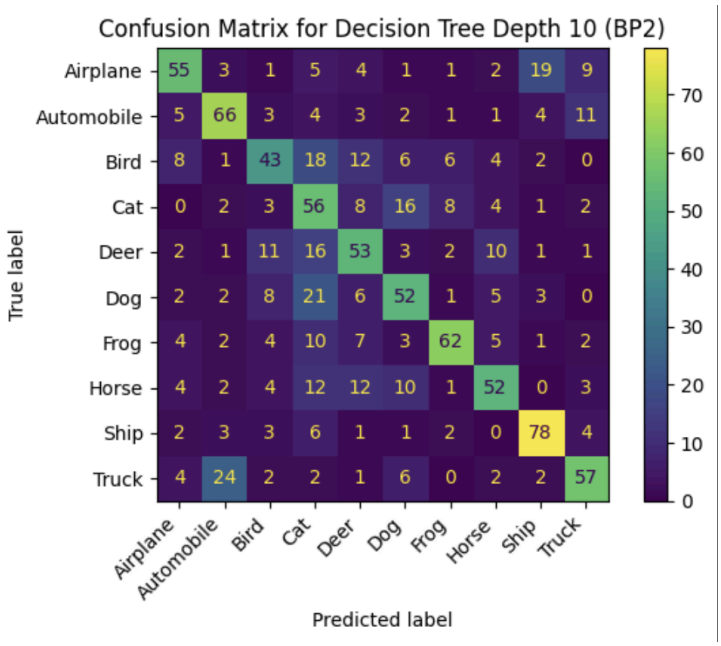
Scikit's Decision Tree classifier under-the-hood implementation is abstracted away. There does not seem to be a huge difference between ours and Scikit's implementation other than Scikit's numerous optimizations using C/C++ backend code. We can deduce our implementations are very similar. They both have extremely similar confusion matrices and classification reports when given the same max_depth to work with. Our implementation seems to train much slower than Scikit's implementation due to not fully optimizing wherever possible.

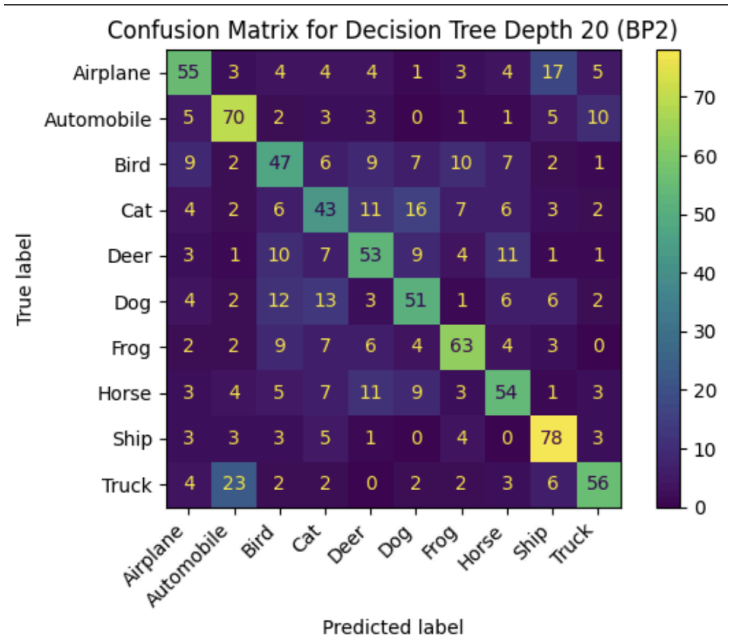
The variants of our main model change the max_depth of the decision tree. We test using depths 10, 20, 30, 40 and 50. There are no notable differences in the confusion matrices and classification reports of all the depths. There is a graph detailing the relationship between the number of depths vs accuracy in our code.

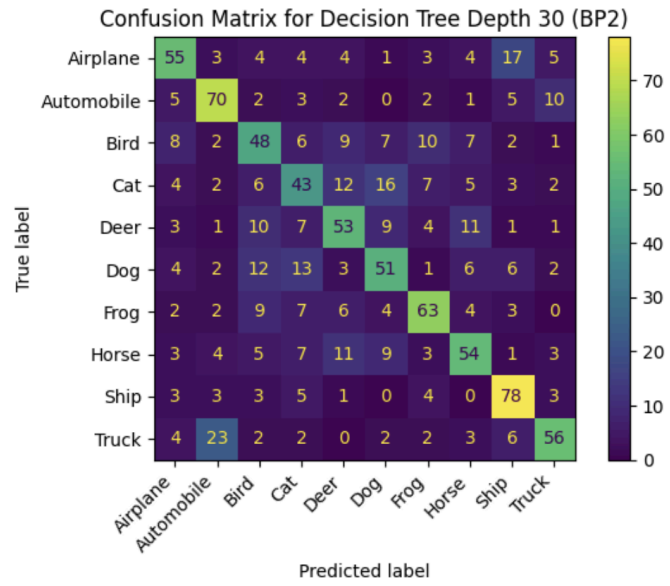
Evaluation

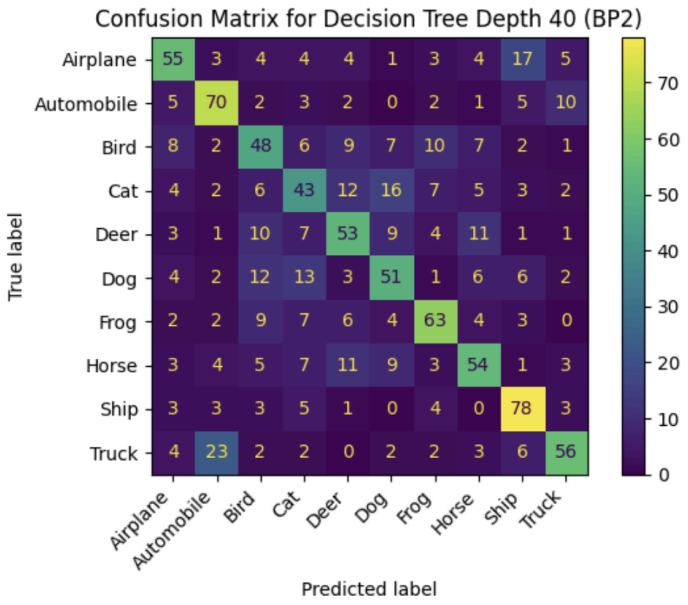
Confusion Matrices:

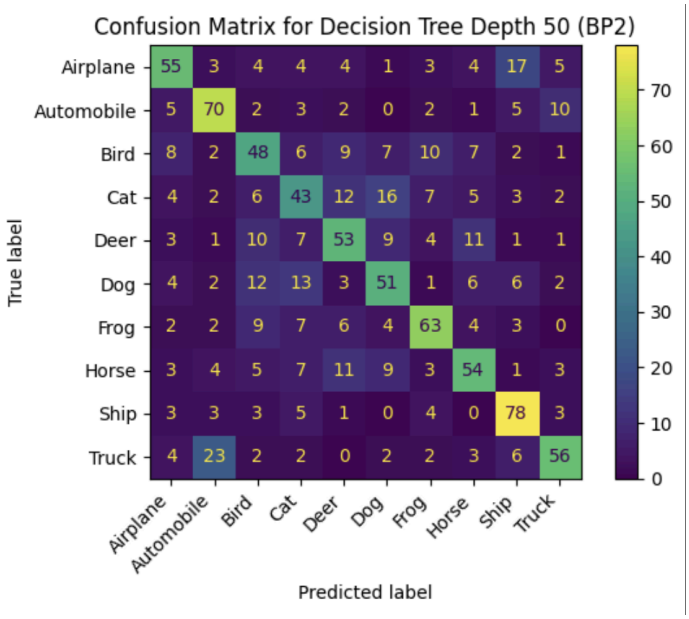


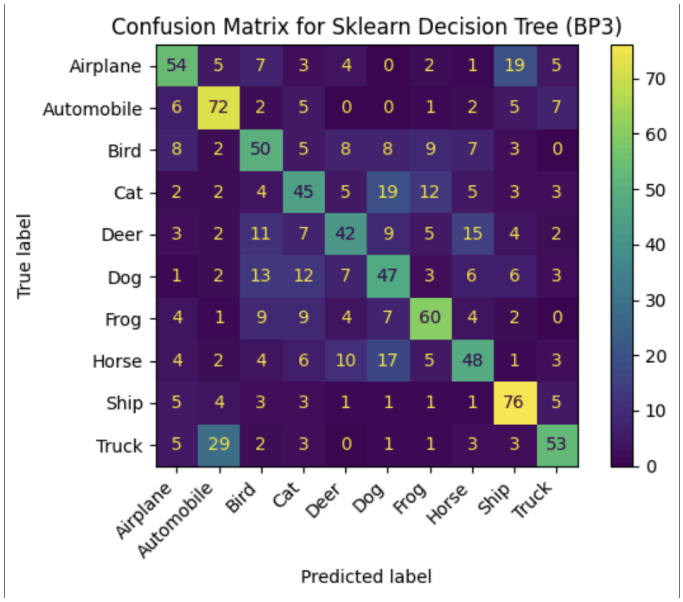



Confusion Matrix for Decision Tree Depth 20 (BP2)
Airplane 55 3 4 4 4 1 3 4 17 5
Automobile 5 70 2 3 3 0 1 1 5 10
Bird 9 2 47 6 9 7 10 7 2 1
Cat 4 2 6 43 11 16 7 6 3 2
Deer 3 1 10 7 53 9 4 11 1 1
Dog 4 2 12 13 3 51 1 6 6 2
Frog 2 2 9 7 6 4 63 4 3 0
Horse 3 4 5 7 11 9 3 54 1 3
Ship 3 3 3 5 1 0 4 0 78 3
Truck 4 23 2 2 0 2 2 3 6 56
Airplane Automobile Bird Cat Deer Dog Frog Horse Ship Truck
True label
Predicted label
0 10 20 30 40 50 60 70


Confusion Matrix for Decision Tree Depth 30 (BP2)
Airplane 55 3 4 4 4 1 3 4 17 5
Automobile 5 70 2 3 2 0 2 1 5 10
Bird 8 2 48 6 9 7 10 7 2 1
Cat 4 2 6 43 12 16 7 5 3 2
Deer 3 1 10 7 53 9 4 11 1 1
Dog 4 2 12 13 3 51 1 6 6 2
Frog 2 2 9 7 6 4 63 4 3 0
Horse 3 4 5 7 11 9 3 54 1 3
Ship 3 3 3 5 1 0 4 0 78 3
Truck 4 23 2 2 0 2 2 3 6 56
Airplane Automobile Bird Cat Deer Dog Frog Horse Ship Truck
True label
Predicted label
0 10 20 30 40 50 60 70


Confusion Matrix for Decision Tree Depth 40 (BP2)
Airplane 55 3 4 4 4 1 3 4 17 5
Automobile 5 70 2 3 2 0 2 1 5 10
Bird 8 2 48 6 9 7 10 7 2 1
Cat 4 2 6 43 12 16 7 5 3 2
Deer 3 1 10 7 53 9 4 11 1 1
Dog 4 2 12 13 3 51 1 6 6 2
Frog 2 2 9 7 6 4 63 4 3 0
Horse 3 4 5 7 11 9 3 54 1 3
Ship 3 3 3 5 1 0 4 0 78 3
Truck 4 23 2 2 0 2 2 3 6 56
Airplane Automobile Bird Cat Deer Dog Frog Horse Ship Truck
True label
Predicted label
70 60 50 40 30 20 10 0


Confusion Matrix for Decision Tree Depth 50 (BP2)
Airplane 55 3 4 4 4 1 3 4 17 5
Automobile 5 70 2 3 2 0 2 1 5 10
Bird 8 2 48 6 9 7 10 7 2 1
Cat 4 2 6 43 12 16 7 5 3 2
Deer 3 1 10 7 53 9 4 11 1 1
Dog 4 2 12 13 3 51 1 6 6 2
Frog 2 2 9 7 6 4 63 4 3 0
Horse 3 4 5 7 11 9 3 54 1 3
Ship 3 3 3 5 1 0 4 0 78 3
Truck 4 23 2 2 0 2 2 3 6 56
Airplane Automobile Bird Cat Deer Dog Frog Horse Ship Truck
True label
Predicted label
70 60 50 40 30 20 10 0


Confusion Matrix for Sklearn Decision Tree (BP3)
Airplane 54 5 7 3 4 0 2 1 19 5
Automobile 6 72 2 5 0 0 1 2 5 7
Bird 8 2 50 5 8 8 9 7 3 0
Cat 2 2 4 45 5 19 12 5 3 3
Deer 3 2 11 7 42 9 5 15 4 2
Dog 1 2 13 12 7 47 3 6 6 3
Frog 4 1 9 9 4 7 60 4 2 0
Horse 4 2 4 6 10 17 5 48 1 3
Ship 5 4 3 3 1 1 1 1 76 5
Truck 5 29 2 3 0 1 1 3 3 53
True label
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
Predicted label
70
60
50
40
30
20
10
0

Classification Reports:

Classification Report for Decision Tree BP1:

	precision	recall	f1-score	support
Airplane	0.60	0.55	0.58	100
Automobile	0.62	0.70	0.66	100
Bird	0.48	0.48	0.48	100
Cat	0.44	0.43	0.44	100
Deer	0.52	0.53	0.53	100
Dog	0.52	0.51	0.51	100
Frog	0.64	0.63	0.63	100
Horse	0.57	0.54	0.55	100
Ship	0.64	0.78	0.70	100
Truck	0.67	0.56	0.61	100
accuracy			0.57	1000
macro avg	0.57	0.57	0.57	1000
weighted avg	0.57	0.57	0.57	1000

Note: This is the self-made decision tree implementation's classification report

Classification Report for Decision Tree Depth 10:

	precision	recall	f1-score	support
Airplane	0.64	0.55	0.59	100
Automobile	0.62	0.66	0.64	100
Bird	0.52	0.43	0.47	100
Cat	0.37	0.56	0.45	100
Deer	0.50	0.53	0.51	100
Dog	0.52	0.52	0.52	100
Frog	0.74	0.62	0.67	100
Horse	0.61	0.52	0.56	100
Ship	0.70	0.78	0.74	100
Truck	0.64	0.57	0.60	100
accuracy			0.57	1000
macro avg	0.59	0.57	0.58	1000
weighted avg	0.59	0.57	0.58	1000

Classification Report for Decision Tree Depth 20:

	precision	recall	f1-score	support
Airplane	0.60	0.55	0.57	100
Automobile	0.62	0.70	0.66	100
Bird	0.47	0.47	0.47	100
Cat	0.44	0.43	0.44	100
Deer	0.52	0.53	0.53	100
Dog	0.52	0.51	0.51	100
Frog	0.64	0.63	0.64	100
Horse	0.56	0.54	0.55	100
Ship	0.64	0.78	0.70	100
Truck	0.67	0.56	0.61	100
accuracy			0.57	1000
macro avg	0.57	0.57	0.57	1000
weighted avg	0.57	0.57	0.57	1000

```
Classification Report for Decision Tree Depth 30:
============================================================
              precision    recall  f1-score   support

    Airplane       0.60      0.55      0.58       100
  Automobile       0.62      0.70      0.66       100
        Bird       0.48      0.48      0.48       100
         Cat       0.44      0.43      0.44       100
        Deer       0.52      0.53      0.53       100
         Dog       0.52      0.51      0.51       100
        Frog       0.64      0.63      0.63       100
       Horse       0.57      0.54      0.55       100
        Ship       0.64      0.78      0.70       100
       Truck       0.67      0.56      0.61       100

    accuracy                           0.57      1000
   macro avg       0.57      0.57      0.57      1000
weighted avg       0.57      0.57      0.57      1000

============================================================
```

Classification Report for Decision Tree Depth 40:

	precision	recall	f1-score	support
Airplane	0.60	0.55	0.58	100
Automobile	0.62	0.70	0.66	100
Bird	0.48	0.48	0.48	100
Cat	0.44	0.43	0.44	100
Deer	0.52	0.53	0.53	100
Dog	0.52	0.51	0.51	100
Frog	0.64	0.63	0.63	100
Horse	0.57	0.54	0.55	100
Ship	0.64	0.78	0.70	100
Truck	0.67	0.56	0.61	100
accuracy			0.57	1000
macro avg	0.57	0.57	0.57	1000
weighted avg	0.57	0.57	0.57	1000

Classification Report for Decision Tree Depth 50:

	precision	recall	f1-score	support
Airplane	0.60	0.55	0.58	100
Automobile	0.62	0.70	0.66	100
Bird	0.48	0.48	0.48	100
Cat	0.44	0.43	0.44	100
Deer	0.52	0.53	0.53	100
Dog	0.52	0.51	0.51	100
Frog	0.64	0.63	0.63	100
Horse	0.57	0.54	0.55	100
Ship	0.64	0.78	0.70	100
Truck	0.67	0.56	0.61	100
accuracy			0.57	1000
macro avg	0.57	0.57	0.57	1000
weighted avg	0.57	0.57	0.57	1000

```
Classification Report for Sklearn Decision Tree BP3:
=====================================================
              precision    recall  f1-score   support

    Airplane       0.59      0.54      0.56       100
  Automobile       0.60      0.72      0.65       100
        Bird       0.48      0.50      0.49       100
         Cat       0.46      0.45      0.45       100
        Deer       0.52      0.42      0.46       100
         Dog       0.43      0.47      0.45       100
        Frog       0.61      0.60      0.60       100
       Horse       0.52      0.48      0.50       100
        Ship       0.62      0.76      0.68       100
       Truck       0.65      0.53      0.59       100

    accuracy                           0.55      1000
   macro avg       0.55      0.55      0.54      1000
weighted avg       0.55      0.55      0.54      1000

=====================================================
```

Multi-Layer Perceptron (MLP)

Main Model

The Multi-Layer Perceptron (MLP) is a supervised learning model that uses traditional iterative training processes. The number of epochs used is 100. The learning rate is using the SGD optimizer with a value of 0.01 and momentum with a value of 0.9. The loss function used is Cross-Entropy Loss. The batch size is 64 using mini-batch gradient descent. We also use normalization.

Variants

We created two variants of the MLP model, both are very similar in implementation. One variant trains and tests by varying the depth of the model by adding and removing layers based on the main model's implementation. We use depths ranging from 1 to 20 layers. The other variant trains and tests using varying sizes of hidden layers based on the main model's implementation. We use hidden layers 128, 265, 512, 1024 and 2048.

Analysis of variants

Varying depth of model by adding or removing layers

Shallow networks such as layers 1-3 layers perform reasonably well, achieving the best accuracy at 69.3% for 1 layer. This indicates to us that shallow networks are sufficient enough for capturing the relationships between the input features and the output labels. Adding a few layers initially does not help much, if at all.

For very medium to deep networks such as 4-10 layers: As the depth increases, the performance stabilizes with no visible improvements. This might be due to vanishing gradients during training, where the gradient-based learning struggles to effectively update weights in the earlier layers.

For very deep networks such as 11-20 layers: the deeper the network, the less effective it seems to become, with accuracy dropping to 65.8% for a depth of 20. Overfitting is significant here. The model with more parameters tends to memorize the training data rather than learn useful generalizations, especially when there is not enough data to support such a large number of parameters. Optimization difficulties can also play a role, such as the difficulty of backpropagating through very deep networks, could also be affecting the performance of these very deep models.

From these observations, it is clear that increasing the number of layers in an MLP doesn't always lead to better results. The model's architecture and hyperparameters need to be matched to the complexity of the problem and the data available.

Varying the hidden layers size of model

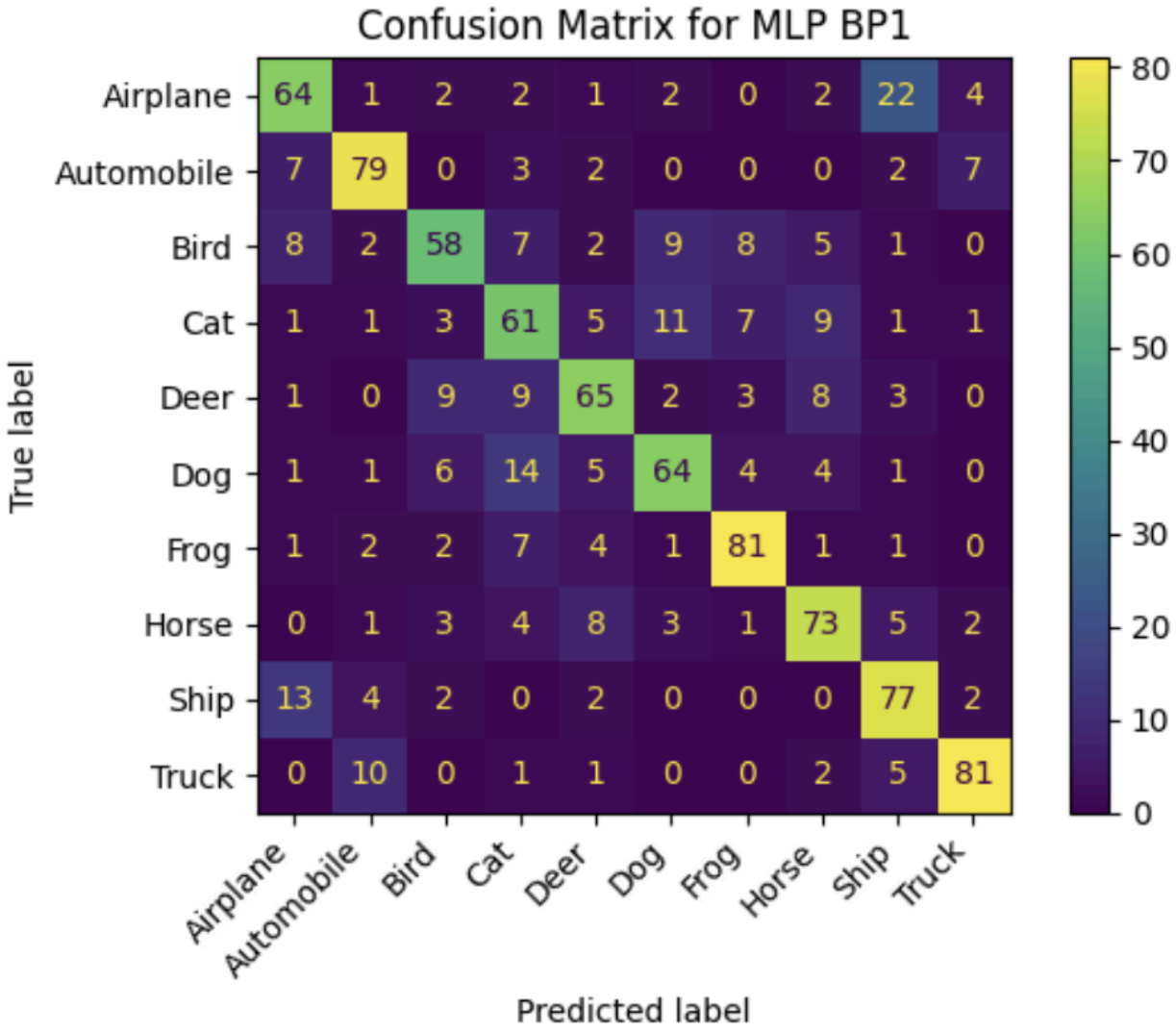
The hidden layer size of 512 provides the best accuracy of 71.1% and as we increased the hidden layer size beyond 512, we experienced diminishing returns and lower accuracy, due to overfitting and the inability to train effectively with the available data. Smaller hidden layers such as 128 or 256 layers are not as computationally expensive and train faster, but they lack the necessary capacity to fully capture the complexity of the data. Models with larger hidden layers such as 1024 or 2048 layers come with an increase in computational cost without a

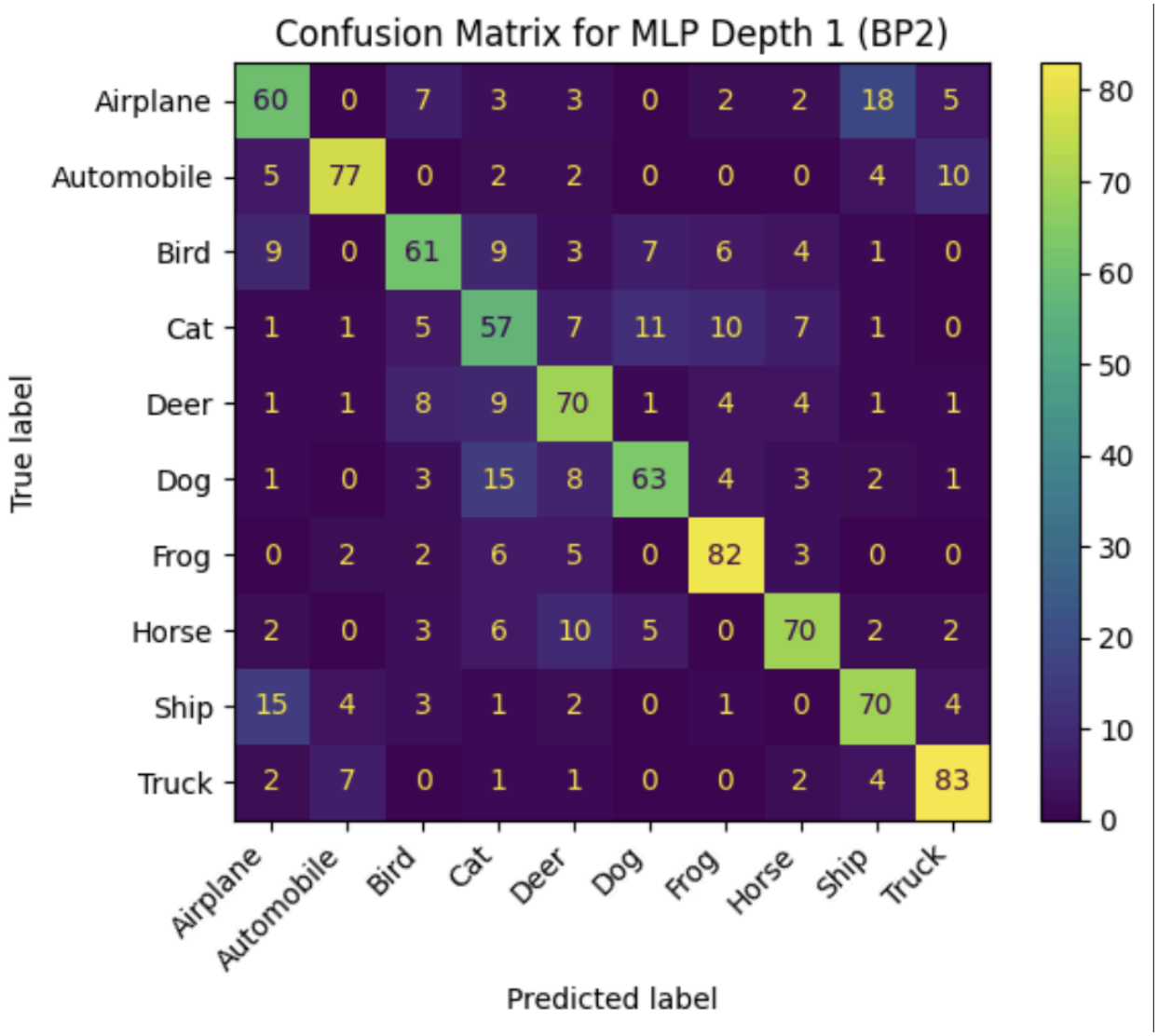
corresponding improvement in accuracy. The model's performance decreases, which suggests that these sizes introduce unnecessary complexity.

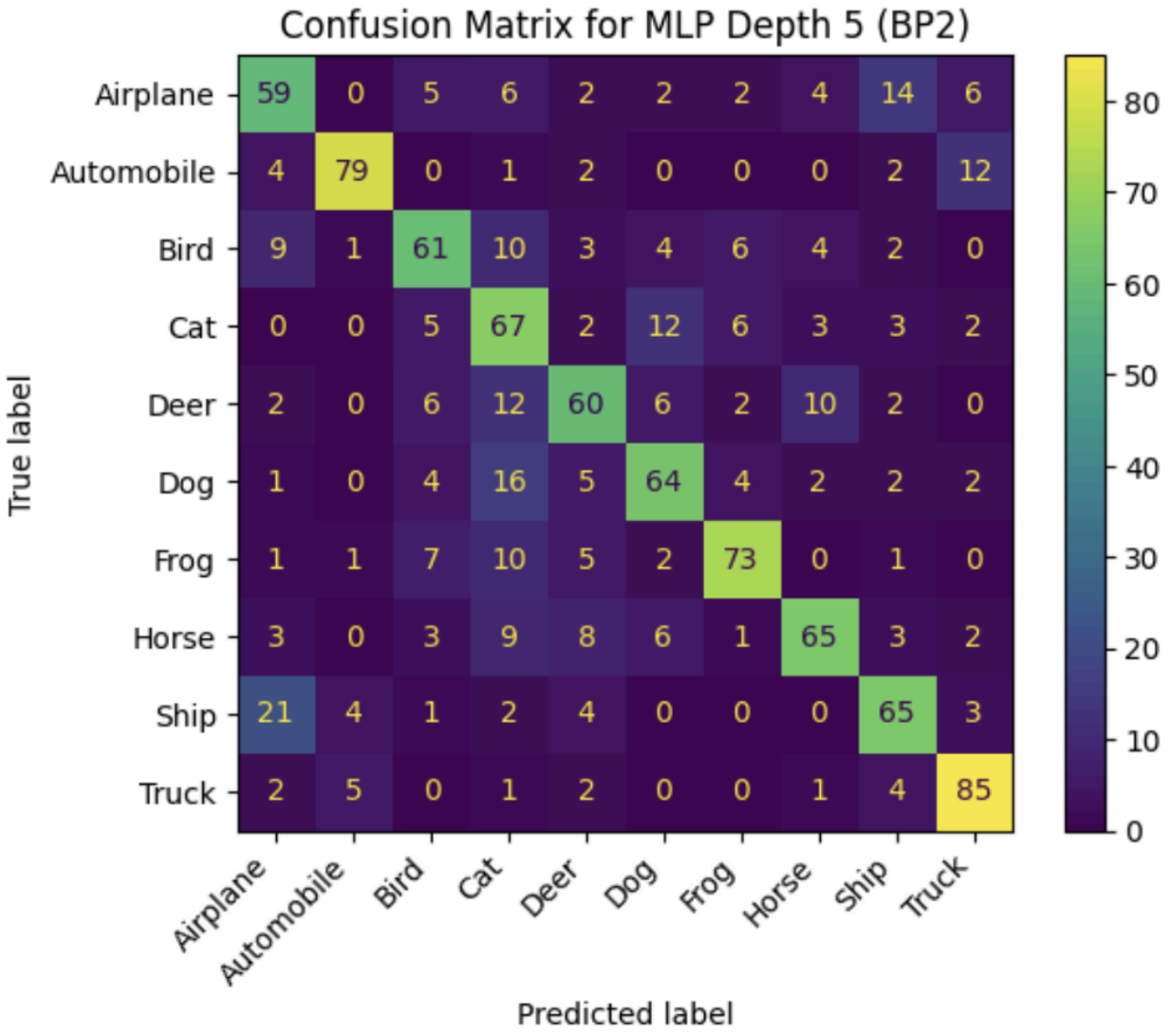
From these observations, it is clear that increasing the number of hidden layers does not always improve model performance. Finding the right balance between complexity and generalization is crucial for achieving optimal performance while managing computational costs effectively.

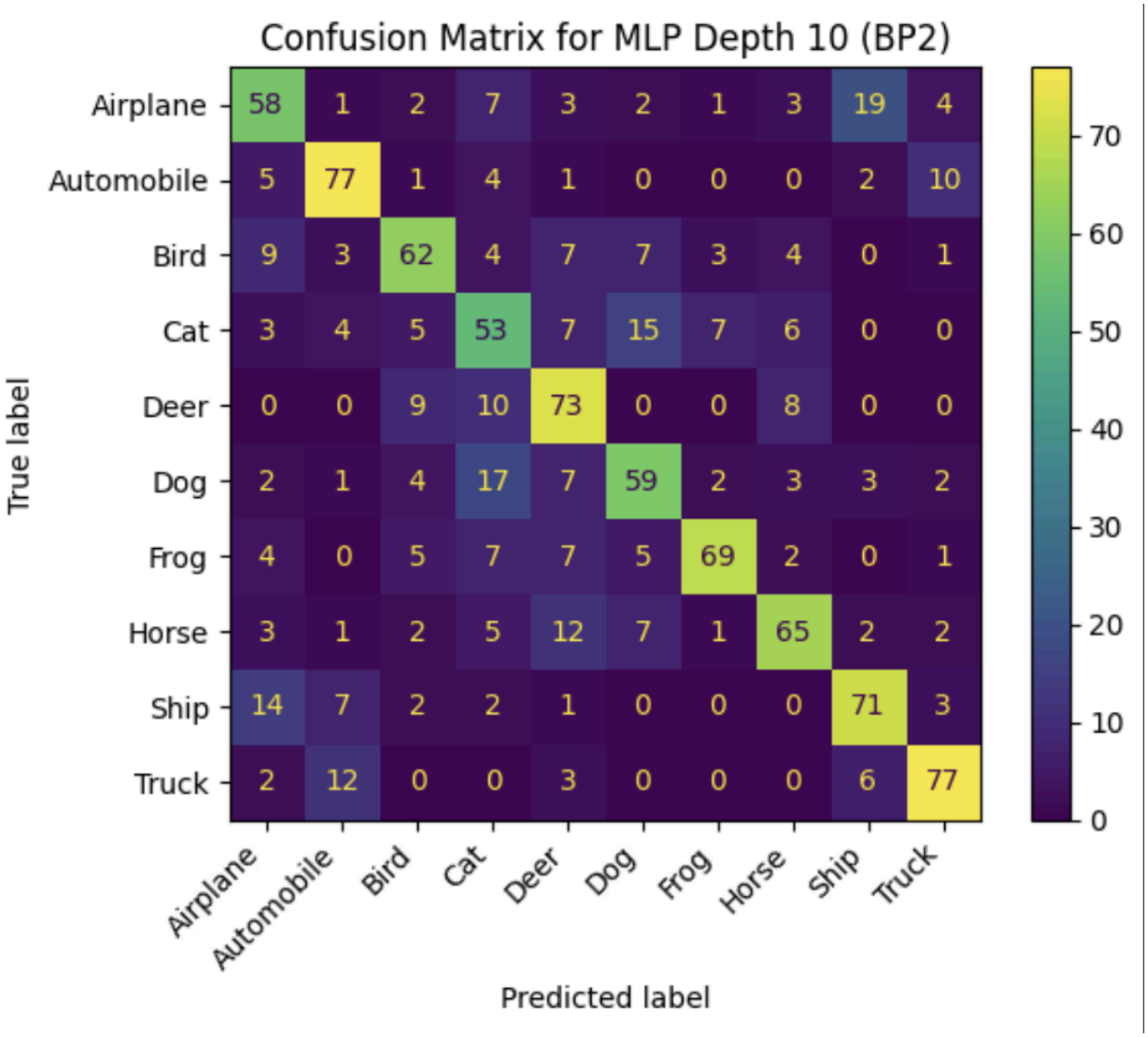
Evaluation

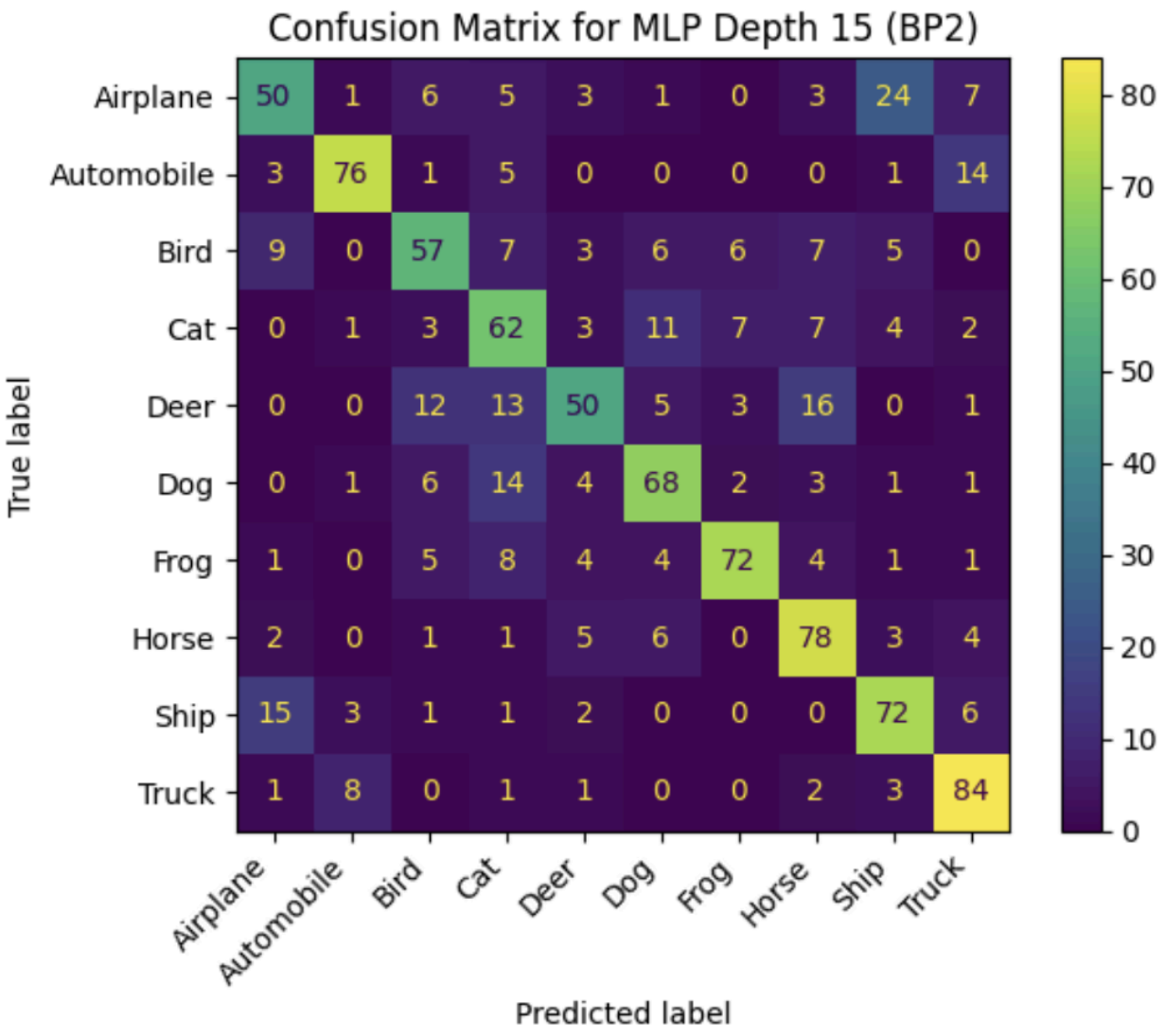
Confusion Matrices:

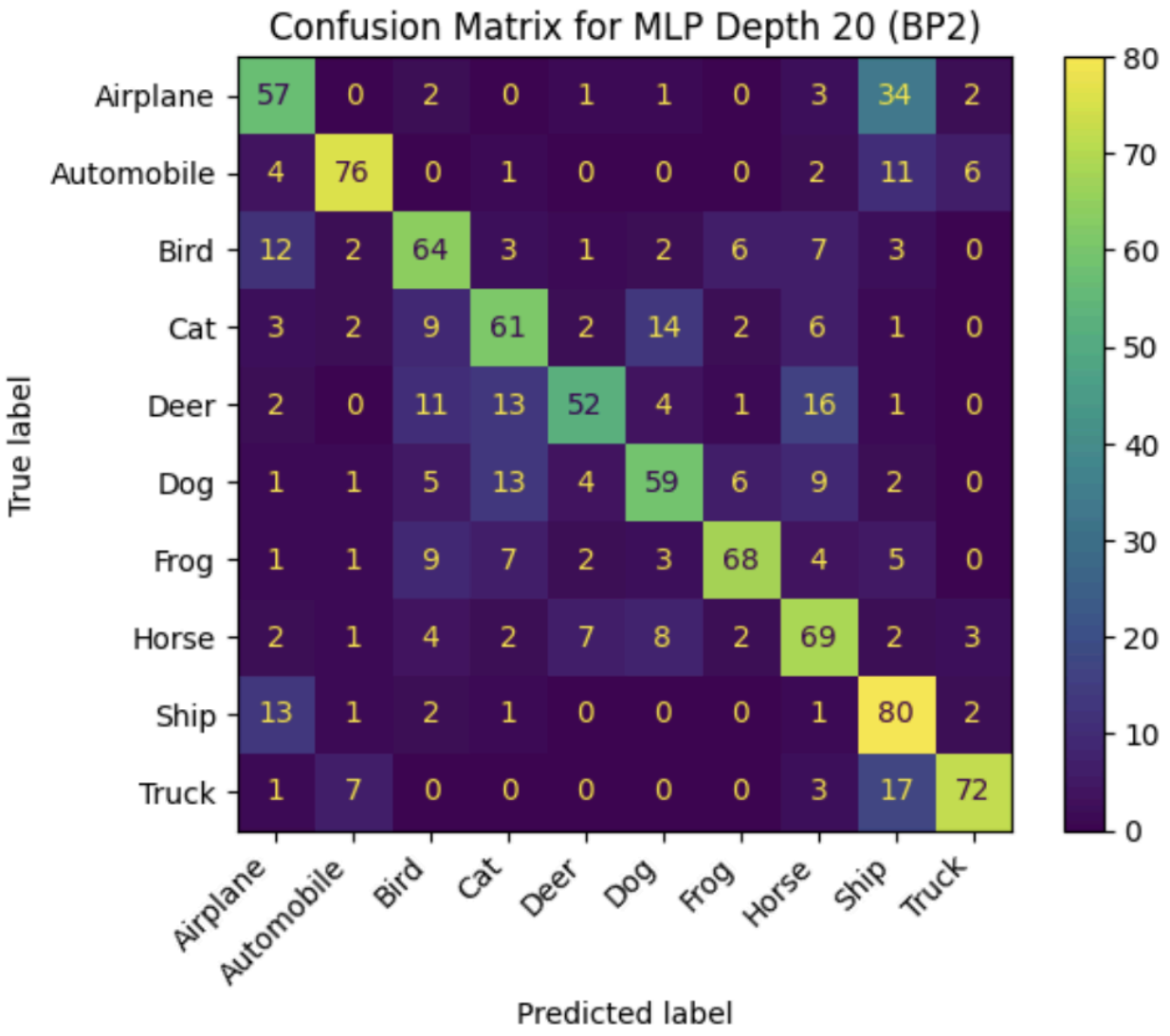


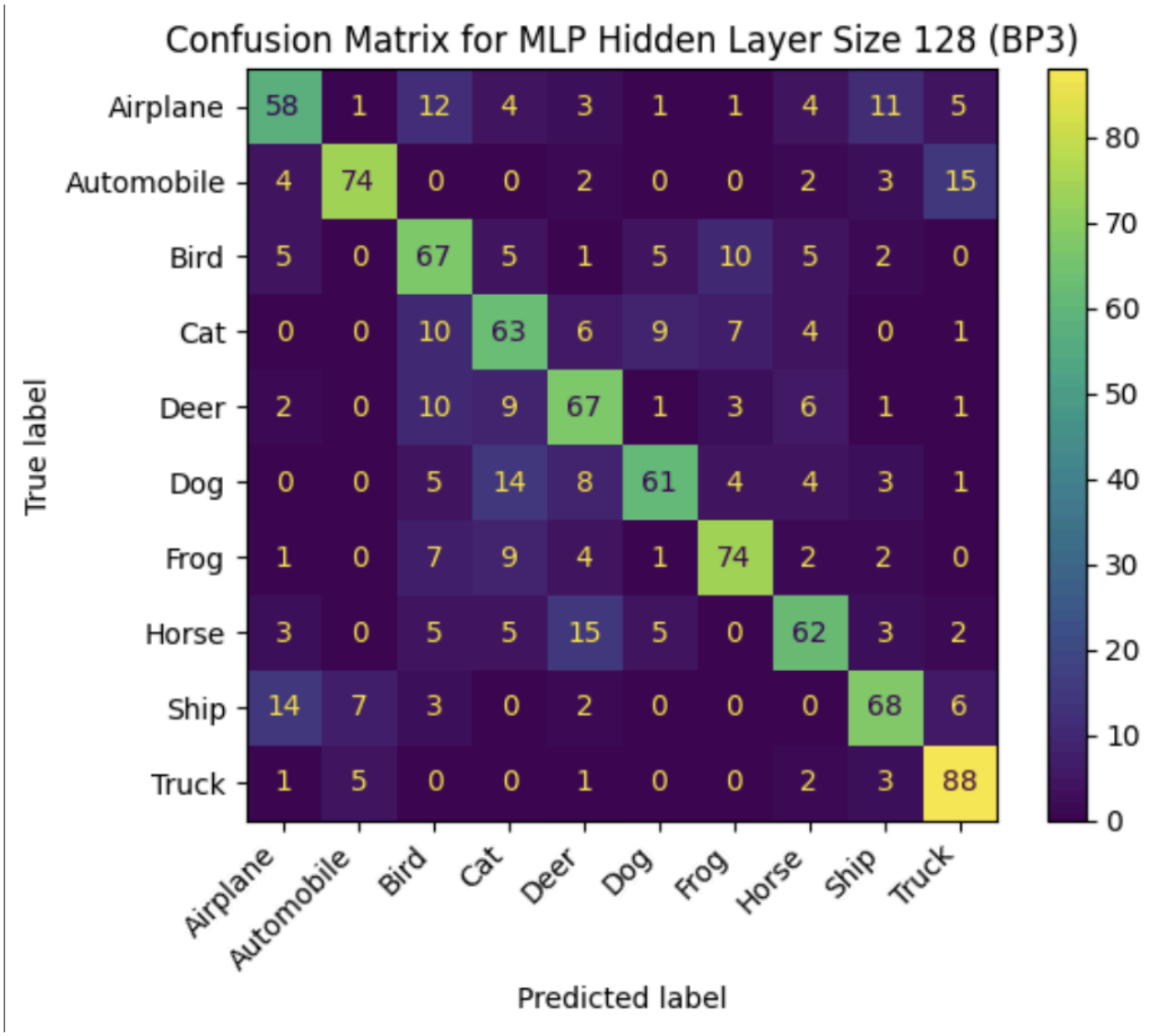

Confusion Matrix for MLP Depth 1 (BP2)
Airplane 60 0 7 3 3 0 2 2 18 5
Automobile 5 77 0 2 2 0 0 0 4 10
Bird 9 0 61 9 3 7 6 4 1 0
Cat 1 1 5 57 7 11 10 7 1 0
Deer 1 1 8 9 70 1 4 4 1 1
Dog 1 0 3 15 8 63 4 3 2 1
Frog 0 2 2 6 5 0 82 3 0 0
Horse 2 0 3 6 10 5 0 70 2 2
Ship 15 4 3 1 2 0 1 0 70 4
Truck 2 7 0 1 1 0 0 2 4 83
True label
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
Predicted label
80
70
60
50
40
30
20
10
0

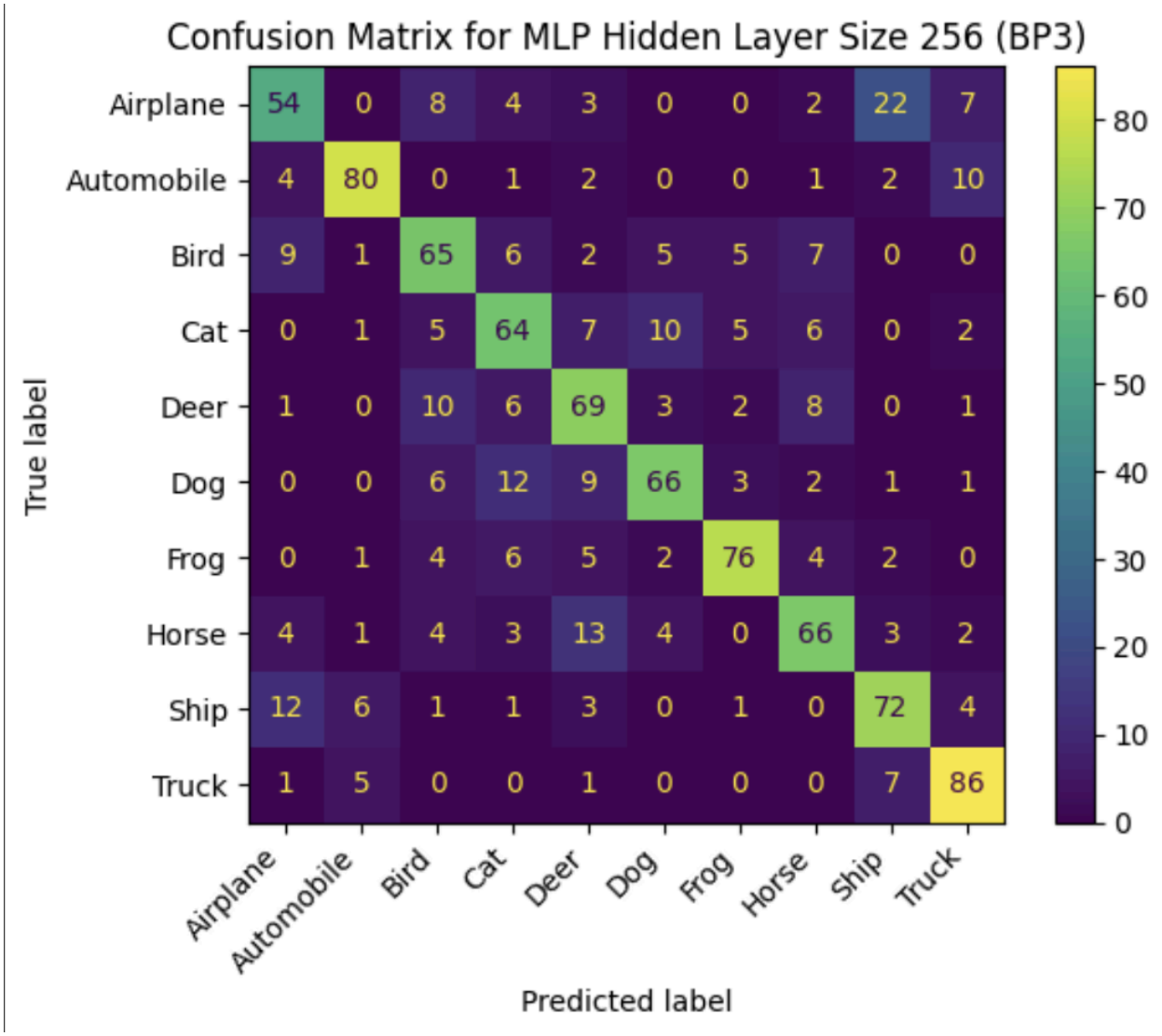

Confusion Matrix for MLP Depth 5 (BP2)
Airplane 59 0 5 6 2 2 2 4 14 6
Automobile 4 79 0 1 2 0 0 0 2 12
Bird 9 1 61 10 3 4 6 4 2 0
Cat 0 0 5 67 2 12 6 3 3 2
Deer 2 0 6 12 60 6 2 10 2 0
Dog 1 0 4 16 5 64 4 2 2 2
Frog 1 1 7 10 5 2 73 0 1 0
Horse 3 0 3 9 8 6 1 65 3 2
Ship 21 4 1 2 4 0 0 0 65 3
Truck 2 5 0 1 2 0 0 1 4 85
True label
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
Predicted label
80
70
60
50
40
30
20
10
0

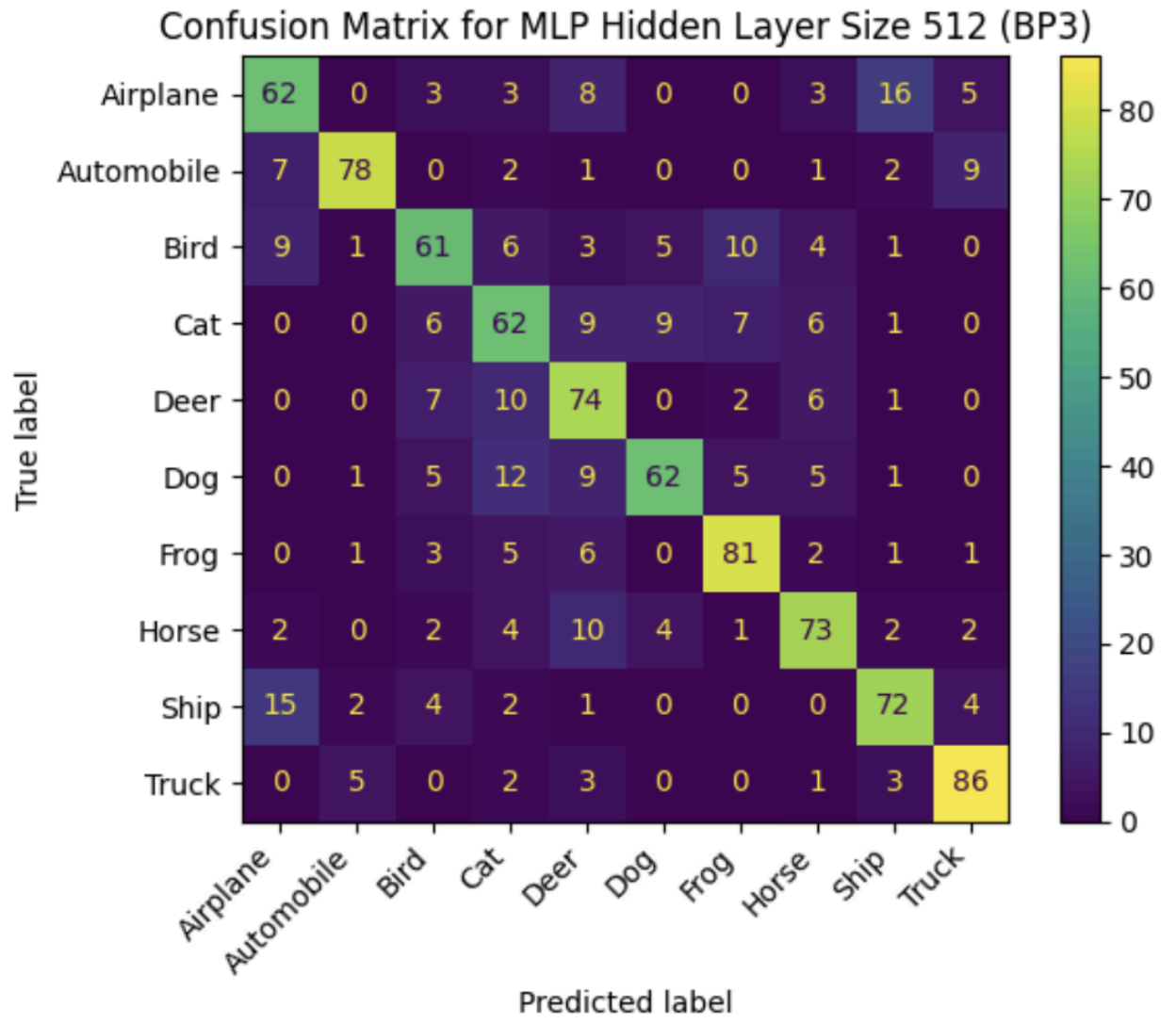

Confusion Matrix for MLP Depth 10 (BP2)
Airplane 58 1 2 7 3 2 1 3 19 4
Automobile 5 77 1 4 1 0 0 0 2 10
Bird 9 3 62 4 7 7 3 4 0 1
Cat 3 4 5 53 7 15 7 6 0 0
Deer 0 0 9 10 73 0 0 8 0 0
Dog 2 1 4 17 7 59 2 3 3 2
Frog 4 0 5 7 7 5 69 2 0 1
Horse 3 1 2 5 12 7 1 65 2 2
Ship 14 7 2 2 1 0 0 0 71 3
Truck 2 12 0 0 3 0 0 0 6 77
True label
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
Predicted label
70
60
50
40
30
20
10
0

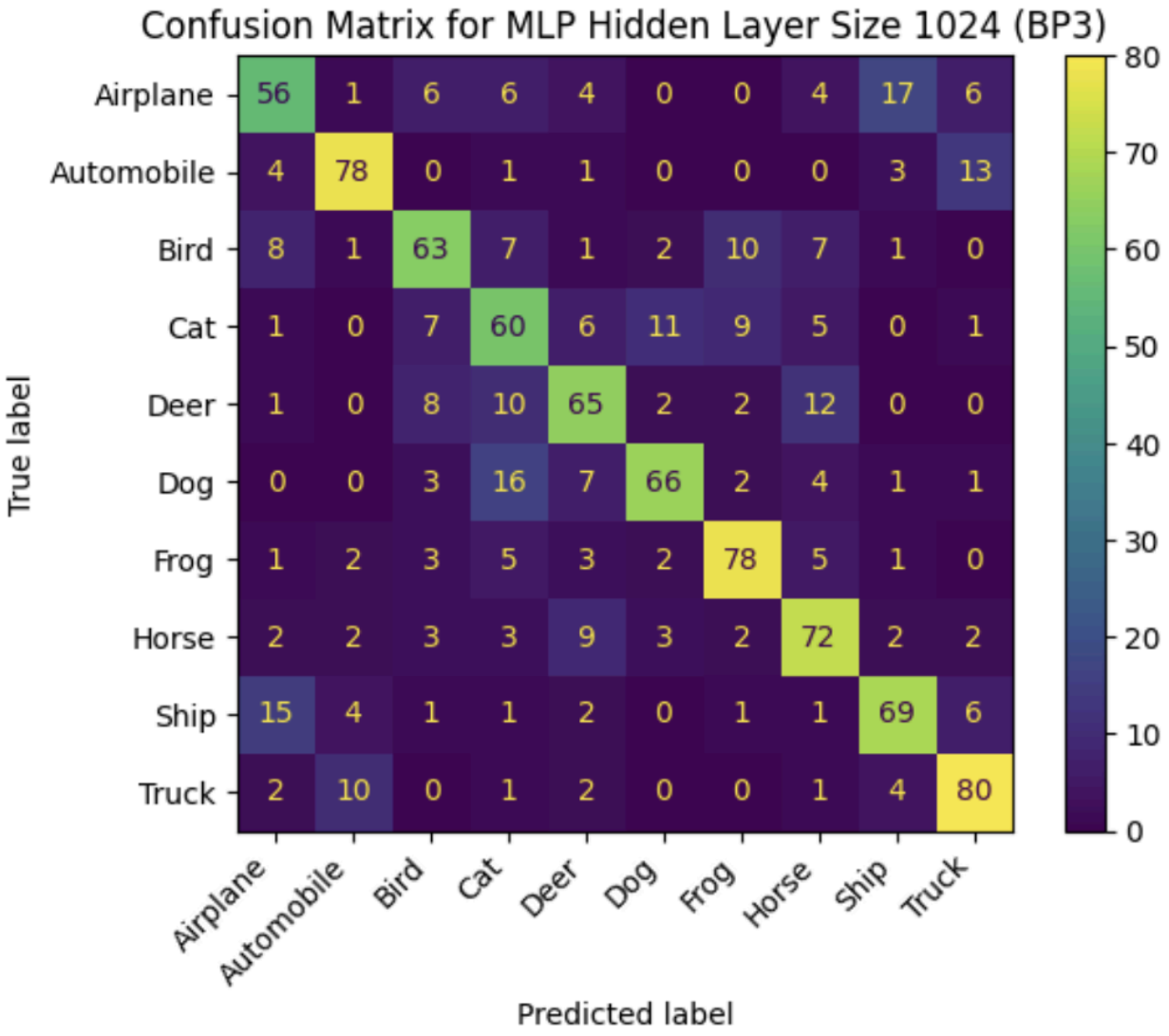

Confusion Matrix for MLP Depth 15 (BP2)
Airplane 50 1 6 5 3 1 0 3 24 7
Automobile 3 76 1 5 0 0 0 0 1 14
Bird 9 0 57 7 3 6 6 7 5 0
Cat 0 1 3 62 3 11 7 7 4 2
Deer 0 0 12 13 50 5 3 16 0 1
Dog 0 1 6 14 4 68 2 3 1 1
Frog 1 0 5 8 4 4 72 4 1 1
Horse 2 0 1 1 5 6 0 78 3 4
Ship 15 3 1 1 2 0 0 0 72 6
Truck 1 8 0 1 1 0 0 2 3 84
True label
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
Predicted label
80
70
60
50
40
30
20
10
0

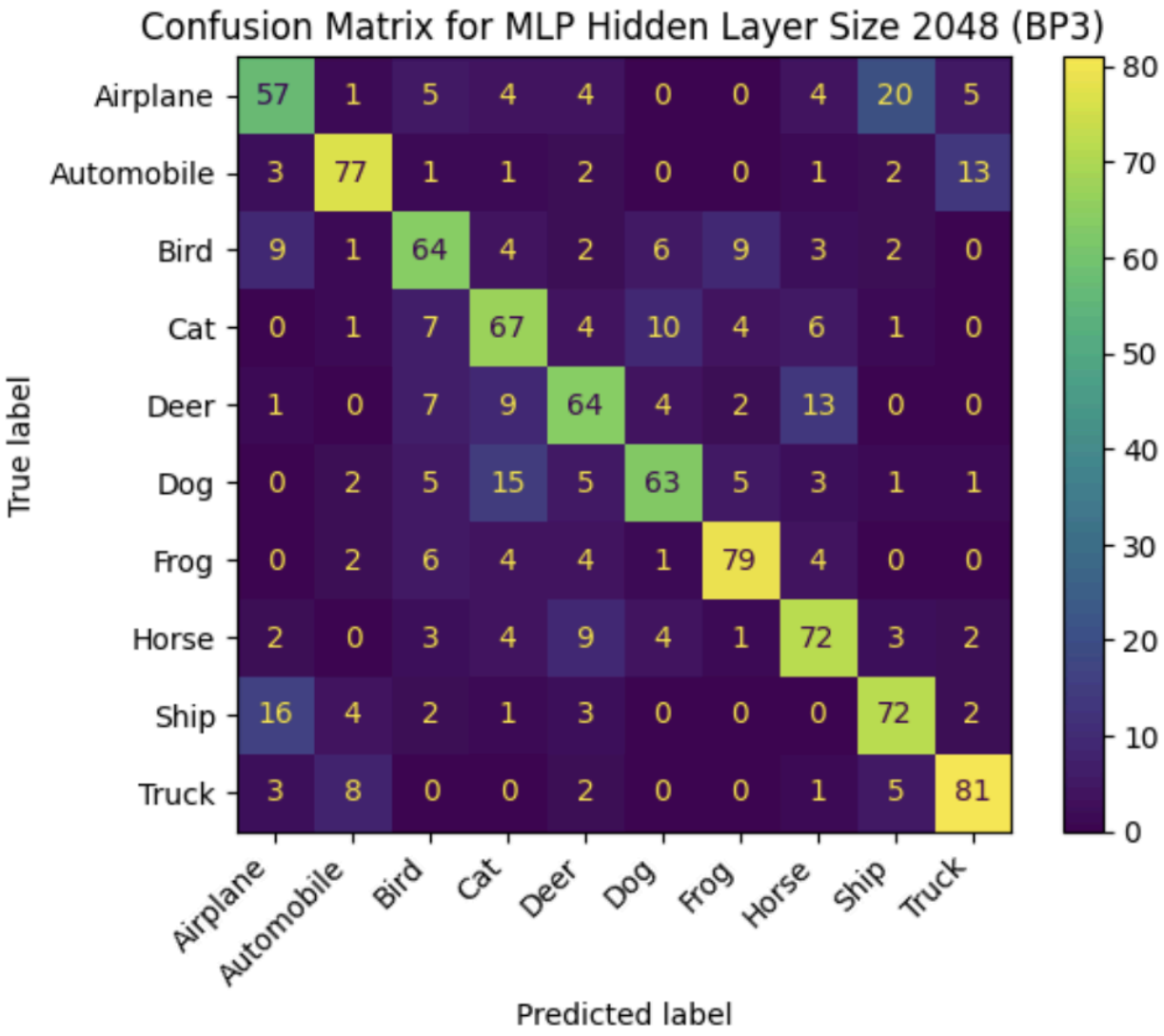

Confusion Matrix for MLP Depth 20 (BP2)
Airplane 57 0 2 0 1 1 0 3 34 2
Automobile 4 76 0 1 0 0 0 2 11 6
Bird 12 2 64 3 1 2 6 7 3 0
Cat 3 2 9 61 2 14 2 6 1 0
Deer 2 0 11 13 52 4 1 16 1 0
Dog 1 1 5 13 4 59 6 9 2 0
Frog 1 1 9 7 2 3 68 4 5 0
Horse 2 1 4 2 7 8 2 69 2 3
Ship 13 1 2 1 0 0 0 1 80 2
Truck 1 7 0 0 0 0 0 3 17 72
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
True label
Predicted label
80
70
60
50
40
30
20
10
0


Confusion Matrix for MLP Hidden Layer Size 128 (BP3)
True label
Predicted label
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
58 1 12 4 3 1 1 4 11 5
4 74 0 0 2 0 0 2 3 15
5 0 67 5 1 5 10 5 2 0
0 0 10 63 6 9 7 4 0 1
2 0 10 9 67 1 3 6 1 1
0 0 5 14 8 61 4 4 3 1
1 0 7 9 4 1 74 2 2 0
3 0 5 5 15 5 0 62 3 2
14 7 3 0 2 0 0 0 68 6
1 5 0 0 1 0 0 2 3 88
80
70
60
50
40
30
20
10
0


Confusion Matrix for MLP Hidden Layer Size 256 (BP3)
Airplane 54 0 8 4 3 0 0 2 22 7
Automobile 4 80 0 1 2 0 0 1 2 10
Bird 9 1 65 6 2 5 5 7 0 0
Cat 0 1 5 64 7 10 5 6 0 2
Deer 1 0 10 6 69 3 2 8 0 1
Dog 0 0 6 12 9 66 3 2 1 1
Frog 0 1 4 6 5 2 76 4 2 0
Horse 4 1 4 3 13 4 0 66 3 2
Ship 12 6 1 1 3 0 1 0 72 4
Truck 1 5 0 0 1 0 0 0 7 86
True label
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
Predicted label
80
70
60
50
40
30
20
10
0


Confusion Matrix for MLP Hidden Layer Size 512 (BP3)
Airplane 62 0 3 3 8 0 0 3 16 5
Automobile 7 78 0 2 1 0 0 1 2 9
Bird 9 1 61 6 3 5 10 4 1 0
Cat 0 0 6 62 9 9 7 6 1 0
Deer 0 0 7 10 74 0 2 6 1 0
Dog 0 1 5 12 9 62 5 5 1 0
Frog 0 1 3 5 6 0 81 2 1 1
Horse 2 0 2 4 10 4 1 73 2 2
Ship 15 2 4 2 1 0 0 0 72 4
Truck 0 5 0 2 3 0 0 1 3 86
True label
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
Predicted label
80
70
60
50
40
30
20
10
0


Confusion Matrix for MLP Hidden Layer Size 1024 (BP3)
Airplane 56 1 6 6 4 0 0 4 17 6
Automobile 4 78 0 1 1 0 0 0 3 13
Bird 8 1 63 7 1 2 10 7 1 0
Cat 1 0 7 60 6 11 9 5 0 1
Deer 1 0 8 10 65 2 2 12 0 0
Dog 0 0 3 16 7 66 2 4 1 1
Frog 1 2 3 5 3 2 78 5 1 0
Horse 2 2 3 3 9 3 2 72 2 2
Ship 15 4 1 1 2 0 1 1 69 6
Truck 2 10 0 1 2 0 0 1 4 80
True label
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
Predicted label
80
70
60
50
40
30
20
10
0


Confusion Matrix for MLP Hidden Layer Size 2048 (BP3)
Airplane 57 1 5 4 4 0 0 4 20 5
Automobile 3 77 1 1 2 0 0 1 2 13
Bird 9 1 64 4 2 6 9 3 2 0
Cat 0 1 7 67 4 10 4 6 1 0
Deer 1 0 7 9 64 4 2 13 0 0
Dog 0 2 5 15 5 63 5 3 1 1
Frog 0 2 6 4 4 1 79 4 0 0
Horse 2 0 3 4 9 4 1 72 3 2
Ship 16 4 2 1 3 0 0 0 72 2
Truck 3 8 0 0 2 0 0 1 5 81
True label
Airplane
Automobile
Bird
Cat
Deer
Dog
Frog
Horse
Ship
Truck
Predicted label
80
70
60
50
40
30
20
10
0

Classification Reports:

Classification Report for MLP Depth 1 (BP2):

	precision	recall	f1-score	support
Airplane	0.62	0.60	0.61	100
Automobile	0.84	0.77	0.80	100
Bird	0.66	0.61	0.64	100
Cat	0.52	0.57	0.55	100
Deer	0.63	0.70	0.66	100
Dog	0.72	0.63	0.67	100
Frog	0.75	0.82	0.78	100
Horse	0.74	0.70	0.72	100
Ship	0.68	0.70	0.69	100
Truck	0.78	0.83	0.81	100
accuracy			0.69	1000
macro avg	0.70	0.69	0.69	1000
weighted avg	0.70	0.69	0.69	1000

Classification Report for MLP Depth 5 (BP2):

	precision	recall	f1-score	support
Airplane	0.58	0.59	0.58	100
Automobile	0.88	0.79	0.83	100
Bird	0.66	0.61	0.64	100
Cat	0.50	0.67	0.57	100
Deer	0.65	0.60	0.62	100
Dog	0.67	0.64	0.65	100
Frog	0.78	0.73	0.75	100
Horse	0.73	0.65	0.69	100
Ship	0.66	0.65	0.66	100
Truck	0.76	0.85	0.80	100
accuracy			0.68	1000
macro avg	0.69	0.68	0.68	1000
weighted avg	0.69	0.68	0.68	1000

Classification Report for MLP Depth 10 (BP2):

	precision	recall	f1-score	support
Airplane	0.58	0.58	0.58	100
Automobile	0.73	0.77	0.75	100
Bird	0.67	0.62	0.65	100
Cat	0.49	0.53	0.51	100
Deer	0.60	0.73	0.66	100
Dog	0.62	0.59	0.61	100
Frog	0.83	0.69	0.75	100
Horse	0.71	0.65	0.68	100
Ship	0.69	0.71	0.70	100
Truck	0.77	0.77	0.77	100
accuracy			0.66	1000
macro avg	0.67	0.66	0.67	1000
weighted avg	0.67	0.66	0.67	1000

Classification Report for MLP Depth 15 (BP2):

	precision	recall	f1-score	support
Airplane	0.62	0.50	0.55	100
Automobile	0.84	0.76	0.80	100
Bird	0.62	0.57	0.59	100
Cat	0.53	0.62	0.57	100
Deer	0.67	0.50	0.57	100
Dog	0.67	0.68	0.68	100
Frog	0.80	0.72	0.76	100
Horse	0.65	0.78	0.71	100
Ship	0.63	0.72	0.67	100
Truck	0.70	0.84	0.76	100
accuracy			0.67	1000
macro avg	0.67	0.67	0.67	1000
weighted avg	0.67	0.67	0.67	1000

Classification Report for MLP Depth 20 (BP2):

	precision	recall	f1-score	support
Airplane	0.59	0.57	0.58	100
Automobile	0.84	0.76	0.80	100
Bird	0.60	0.64	0.62	100
Cat	0.60	0.61	0.61	100
Deer	0.75	0.52	0.62	100
Dog	0.65	0.59	0.62	100
Frog	0.80	0.68	0.74	100
Horse	0.57	0.69	0.63	100
Ship	0.51	0.80	0.62	100
Truck	0.85	0.72	0.78	100
accuracy			0.66	1000
macro avg	0.68	0.66	0.66	1000
weighted avg	0.68	0.66	0.66	1000

Classification Report for MLP Hidden Layer Size 128 (BP3):

	precision	recall	f1-score	support
Airplane	0.66	0.58	0.62	100
Automobile	0.85	0.74	0.79	100
Bird	0.56	0.67	0.61	100
Cat	0.58	0.63	0.60	100
Deer	0.61	0.67	0.64	100
Dog	0.73	0.61	0.67	100
Frog	0.75	0.74	0.74	100
Horse	0.68	0.62	0.65	100
Ship	0.71	0.68	0.69	100
Truck	0.74	0.88	0.80	100
accuracy			0.68	1000
macro avg	0.69	0.68	0.68	1000
weighted avg	0.69	0.68	0.68	1000

Classification Report for MLP Hidden Layer Size 256 (BP3):

	precision	recall	f1-score	support
Airplane	0.64	0.54	0.58	100
Automobile	0.84	0.80	0.82	100
Bird	0.63	0.65	0.64	100
Cat	0.62	0.64	0.63	100
Deer	0.61	0.69	0.64	100
Dog	0.73	0.66	0.69	100
Frog	0.83	0.76	0.79	100
Horse	0.69	0.66	0.67	100
Ship	0.66	0.72	0.69	100
Truck	0.76	0.86	0.81	100
accuracy			0.70	1000
macro avg	0.70	0.70	0.70	1000
weighted avg	0.70	0.70	0.70	1000

Classification Report for MLP Hidden Layer Size 512 (BP3):

	precision	recall	f1-score	support
Airplane	0.65	0.62	0.64	100
Automobile	0.89	0.78	0.83	100
Bird	0.67	0.61	0.64	100
Cat	0.57	0.62	0.60	100
Deer	0.60	0.74	0.66	100
Dog	0.78	0.62	0.69	100
Frog	0.76	0.81	0.79	100
Horse	0.72	0.73	0.73	100
Ship	0.72	0.72	0.72	100
Truck	0.80	0.86	0.83	100
accuracy			0.71	1000
macro avg	0.72	0.71	0.71	1000
weighted avg	0.72	0.71	0.71	1000

Classification Report for MLP Hidden Layer Size 1024 (BP3):

	precision	recall	f1-score	support
Airplane	0.62	0.56	0.59	100
Automobile	0.80	0.78	0.79	100
Bird	0.67	0.63	0.65	100
Cat	0.55	0.60	0.57	100
Deer	0.65	0.65	0.65	100
Dog	0.77	0.66	0.71	100
Frog	0.75	0.78	0.76	100
Horse	0.65	0.72	0.68	100
Ship	0.70	0.69	0.70	100
Truck	0.73	0.80	0.77	100
accuracy			0.69	1000
macro avg	0.69	0.69	0.69	1000
weighted avg	0.69	0.69	0.69	1000

Classification Report for MLP Hidden Layer Size 2048 (BP3):

	precision	recall	f1-score	support
Airplane	0.63	0.57	0.60	100
Automobile	0.80	0.77	0.79	100
Bird	0.64	0.64	0.64	100
Cat	0.61	0.67	0.64	100
Deer	0.65	0.64	0.64	100
Dog	0.72	0.63	0.67	100
Frog	0.79	0.79	0.79	100
Horse	0.67	0.72	0.70	100
Ship	0.68	0.72	0.70	100
Truck	0.78	0.81	0.79	100
accuracy			0.70	1000
macro avg	0.70	0.70	0.70	1000
weighted avg	0.70	0.70	0.70	1000

Note: The rest of the confusion matrices and classification reports can be found in the code in the relevant section.

Convolutional Neural Network (CNN)

Main Model

The Convolutional Neural Network (CNN) is a supervised learning model that uses traditional iterative training processes such as forward pass, loss calculation, backpropagation, weight update and epoch loops. The number of epochs used is 25. The learning rate is using the SGD optimizer with a value of 0.01 and momentum with a value of 0.9. The loss function used is Cross-Entropy Loss. The batch size is 256 using mini-batch gradient descent. We also use batch normalization and dropout layers with a dropout rate of 0.5.

Variants

We created three variants of the main CNN model, the first variant removes layers and trains and tests. We use 5 instead of 8 convolutional layers The other two train and test using varying sizes of kernels in the convolutional layers. We use kernel sizes: 2x2 and 7x7.

Analysis of variants

Removal of convolutional layers

With the removal of convolutional layers from 8 to 5, we did not see any notable difference in the confusion matrices and classification reports. We can deduce that the additional layers were not necessary and that the reduced model was able to learn sufficient features with fewer layers with better performance and in less time. The CIFAR-10 dataset may have been too simple of a dataset for this model to warrant more layers.

Larger kernels (7x7)

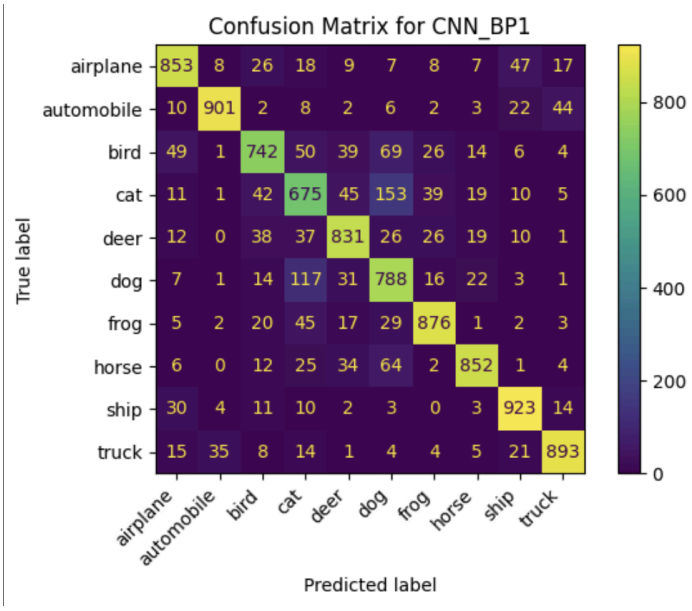
With larger kernels (7x7) up from (3x3) there was a very slight decrease (83% to 79%) in accuracy in the classification reports. As the kernels increased we can deduce more broader information about the data that was found but we lost some local information about the data. The lower accuracy can also be attributed to the fact that there are less convolutional layers due to using larger kernel sizes.

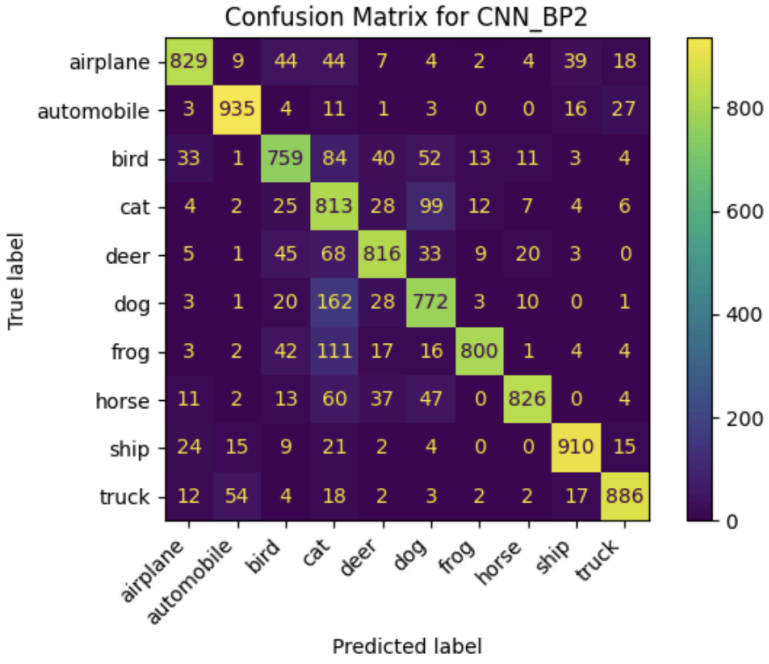
Smaller Kernels (2x2)

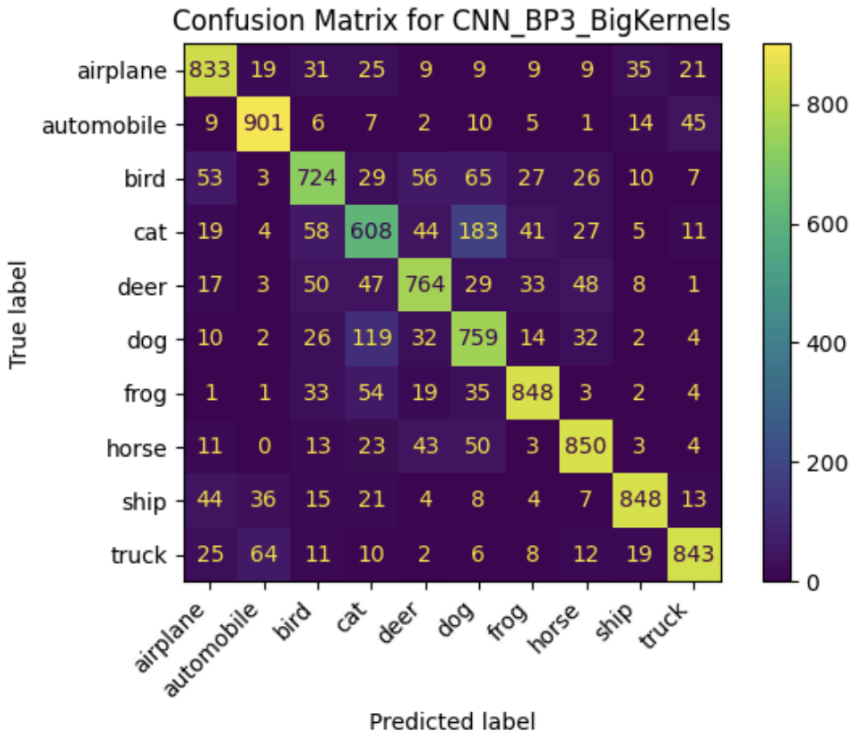
With smaller kernels (2x2) down from (3x3) there was a very slight decrease in accuracy (83% to 80%) in the classification reports. As kernels decreased we can deduce that less details were captured leading to lower accuracy and worse generalization although very slight. It also ran slightly faster.

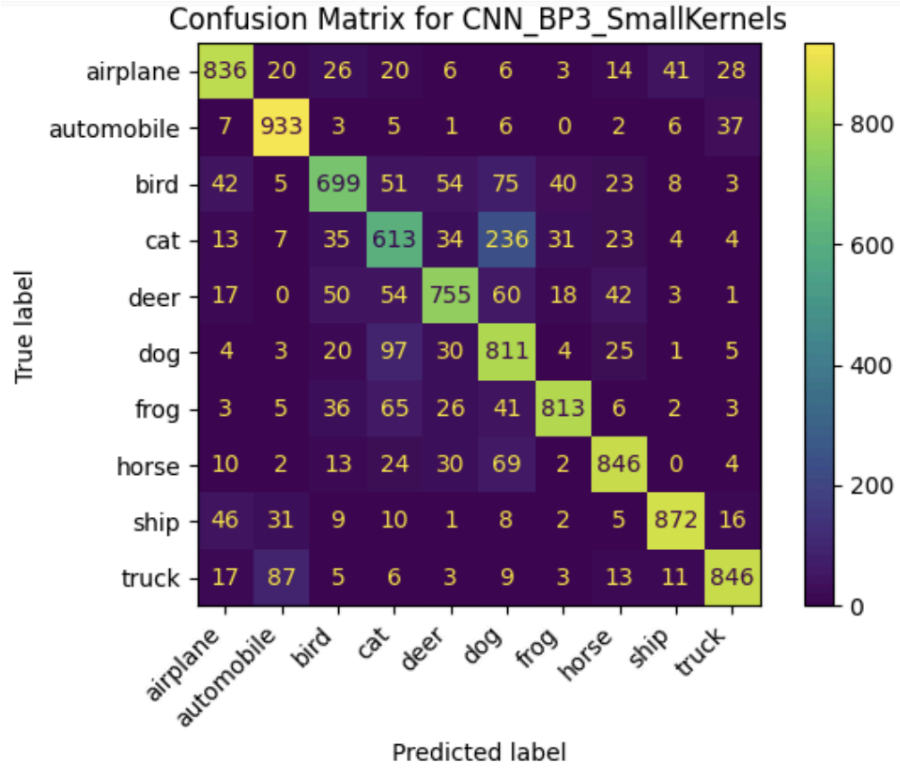
Evaluation

Confusion Matrices:


Confusion Matrix for CNN_BP1
airplane 853 8 26 18 9 7 8 7 47 17
automobile 10 901 2 8 2 6 2 3 22 44
bird 49 1 742 50 39 69 26 14 6 4
cat 11 1 42 675 45 153 39 19 10 5
deer 12 0 38 37 831 26 26 19 10 1
dog 7 1 14 117 31 788 16 22 3 1
frog 5 2 20 45 17 29 876 1 2 3
horse 6 0 12 25 34 64 2 852 1 4
ship 30 4 11 10 2 3 0 3 923 14
truck 15 35 8 14 1 4 4 5 21 893
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
800
600
400
200
0


Confusion Matrix for CNN_BP2
airplane 829 9 44 44 7 4 2 4 39 18
automobile 3 935 4 11 1 3 0 0 16 27
bird 33 1 759 84 40 52 13 11 3 4
cat 4 2 25 813 28 99 12 7 4 6
deer 5 1 45 68 816 33 9 20 3 0
dog 3 1 20 162 28 772 3 10 0 1
frog 3 2 42 111 17 16 800 1 4 4
horse 11 2 13 60 37 47 0 826 0 4
ship 24 15 9 21 2 4 0 0 910 15
truck 12 54 4 18 2 3 2 2 17 886
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
800
600
400
200
0


Confusion Matrix for CNN_BP3_BigKernels
airplane 833 19 31 25 9 9 9 9 35 21
automobile 9 901 6 7 2 10 5 1 14 45
bird 53 3 724 29 56 65 27 26 10 7
cat 19 4 58 608 44 183 41 27 5 11
deer 17 3 50 47 764 29 33 48 8 1
dog 10 2 26 119 32 759 14 32 2 4
frog 1 1 33 54 19 35 848 3 2 4
horse 11 0 13 23 43 50 3 850 3 4
ship 44 36 15 21 4 8 4 7 848 13
truck 25 64 11 10 2 6 8 12 19 843
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
800
600
400
200
0


Confusion Matrix for CNN_BP3_SmallKernels
airplane 836 20 26 20 6 6 3 14 41 28
automobile 7 933 3 5 1 6 0 2 6 37
bird 42 5 699 51 54 75 40 23 8 3
cat 13 7 35 613 34 236 31 23 4 4
deer 17 0 50 54 755 60 18 42 3 1
dog 4 3 20 97 30 811 4 25 1 5
frog 3 5 36 65 26 41 813 6 2 3
horse 10 2 13 24 30 69 2 846 0 4
ship 46 31 9 10 1 8 2 5 872 16
truck 17 87 5 6 3 9 3 13 11 846
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
800
600
400
200
0

Classification Reports:

Classification Report for CNN_BP1:

	precision	recall	f1-score	support
airplane	0.85	0.85	0.85	1000
automobile	0.95	0.90	0.92	1000
bird	0.81	0.74	0.77	1000
cat	0.68	0.68	0.68	1000
deer	0.82	0.83	0.83	1000
dog	0.69	0.79	0.73	1000
frog	0.88	0.88	0.88	1000
horse	0.90	0.85	0.88	1000
ship	0.88	0.92	0.90	1000
truck	0.91	0.89	0.90	1000
accuracy			0.83	10000
macro avg	0.84	0.83	0.83	10000
weighted avg	0.84	0.83	0.83	10000

Classification Report for CNN_BP2:

	precision	recall	f1-score	support
airplane	0.89	0.83	0.86	1000
automobile	0.91	0.94	0.92	1000
bird	0.79	0.76	0.77	1000
cat	0.58	0.81	0.68	1000
deer	0.83	0.82	0.83	1000
dog	0.75	0.77	0.76	1000
frog	0.95	0.80	0.87	1000
horse	0.94	0.83	0.88	1000
ship	0.91	0.91	0.91	1000
truck	0.92	0.89	0.90	1000
accuracy			0.83	10000
macro avg	0.85	0.83	0.84	10000
weighted avg	0.85	0.83	0.84	10000

Classification Report for CNN_BP3_BigKernels:

	precision	recall	f1-score	support
airplane	0.82	0.83	0.82	1000
automobile	0.87	0.90	0.89	1000
bird	0.75	0.72	0.74	1000
cat	0.64	0.61	0.63	1000
deer	0.78	0.76	0.77	1000
dog	0.66	0.76	0.70	1000
frog	0.85	0.85	0.85	1000
horse	0.84	0.85	0.84	1000
ship	0.90	0.85	0.87	1000
truck	0.88	0.84	0.86	1000
accuracy			0.80	10000
macro avg	0.80	0.80	0.80	10000
weighted avg	0.80	0.80	0.80	10000

Classification Report for CNN_BP3_SmallKernels:

	precision	recall	f1-score	support
airplane	0.84	0.84	0.84	1000
automobile	0.85	0.93	0.89	1000
bird	0.78	0.70	0.74	1000
cat	0.65	0.61	0.63	1000
deer	0.80	0.76	0.78	1000
dog	0.61	0.81	0.70	1000
frog	0.89	0.81	0.85	1000
horse	0.85	0.85	0.85	1000
ship	0.92	0.87	0.90	1000
truck	0.89	0.85	0.87	1000
accuracy			0.80	10000
macro avg	0.81	0.80	0.80	10000
weighted avg	0.81	0.80	0.80	10000

Performance of Model Relative to Others

Primary Findings

Gaussian Naive Bayes: This model performed the second best with accuracy around 80%. Its simplicity and the assumption of feature independence made it computationally efficient and robust, especially with PCA reducing dimensionality.

Decision Tree: This model performed the worst with accuracies ranging from 57% to 54%. Poor performance due to overfitting at higher depths and sensitivity to noise in PCA-reduced features.

Multi-Layer Perceptron: This model performed second worst with accuracy around 69%. Reasonable performance because of possible overfitting and reliance on PCA-reduced features that could not fully capture data complexity

Convolutional Neural Network: This model performed the best but not by much with accuracies ranging from 80% to 83%. Due to its architecture, it lends itself well to feature extraction, and avoids overfitting even with the limited data size.

For the CIFAR-10 dataset and the sizes we were working with, we can deduce that GNB is the best model to use as the simplicity is one of the key strengths of Naive Bayes models, making them fast and computationally efficient. As opposed to CNN which are comparably very heavy models both computationally and in complexity. The accuracies are extremely similar between

both models. If the absolute best generalization is needed, use CNN models. If a balance of computational efficiency and generalization is needed, then use GNB.

The frequently confused classes were Airplane vs. Bird and Cat vs. Dog because of issues in identifying visually similar objects, especially when features overlap. The models varied in their ability to identify these classes. Simpler models like Naive Bayes struggled due to their assumptions. Decision Trees overfit, leading to limited generalization. MLPs and CNNs performed better by capturing complex relationships, with CNNs outperforming due to their ability to extract spatial features. The well recognized classes were Automobile, Truck and Frog because of distinct visual characteristics, which were easier for all models to classify.

AI Prompts

For Naive Bayes Models:

- "How can I evaluate my manually implemented Gaussian Naive Bayes model and compare it with a Scikit-Learn implementation?"
- "Fix the issue with Naive Bayes BP3 evaluation to include model file names for loading."

For Decision Trees:

- "What is wrong with this Decision Tree code? It's showing an error when predicting with the Scikit-Learn model in BP3."
- "Add logic for evaluating manually implemented decision trees with varying depths and create confusion matrices."

For Multi-Layer Perceptron (MLP):

- "Why is the accuracy of my MLP BP1 model lower during evaluation compared to training? It was showing 72% earlier."
- "How can I evaluate all saved MLP models from BP1, BP2, and BP3 efficiently using PyTorch and Scikit-Learn metrics?"

For Convolutional Neural Networks (CNNs):

- "Does my CNN code overwrite the earlier BP1, BP2, or BP3 models after I renamed the class architecture?"
- "Fix this CNN evaluation logic to correctly load and test models saved in the Google Drive runtime environment."

General Prompts for Errors and Debugging:

- "Fix this FileNotFoundError when trying to load PCA-reduced features."
- "Why is this CNN code suddenly taking much longer to run on the T4 GPU?"

References

[1] PyTorch Documentation, "PyTorch: Tensors and Dynamic neural networks in Python with strong GPU acceleration," [Online]. Available: https://pytorch.org/docs/stable/index.html. [Accessed: Nov. 12, 2024].

[2] scikit-learn Documentation, "GaussianNB — Naive Bayes classifiers," [Online]. Available: https://scikit-learn.org/dev/modules/generated/sklearn.naive_bayes.GaussianNB.html. [Accessed: Nov. 12, 2024].

[3] scikit-learn Documentation, "Decision Trees — scikit-learn 1.5.2," [Online]. Available: https://scikit-learn.org/dev/modules/generated/sklearn.tree.DecisionTreeClassifier.html. [Accessed: Nov. 14, 2024].