HOMEWORK 3B
ECE/CS 8690 2302 Computer Vision

Image Classification

Xuanbo Miao

14422044

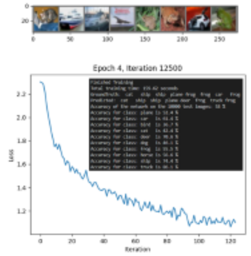
xmiao@mail.missouri.edu

Abstract

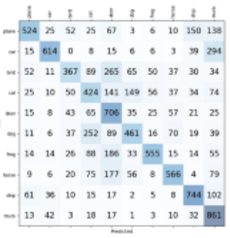
This assignment addresses the enhancement of a convolutional neural network (CNN) for multi-class image classification. Challenges like poor classification performance on specific classes are analyzed using a confusion matrix. Factors affecting performance include insufficient training, similarity between classes, suboptimal network architecture, inadequate data preprocessing, and imbalanced dataset. Various techniques are suggested to improve performance, such as experimenting with network architectures, increasing training data size, applying data augmentation, addressing class imbalance, and using transfer learning. The assignment underscores the importance of understanding and optimizing factors that affect CNN performance in computer vision tasks.

REPORT: Net 1

Net1. Acc Tab	
Plane	52.4 %
Car	61.4 %
Bird	36.7 %
Cat	42.4 %
Deer	70.6 %
Dog	46.1 %
Frog	55.5 %
Horse	56.6 %
Ship	74.4 %
Truck	86.1 %
Ave Acc	58.0 %
Run Tim	157.62 s



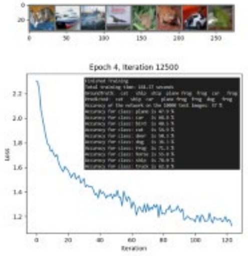
Confusion Matrices



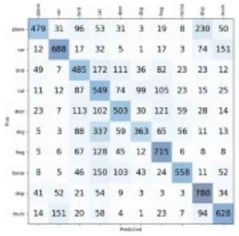
plane	524	25	52	25	67	3	6	10	150	138
car	15	614	0	8	15	6	6	3	39	294
bird	52	11	367	89	265	65	50	37	30	34
cat	25	10	50	424	141	149	56	37	34	74
deer	15	8	43	65	706	35	25	57	21	25
dog	11	6	37	252	89	461	16	70	19	39
frog	14	14	26	88	186	33	555	15	14	55
horse	9	6	20	75	177	56	8	566	4	79
ship	61	36	10	15	17	2	5	8	744	102
truck	13	42	3	18	17	1	3	10	32	861

REPORT: Net 2

Net1. Acc Tab	
Plane	47.9 %
Car	68.8 %
Bird	48.5 %
Cat	54.9 %
Deer	50.3 %
Dog	36.3 %
Frog	71.5 %
Horse	58.8 %
Ship	78.0 %
Truck	62.8 %
Ave Acc	57.0 %
Run Tim	161.37 s



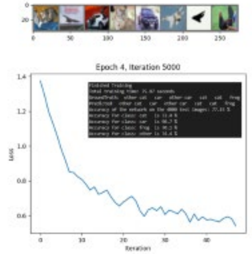
Confusion Matrices



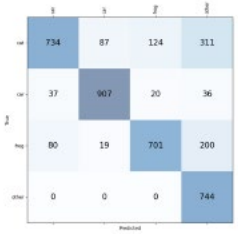
plane	479	31	96	53	31	3	19	8	230	50
car	12	688	17	32	5	1	17	3	74	151
bird	49	7	485	172	111	36	82	23	23	12
cat	11	12	87	549	74	99	105	23	15	25
deer	23	7	113	102	503	30	121	59	28	14
dog	5	3	88	337	59	363	65	56	11	13
frog	5	6	67	128	45	12	715	6	8	8
horse	8	5	46	150	103	43	24	558	11	52
ship	41	52	21	54	9	3	3	3	780	34
truck	14	151	20	58	4	1	23	7	94	628

REPORT: Net 3

Net1. Acc Tab	
Car	90.7 %
Cat	73.4 %
Frog	70.1 %
Other	74.4 %
Ave Acc	77.15 %
Run Tim	75.97 s



Confusion Matrices



cat	734	87	124	311
car	37	907	20	36
frog	80	19	701	200
other	0	0	0	744

Visualization:

Comment on what factors may have caused the poor performance:

1. Insufficient training: The model may not have been trained for enough epochs or with a large enough dataset to learn the distinguishing features of the worst-performing class effectively. This could lead to the model struggling to differentiate between the class and others, especially if they share similar features.
2. Similarity between classes: If the worst-performing class and the class it is most confused with share similar visual features or have overlapping characteristics, the model might find it challenging to differentiate between them. For example, if the classes are "cat" and "dog," the model may struggle because both animals share similar shapes, sizes, and textures.
3. Network architecture: The architecture of the neural network may not be well-suited for the task at hand. A deeper or more complex network with additional layers, filters, or more advanced techniques (such as residual connections) might be needed to capture the relevant features and learn the difference between the classes more effectively.
4. Data preprocessing: The normalization and data augmentation techniques used during preprocessing may not have been sufficient or adequate for the task. Improper preprocessing might cause the model to focus on irrelevant features or miss essential information needed to differentiate between the classes.
5. Imbalanced dataset: If the dataset used for training is imbalanced, with fewer examples of the worst-performing class compared to other classes, the model might struggle to learn the features of that class. Addressing class imbalance through techniques like oversampling, undersampling, or using different loss functions (e.g., weighted cross-entropy loss) could potentially help improve the model's performance on the underrepresented class.

To improve the model's performance, consider experimenting with different network architectures, increasing the training data size, using different data augmentation techniques, addressing class imbalance, or using transfer learning with pre-trained models.

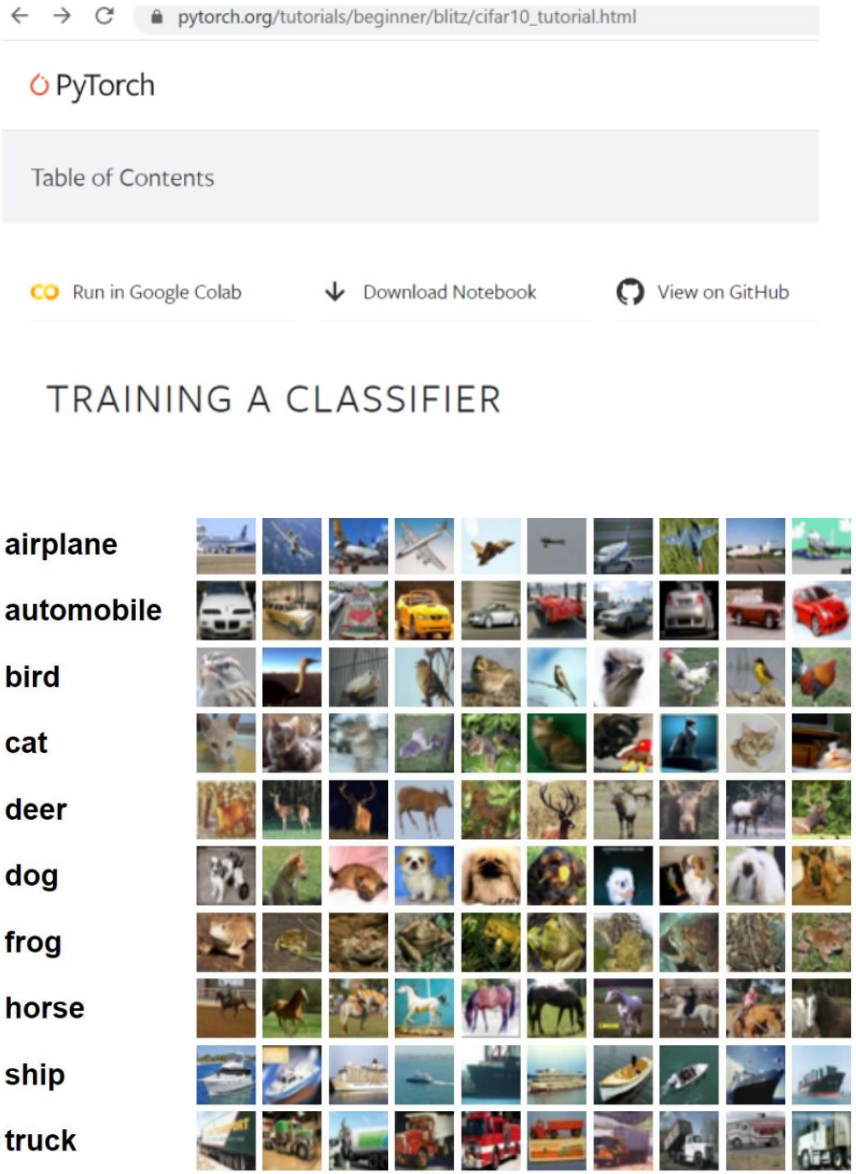
Introduction

Overview

- Create, train, and test a convolutional neural network for image classification
- Modify existing image classification code from PyTorch tutorial
- Evaluate and compare different network architectures
- Use CIFAR10 dataset for training and testing

Dataset: CIFAR10

- 60,000 images of 10 common classes
- Image size: 32 x 32 x 3
- 50,000 training images and 10,000 test images
- Built-in dataset in Torchvision



pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

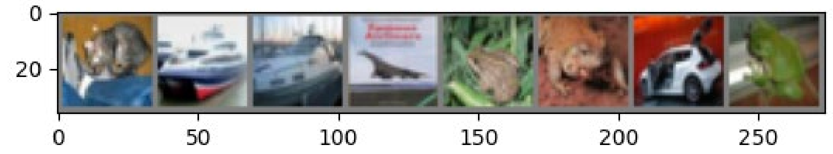
PyTorch

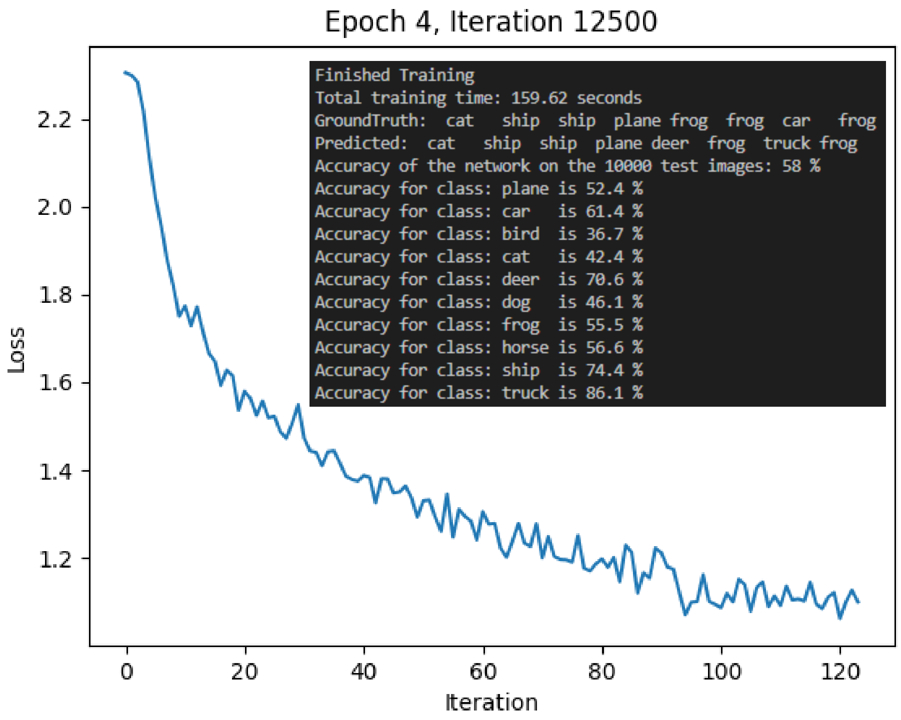
Table of Contents

TRAINING A CLASSIFIER

REPORT: Net 1

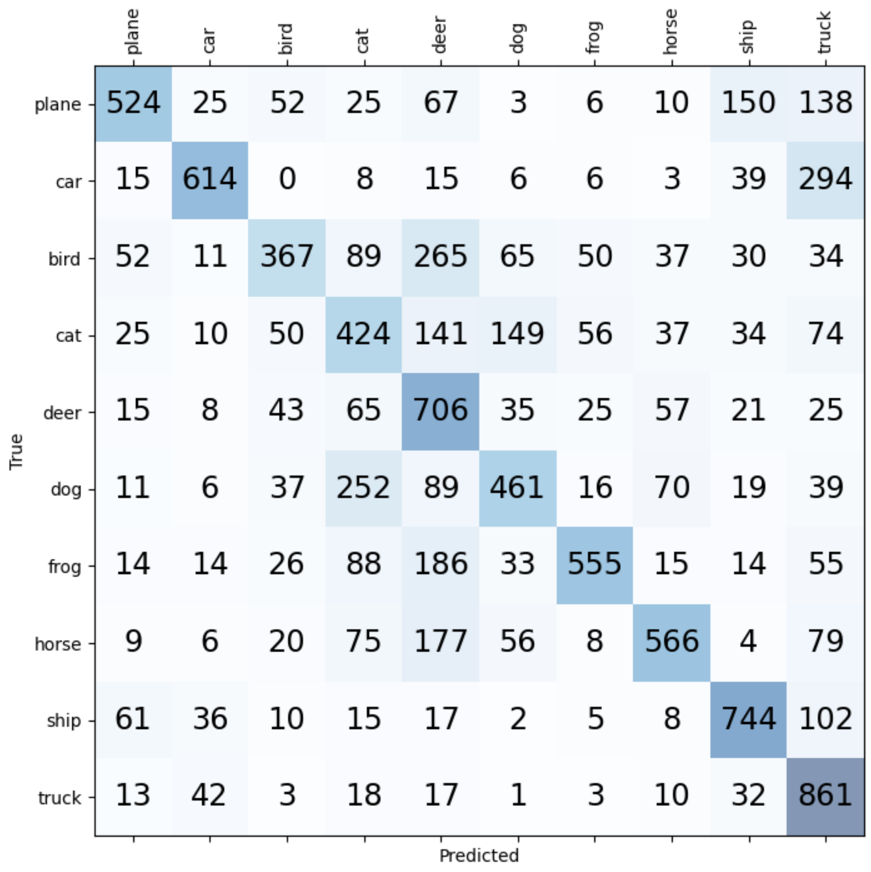
Net1. Acc Tab	
Plane	52.4 %
Car	61.4 %
Bird	36.7 %
Cat	42.4 %
Deer	70.6 %
Dog	46.1 %
Frog	55.5 %
Horse	56.6 %
Ship	74.4 %
Truck	86.1 %
Ave Acc	58.0 %
Run Tim	157.62 s





```
Finished Training
Total training time: 159.62 seconds
GroundTruth:  cat   ship  ship  plane frog  frog  car   frog
Predicted:  cat   ship  ship  plane deer  frog  truck frog
Accuracy of the network on the 10000 test images: 58 %
Accuracy for class: plane is 52.4 %
Accuracy for class: car   is 61.4 %
Accuracy for class: bird  is 36.7 %
Accuracy for class: cat   is 42.4 %
Accuracy for class: deer  is 70.6 %
Accuracy for class: dog   is 46.1 %
Accuracy for class: frog  is 55.5 %
Accuracy for class: horse is 56.6 %
Accuracy for class: ship  is 74.4 %
Accuracy for class: truck is 86.1 %
```

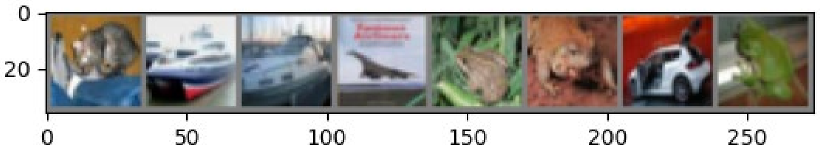
Confusion Matrices

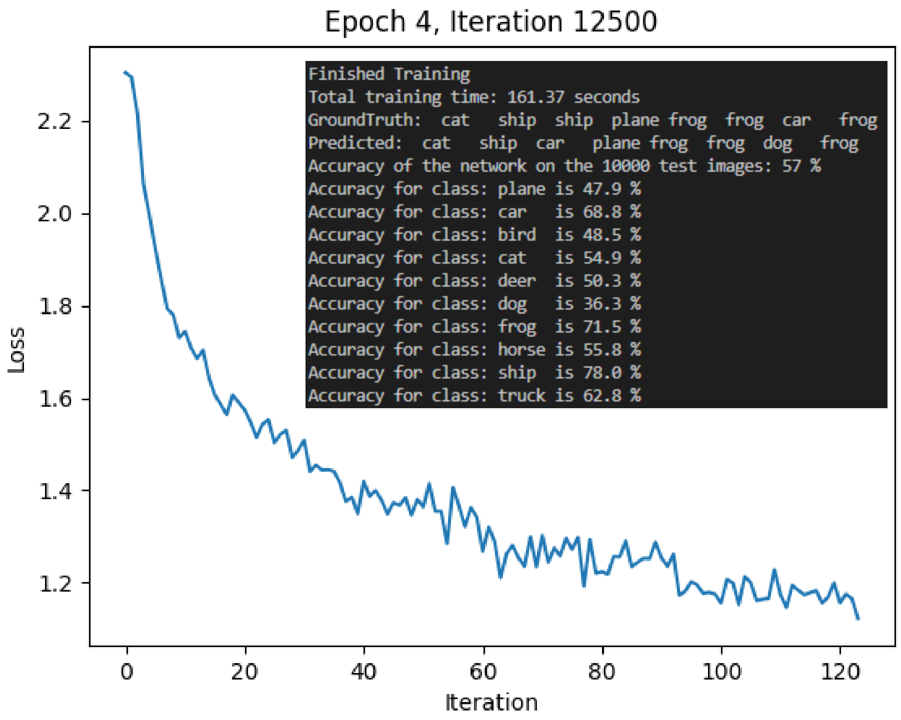


	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
plane	524	25	52	25	67	3	6	10	150	138
car	15	614	0	8	15	6	6	3	39	294
bird	52	11	367	89	265	65	50	37	30	34
cat	25	10	50	424	141	149	56	37	34	74
deer	15	8	43	65	706	35	25	57	21	25
dog	11	6	37	252	89	461	16	70	19	39
frog	14	14	26	88	186	33	555	15	14	55
horse	9	6	20	75	177	56	8	566	4	79
ship	61	36	10	15	17	2	5	8	744	102
truck	13	42	3	18	17	1	3	10	32	861

REPORT: Net 2

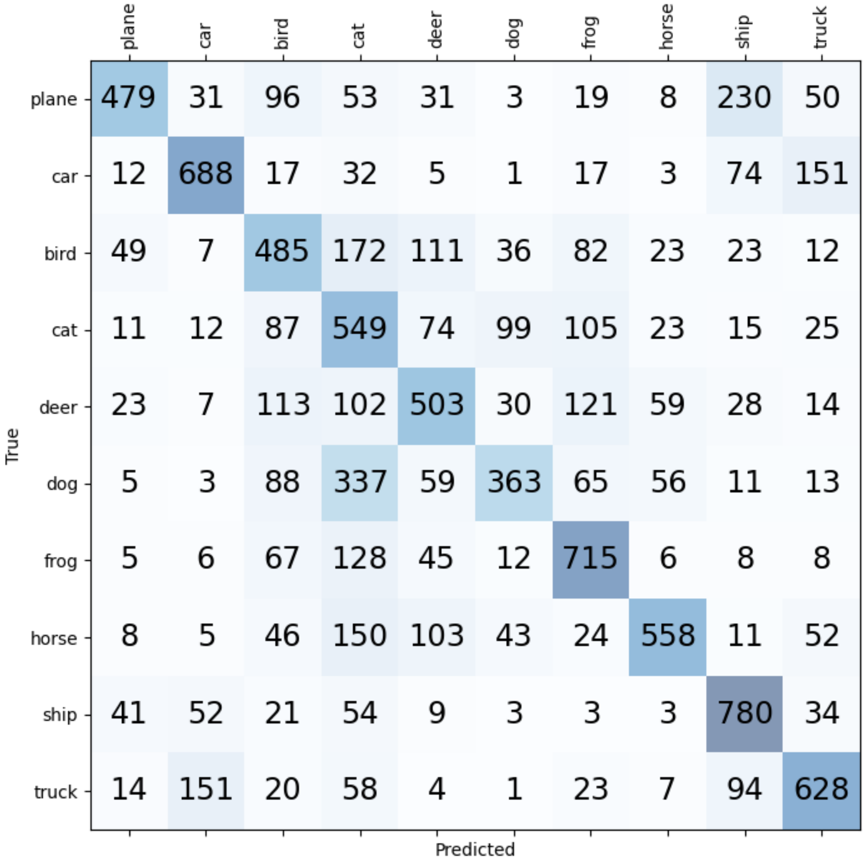
Net1. Acc Tab	
Plane	47.9 %
Car	68.8 %
Bird	48.5 %
Cat	54.9 %
Deer	50.3 %
Dog	36.3 %
Frog	71.5 %
Horse	58.8 %
Ship	78.0 %
Truck	62.8 %
Ave Acc	57.0 %
Run Tim	161.37 s





```
Finished Training
Total training time: 161.37 seconds
GroundTruth:  cat   ship  ship  plane frog  frog  car   frog
Predicted:  cat   ship  car   plane frog  frog  dog   frog
Accuracy of the network on the 10000 test images: 57 %
Accuracy for class: plane is 47.9 %
Accuracy for class: car   is 68.8 %
Accuracy for class: bird  is 48.5 %
Accuracy for class: cat   is 54.9 %
Accuracy for class: deer  is 50.3 %
Accuracy for class: dog   is 36.3 %
Accuracy for class: frog  is 71.5 %
Accuracy for class: horse is 55.8 %
Accuracy for class: ship  is 78.0 %
Accuracy for class: truck is 62.8 %
```

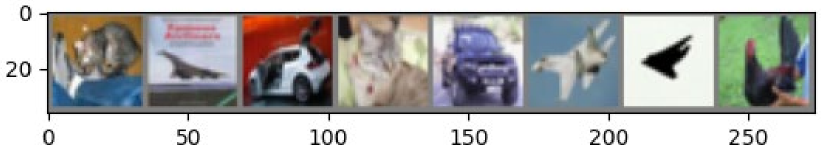
Confusion Matrices

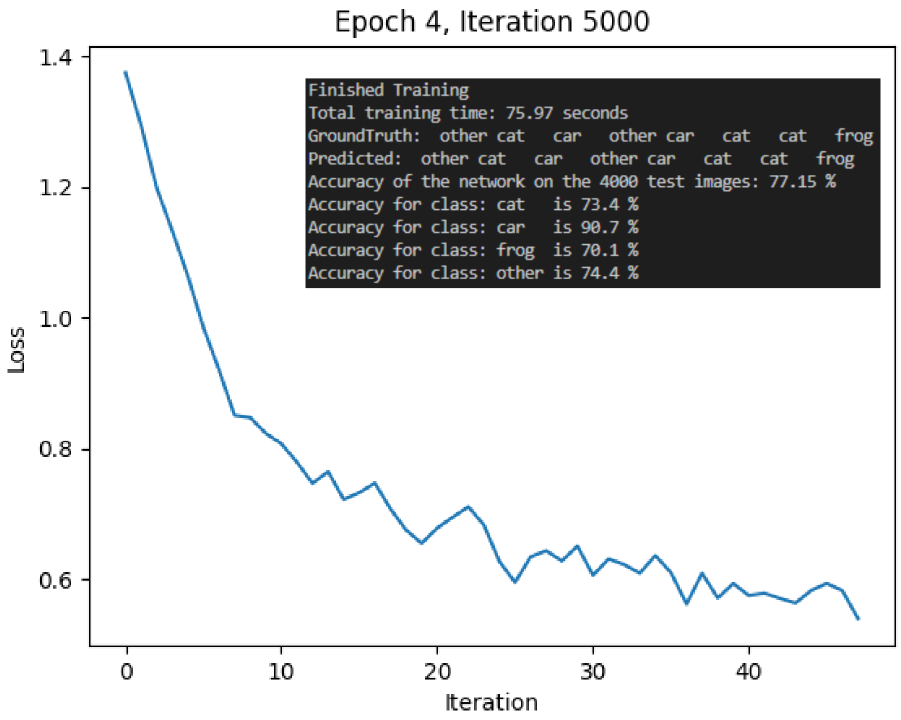


	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
plane	479	31	96	53	31	3	19	8	230	50
car	12	688	17	32	5	1	17	3	74	151
bird	49	7	485	172	111	36	82	23	23	12
cat	11	12	87	549	74	99	105	23	15	25
deer	23	7	113	102	503	30	121	59	28	14
dog	5	3	88	337	59	363	65	56	11	13
frog	5	6	67	128	45	12	715	6	8	8
horse	8	5	46	150	103	43	24	558	11	52
ship	41	52	21	54	9	3	3	3	780	34
truck	14	151	20	58	4	1	23	7	94	628

REPORT: Net 3

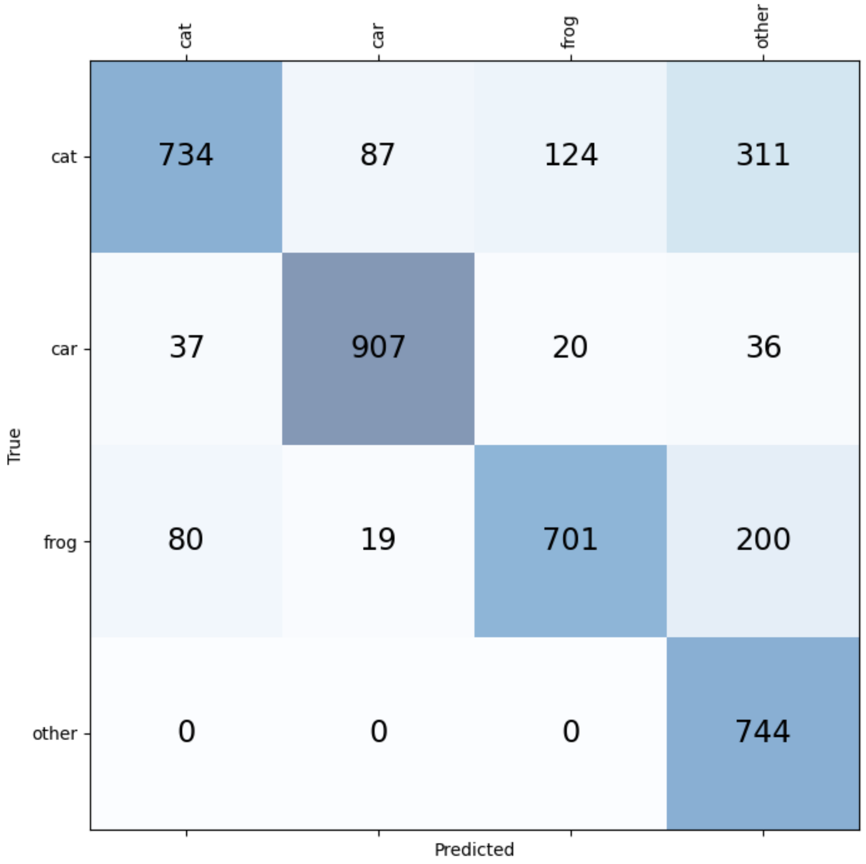
Net1. Acc Tab	
Car	90.7 %
Cat	73.4 %
Frog	70.1 %
Other	74.4 %
Ave Acc	77.15 %
Run Tim	75.97 s





```
Finished Training
Total training time: 75.97 seconds
GroundTruth:  other cat   car   other car   cat   cat   frog
Predicted:  other cat   car   other car   cat   cat   frog
Accuracy of the network on the 4000 test images: 77.15 %
Accuracy for class: cat   is 73.4 %
Accuracy for class: car   is 90.7 %
Accuracy for class: frog  is 70.1 %
Accuracy for class: other is 74.4 %
```

Confusion Matrices



	cat	car	frog	other
cat	734	87	124	311
car	37	907	20	36
frog	80	19	701	200
other	0	0	0	744

Q & A:

1. 10-class classification: for the class on which your model (Network 2) has the worst accuracy?

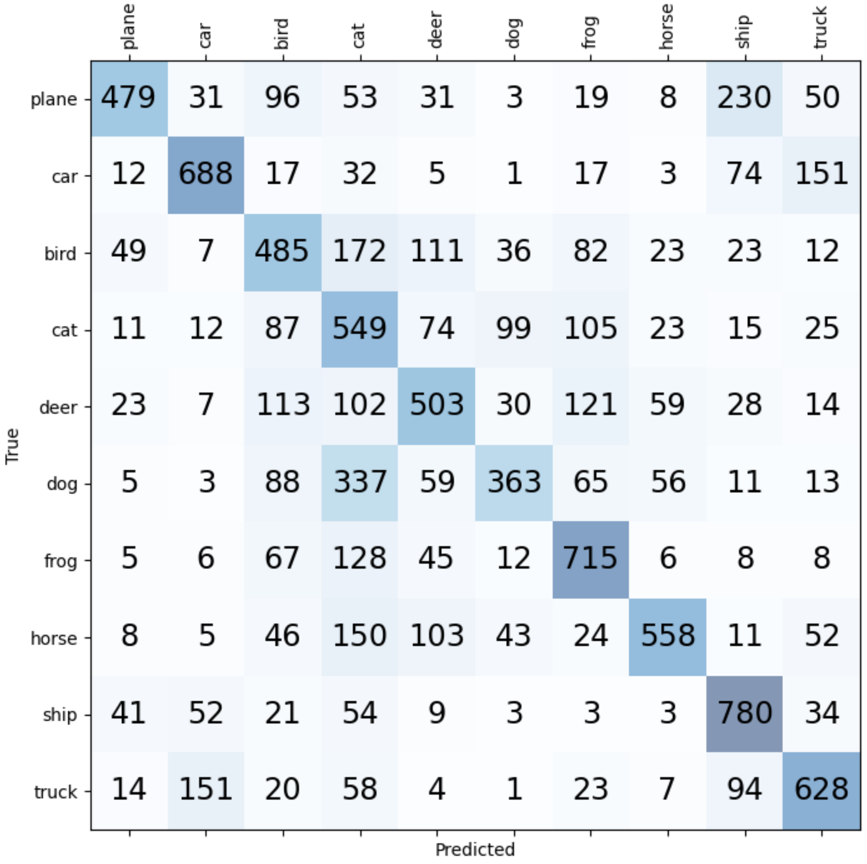
Dog

2. what is the other class it is most confused with?

Cat

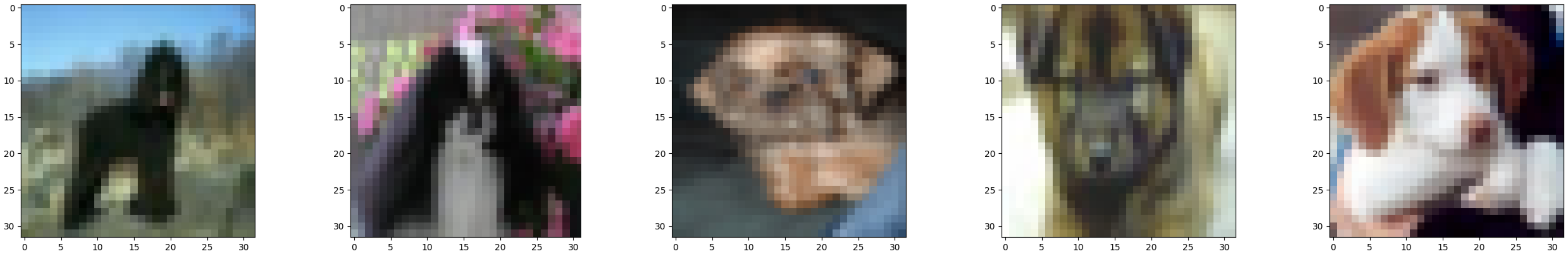
Net1. Acc Tab	
Plane	47.9 %
Car	68.8 %
Bird	48.5 %
Cat	54.9 %
Deer	50.3 %
Dog	36.3 %
Frog	71.5 %
Horse	58.8 %
Ship	78.0 %
Truck	62.8 %
Ave Acc	57.0 %
Run Tim	161.37 s

Confusion Matrices



	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
plane	479	31	96	53	31	3	19	8	230	50
car	12	688	17	32	5	1	17	3	74	151
bird	49	7	485	172	111	36	82	23	23	12
cat	11	12	87	549	74	99	105	23	15	25
deer	23	7	113	102	503	30	121	59	28	14
dog	5	3	88	337	59	363	65	56	11	13
frog	5	6	67	128	45	12	715	6	8	8
horse	8	5	46	150	103	43	24	558	11	52
ship	41	52	21	54	9	3	3	3	780	34
truck	14	151	20	58	4	1	23	7	94	628

Visualization:



Comment on what factors may have caused the poor performance:

1. Insufficient training: The model may not have been trained for enough epochs or with a large enough dataset to learn the distinguishing features of the worst-performing class effectively. This could lead to the model struggling to differentiate between the class and others, especially if they share similar features.
2. Similarity between classes: If the worst-performing class and the class it is most confused with share similar visual features or have overlapping characteristics, the model might find it challenging to differentiate between them. For example, if the classes are "cat" and "dog," the model may struggle because both animals share similar shapes, sizes, and textures.
3. Network architecture: The architecture of the neural network may not be well-suited for the task at hand. A deeper or more complex network with additional layers, filters, or more advanced techniques (such as residual connections) might be needed to capture the relevant features and learn the difference between the classes more effectively.
4. Data preprocessing: The normalization and data augmentation techniques used during preprocessing may not have been sufficient or adequate for the task. Improper preprocessing might cause the model to focus on irrelevant features or miss essential information needed to differentiate between the classes.
5. Imbalanced dataset: If the dataset used for training is imbalanced, with fewer examples of the worst-performing class compared to other classes, the model might struggle to learn the features of that class. Addressing class imbalance through techniques like oversampling, undersampling, or using different loss functions (e.g., weighted cross-entropy loss) could potentially help improve the model's performance on the underrepresented class.

To improve the model's performance, consider experimenting with different network architectures, increasing the training data size, using different data augmentation techniques, addressing class imbalance, or using transfer learning with pre-trained models.

Interpretation & Discussion

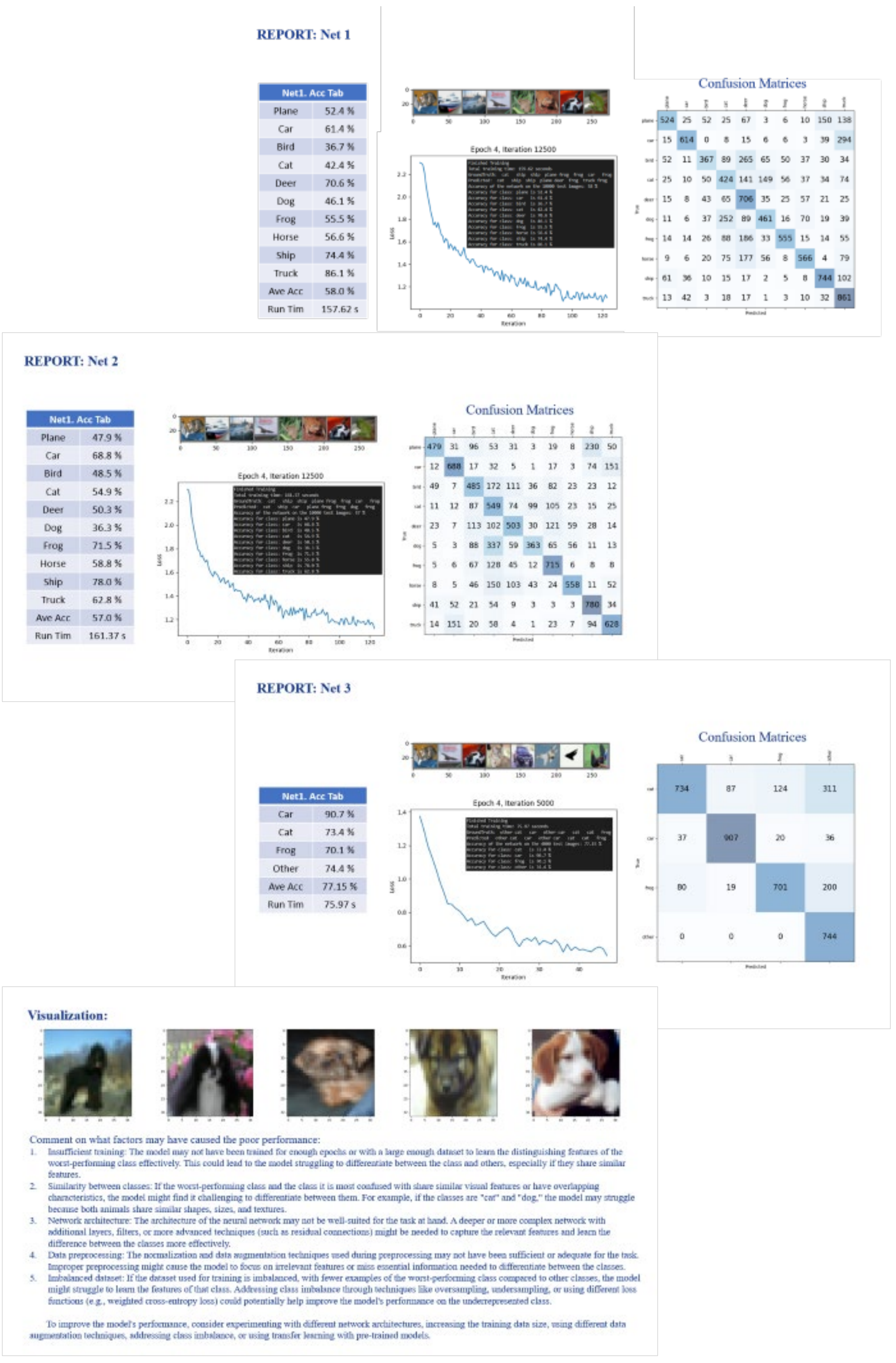
This assignment aimed to implement a computer vision model, likely a convolutional neural network (CNN), to perform image classification on a given dataset. The goal was to train the model to learn the features of the images and correctly classify them into their respective classes.

Objectives: 1. Understand the dataset: It is crucial to explore and analyze the dataset to get an idea of the distribution of classes, the variety of images, and any potential class imbalance. This step helps determine the appropriate preprocessing techniques, data augmentation strategies, and network architectures; 2. Preprocess the data: The dataset needs to be preprocessed, including resizing, normalization, and data augmentation, to improve the model's ability to learn the features and generalize better; 3. Implement the model: Design and implement a suitable CNN architecture for image classification. The model should be able to learn the distinguishing features of the different classes and classify images with reasonable accuracy; 4. Train and evaluate the model: Train the model using an appropriate loss function and optimization algorithm. Evaluate the model's performance using relevant metrics, such as accuracy, precision, recall, and F1-score. A confusion matrix can also be used to identify the classes that the model struggles with the most.

Challenges: 1. Choosing the right network architecture: Designing an optimal CNN architecture for the task can be challenging. It requires experimenting with different layer types, depth, and activation functions; 2. Dealing with class imbalance: If the dataset is imbalanced, the model might struggle to learn the features of underrepresented classes. Techniques like oversampling, undersampling, or using different loss functions can help address this issue; 3. Handling similar classes: If certain classes share similar visual features, the model might have difficulties differentiating between them.

Potential Improvements: 2. Experiment with different network architectures: Try deeper or more complex networks, or use pre-trained models via transfer learning to improve the model's performance; 2.Use advanced techniques: Implement advanced techniques like residual connections, batch normalization, or dropout to enhance the model's ability to generalize; 3. Address class imbalance: Experiment with techniques like oversampling, undersampling, or using different loss functions to improve the model's performance on underrepresented classes; 4. Optimize hyperparameters: Perform hyperparameter tuning, such as adjusting the learning rate, batch size, and number of epochs, to find the best combination that maximizes the model's performance.

In conclusion, this assignment provided valuable experience in implementing and evaluating a computer vision model for image classification. By addressing the challenges and implementing potential improvements, the model's performance can be further enhanced, paving the way for more accurate and robust image classification solutions.



REPORT: Net 1

Net1. Acc Tab	
Plane	52.4 %
Car	61.4 %
Bird	36.7 %
Cat	42.4 %
Deer	70.6 %
Dog	46.1 %
Frog	55.5 %
Horse	56.6 %
Ship	74.4 %
Truck	86.1 %
Ave Acc	58.0 %
Run Tim	157.62 s

Confusion Matrices

plane	524	25	52	25	67	3	6	10	150	138
car	15	614	0	8	15	6	6	3	39	294
bird	52	11	367	89	265	65	50	37	30	34
cat	25	10	50	424	141	149	56	37	34	74
deer	15	8	43	65	706	35	25	57	21	25
dog	11	6	37	252	89	461	16	70	19	39
frog	14	14	26	88	186	33	555	15	14	55
horse	9	6	20	75	177	56	8	566	4	79
ship	61	36	10	15	17	2	5	8	744	102
truck	13	42	3	18	17	1	3	10	32	861

REPORT: Net 2

Net1. Acc Tab	
Plane	47.9 %
Car	68.8 %
Bird	48.5 %
Cat	54.9 %
Deer	50.3 %
Dog	36.3 %
Frog	71.5 %
Horse	58.8 %
Ship	78.0 %
Truck	62.8 %
Ave Acc	57.0 %
Run Tim	161.37 s

Confusion Matrices

plane	479	31	96	53	31	3	19	8	230	50
car	12	688	17	32	5	1	17	3	74	151
bird	49	7	485	172	111	36	82	23	23	12
cat	11	12	87	549	74	99	105	23	15	25
deer	23	7	113	102	503	30	121	59	28	14
dog	5	3	88	337	59	363	65	56	11	13
frog	5	6	67	128	45	12	715	6	8	8
horse	8	5	46	150	103	43	24	558	11	52
ship	41	52	21	54	9	3	3	3	780	34
truck	14	151	20	58	4	1	23	7	94	628

REPORT: Net 3

Net1. Acc Tab	
Car	90.7 %
Cat	73.4 %
Frog	70.1 %
Other	74.4 %
Ave Acc	77.15 %
Run Tim	75.97 s

Confusion Matrices

cat	734	87	124	311
car	37	907	20	36
frog	80	19	701	200
other	0	0	0	744

Visualization:

Comment on what factors may have caused the poor performance:

1. Insufficient training: The model may not have been trained for enough epochs or with a large enough dataset to learn the distinguishing features of the worst-performing class effectively. This could lead to the model struggling to differentiate between the class and others, especially if they share similar features.
2. Similarity between classes: If the worst-performing class and the class it is most confused with share similar visual features or have overlapping characteristics, the model might find it challenging to differentiate between them. For example, if the classes are "cat" and "dog," the model may struggle because both animals share similar shapes, sizes, and textures.
3. Network architecture: The architecture of the neural network may not be well-suited for the task at hand. A deeper or more complex network with additional layers, filters, or more advanced techniques (such as residual connections) might be needed to capture the relevant features and learn the difference between the classes more effectively.
4. Data preprocessing: The normalization and data augmentation techniques used during preprocessing may not have been sufficient or adequate for the task. Improper preprocessing might cause the model to focus on irrelevant features or miss essential information needed to differentiate between the classes.
5. Imbalanced dataset: If the dataset used for training is imbalanced, with fewer examples of the worst-performing class compared to other classes, the model might struggle to learn the features of that class. Addressing class imbalance through techniques like oversampling, undersampling, or using different loss functions (e.g., weighted cross-entropy loss) could potentially help improve the model's performance on the underrepresented class.

To improve the model's performance, consider experimenting with different network architectures, increasing the training data size, using different data augmentation techniques, addressing class imbalance, or using transfer learning with pre-trained models.