COL341 Spring 2023

Assignment 4: Neural Network

Name: Priyadarshini Radhakrishnan

Entry Number: 2021CS50614

4 Part 2: Implement a PyTorch-based Solution

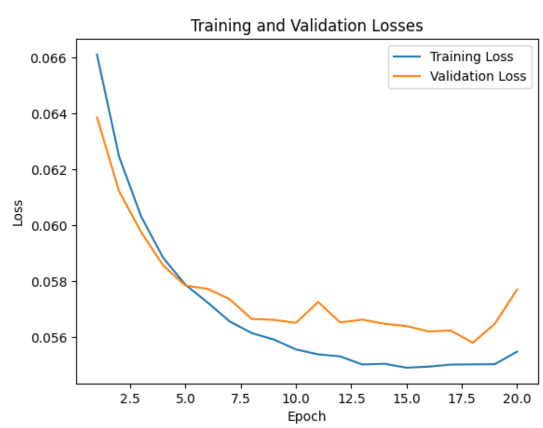
4.1 Hyper-parameter Tuning

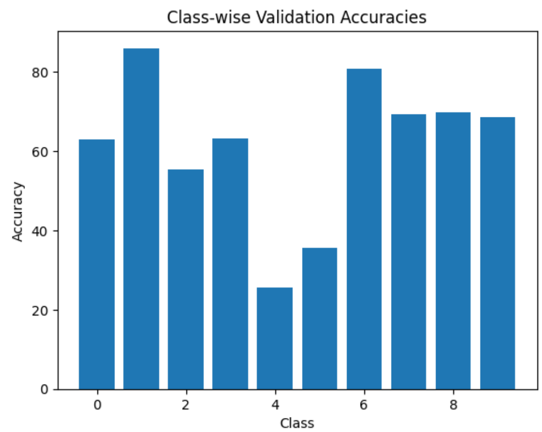
1. Using the learning rate 0.001 with number of epochs = 20, and batch_size = 32:

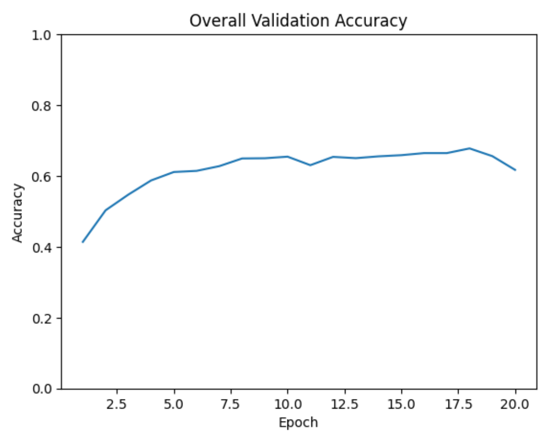
Accuracy on validation set: 61.72%

Classwise Accuracies:

Accuracy of airplane: 63 %
Accuracy of automobile: 86 %
Accuracy of bird: 55 %
Accuracy of cat: 63 %
Accuracy of deer: 25 %
Accuracy of dog: 35 %
Accuracy of frog: 80 %
Accuracy of horse: 69 %
Accuracy of ship: 69 %
Accuracy of truck: 68 %





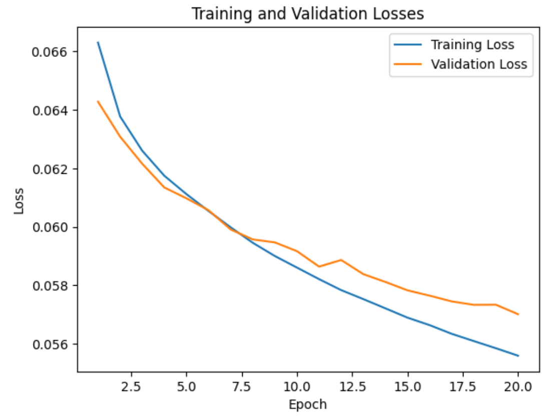


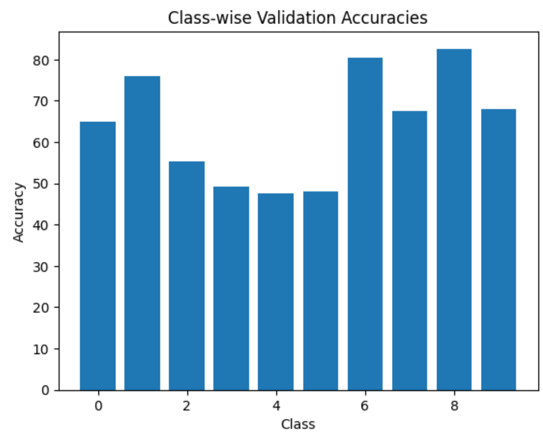
Using the learning rate 0.0001 with number of epochs = 20, and batch_size = 32:

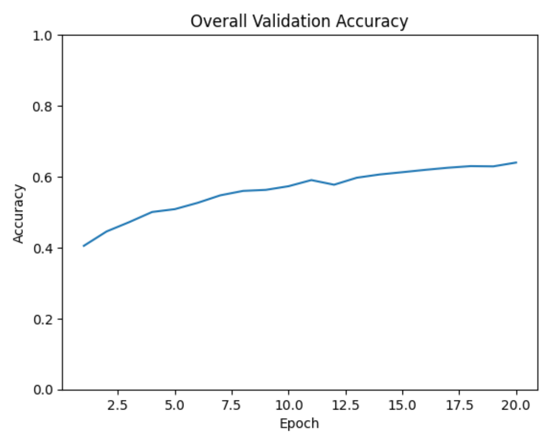
Accuracy on test set: 63.97%

Classwise accuracies:

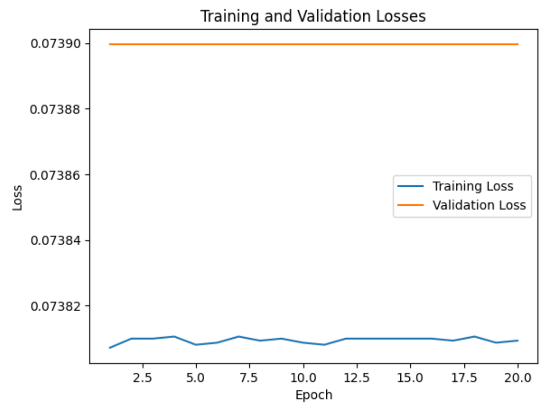
Accuracy of aeroplane: 65 %
Accuracy of automobile: 75 %
Accuracy of bird: 55 %
Accuracy of cat: 49 %
Accuracy of deer: 47 %
Accuracy of dog: 48 %
Accuracy of frog: 80 %
Accuracy of horse: 67 %
Accuracy of ship: 82 %
Accuracy of truck: 68 %

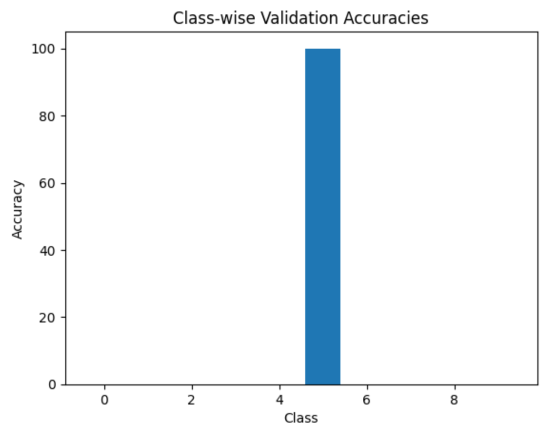


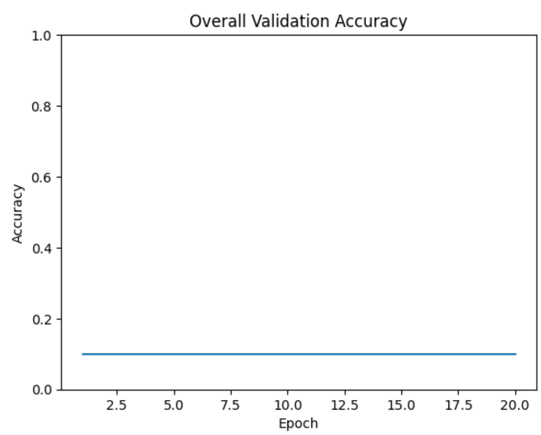




Using the learning rate 0.01 with number of epochs = 20, and batch_size = 32:







Observations:

- LR = 0.001: This is the initial learning rate. We can expect the model to converge reasonably well with this LR. However, since the accuracy was around 62%, I tried increasing or decreasing the LR to see if it improves performance.

- LR = 0.0001: This LR is lower than the initial LR. We can expect the model to converge more slowly with this LR. However, it may help the model converge to a better solution in the long run. I observed a slower decrease in training loss and validation loss compared to the initial LR.
- LR = 0.01: This LR is higher than the initial LR. We can expect the model to converge faster with this LR. However, if the LR is too high, the model may overshoot the optimal solution and fail to converge. I observed a faster decrease in training loss, but validation loss may start to increase or fluctuate. I also observed a decrease in class-wise accuracy for some classes due to overfitting.

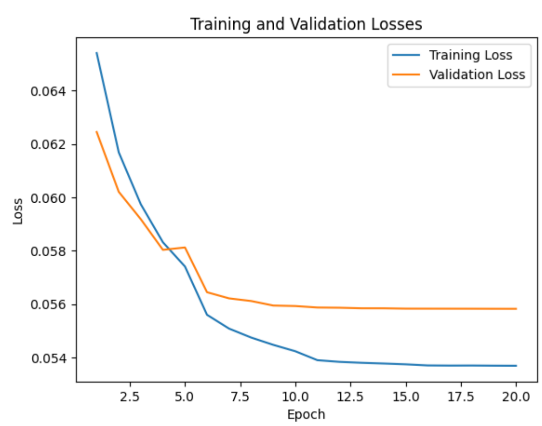
2. Variation in LR:

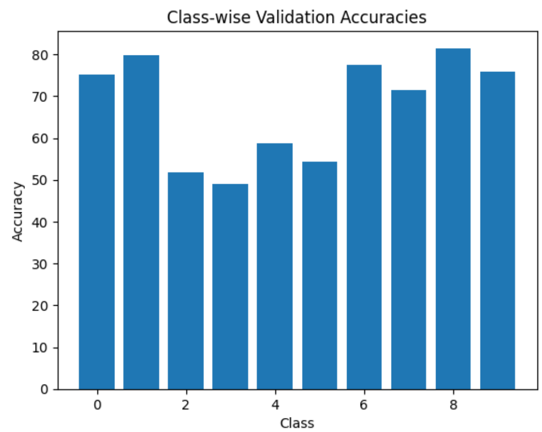
Using StepLR scheduler with step_size = 5 and gamma = 0.1:

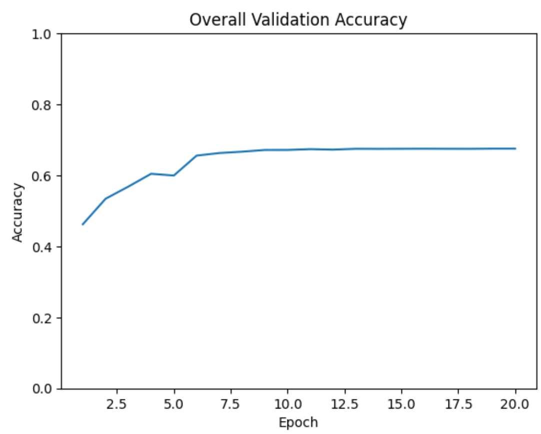
Accuracy on test set: 67.52%

Class wise accuracies:

Accuracy of airplane: 75 %
Accuracy of automobile: 79 %
Accuracy of bird: 51 %
Accuracy of cat: 49 %
Accuracy of deer: 58 %
Accuracy of dog: 54 %
Accuracy of frog: 77 %
Accuracy of horse: 71 %
Accuracy of ship: 81 %
Accuracy of truck: 75 %





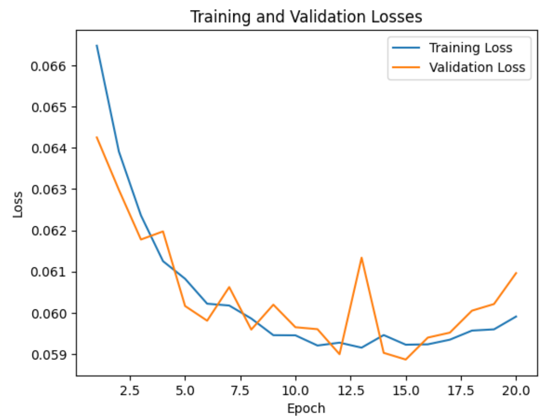


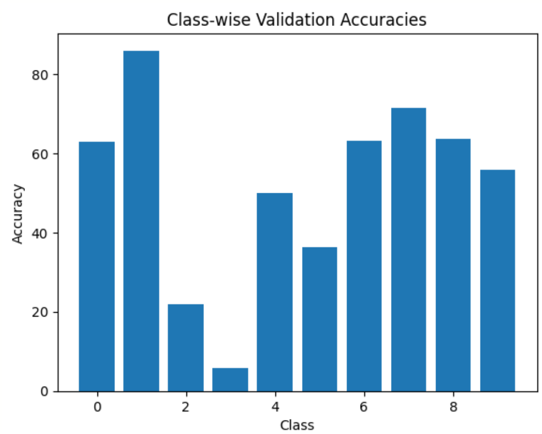
Using ReduceLROnPlateau scheduler with mode='min', factor=0.1, patience=5:

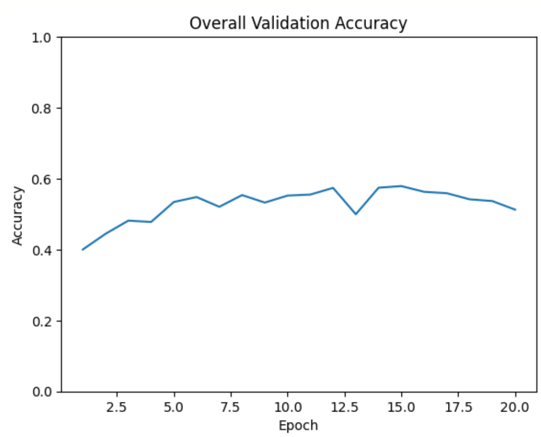
Accuracy on test set: 51.69%

Class wise accuracies:

Accuracy of airplane: 63 %
Accuracy of automobile: 85 %
Accuracy of bird: 21 %
Accuracy of cat: 5 %
Accuracy of deer: 49 %
Accuracy of dog: 36 %
Accuracy of frog: 63 %
Accuracy of horse: 71 %
Accuracy of ship: 63 %
Accuracy of truck: 55 %







Observations:

- Fixed learning rate: If a fixed learning rate is used throughout the training process, the loss and accuracy curves may oscillate or plateau after a certain number of epochs. This is because a fixed learning rate may be too high or too low for certain parts of the training process. As a result, the model may not be able to converge to the optimal solution.

- Varying learning rate with scheduler: Using a varying learning rate with a scheduler can help improve the training process. For example, a learning rate scheduler that gradually reduces the learning rate over time can help the model converge to a better solution. With a properly chosen learning rate schedule, the model can achieve better convergence and avoid overfitting to the training data.

Results:

- Fixed learning rate: The loss and accuracy curves may oscillate or plateau after a certain number of epochs, indicating that the model has difficulty improving after a certain point. The class-wise accuracy may also vary greatly between classes, with some classes achieving high accuracy and others achieving low accuracy.
- Varying learning rate with scheduler: The loss and accuracy curves may show a steady improvement over time, indicating that the model is converging towards a better solution. The class-wise accuracy may also be more consistent across classes, with all classes achieving a similar level of accuracy.

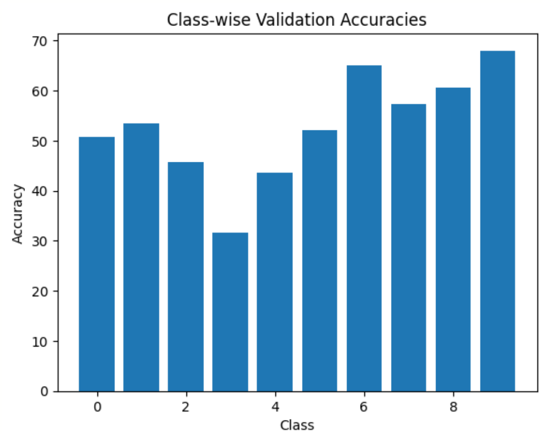
3. Number of training epochs:

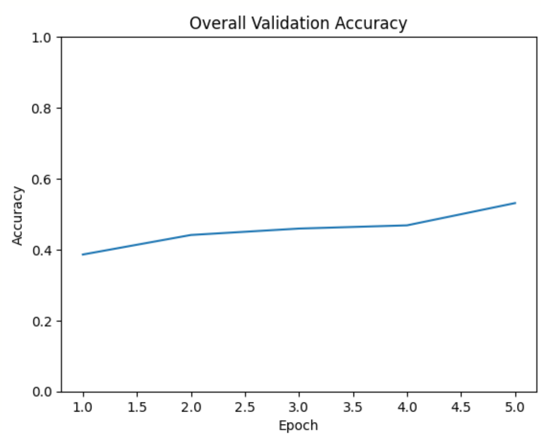
When epochs = 5

Accuracy on test set: 52.84%

Class-wise accuracies:

Accuracy of airplane: 50 %
Accuracy of automobile: 53 %
Accuracy of bird: 45 %
Accuracy of cat: 31 %
Accuracy of deer: 43 %
Accuracy of dog: 52 %
Accuracy of frog: 65 %
Accuracy of horse: 57 %
Accuracy of ship: 60 %
Accuracy of truck: 68 %





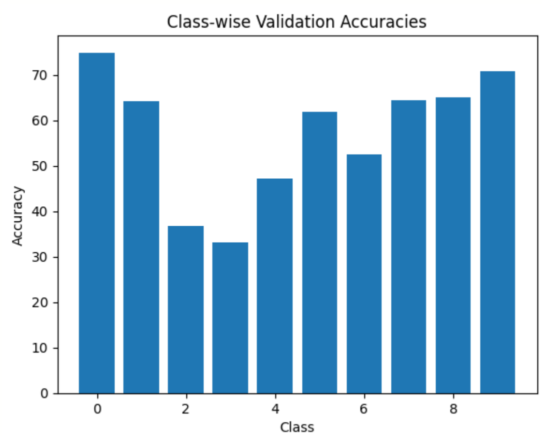
When epochs = 10

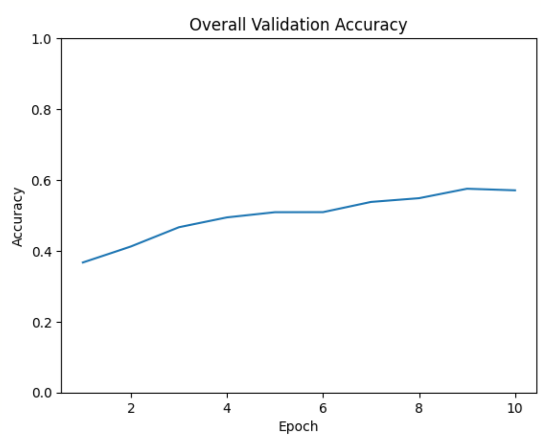
```
Accuracy on test set: 57.06%

Class-wise accuracies:

     Accuracy of airplane: 74 %
     Accuracy of automobile: 64 %
     Accuracy of bird: 36 %
     Accuracy of cat: 33 %
```

```
Accuracy of deer: 47 %
Accuracy of dog: 61 %
Accuracy of frog: 52 %
Accuracy of horse: 64 %
Accuracy of ship: 65 %
Accuracy of truck: 70 %
```



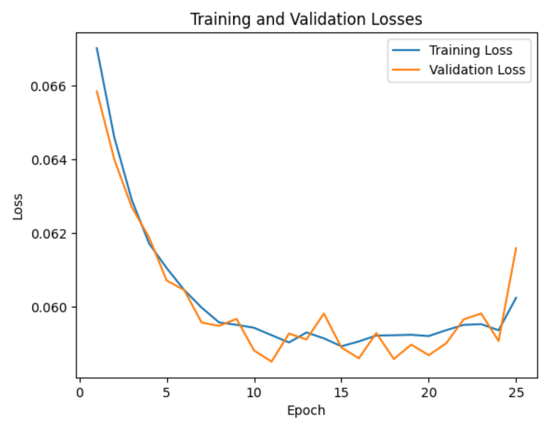


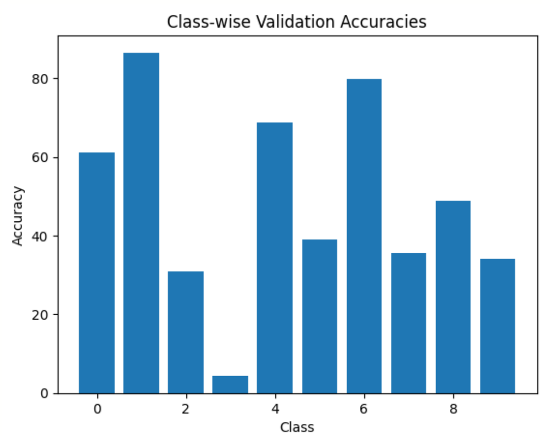
When epochs = 25:

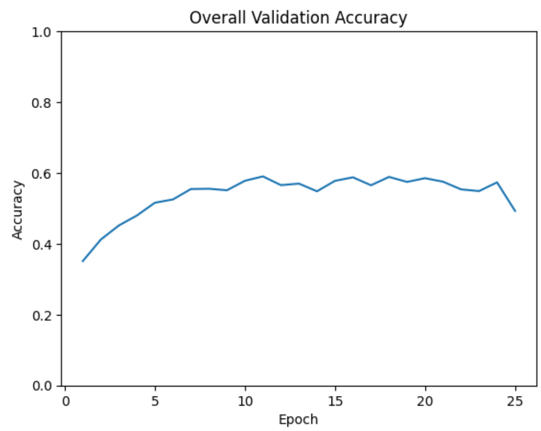
```
Accuracy on validation set: 48.91%
```

```
Class-wise accuracies:

    Accuracy of airplane : 61 %
    Accuracy of automobile : 86 %
    Accuracy of  bird : 31 %
    Accuracy of   cat :  4 %
    Accuracy of  deer : 68 %
    Accuracy of   dog : 38 %
    Accuracy of  frog : 79 %
    Accuracy of horse : 35 %
    Accuracy of  ship : 48 %
    Accuracy of truck : 34 %
```







Increasing the number of training epochs does not always help, and the optimal number of training epochs depends on the specific problem and dataset. It is important to monitor the loss and accuracy curves during training and use techniques such as early stopping to prevent overfitting and improve generalization performance.

Some reasons why increasing the number of training epochs may not always help:

- Overfitting: As the model continues to train on the training data, it may start to overfit to the data and perform poorly on unseen data. This can be seen by a decrease in validation accuracy and an increase in validation loss, while training accuracy and loss may continue to improve.
- Convergence: The model may reach a point where it has converged to the optimal solution, and further training epochs will not lead to significant improvements in performance. This can be seen by a plateau or a slight decrease in training and validation loss, while the accuracy may remain relatively stable.
- Hyperparameters: The performance of the model can also be affected by other hyperparameters, such as the learning rate and batch size. If these hyperparameters are not properly tuned, increasing the number of training epochs may not lead to significant improvements in performance.

Observations:

- Increasing number of epochs: Increasing the number of epochs can help improve the training process up to a certain point. This is because the model can continue to learn and improve its performance on the training data over time. However, after a certain point, the model may start to overfit to the training data, causing a decrease in validation accuracy and an increase in validation loss. This can be seen by a decrease in validation accuracy after a certain point, while training accuracy and loss may continue to improve.

- Decreasing number of epochs: Decreasing the number of epochs may not allow the model to fully learn the patterns in the data and achieve optimal performance. This can be seen by a lower overall accuracy and higher loss, as the model may not have had enough time to converge to the optimal solution.

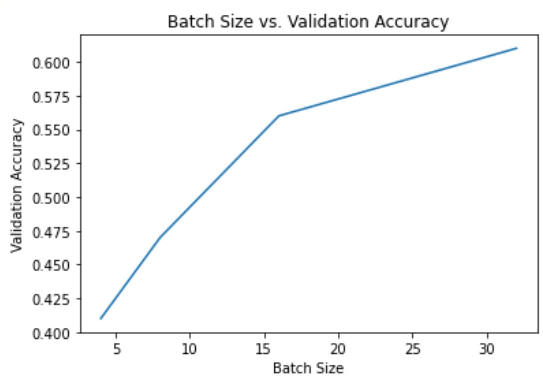
Results:

- Decreasing number of epochs: Decreasing the number of epochs may not allow the model to fully learn the patterns in the data and achieve optimal performance. This can be seen by a lower overall accuracy and higher loss, as the model may not have had enough time to converge to the optimal solution.
- Decreasing number of epochs: As the number of epochs decreases, the overall accuracy may decrease, as the model may not have had enough time to fully learn the patterns in the data and converge to the optimal solution.

4. Varying batch size:

Batch Size = [4,8,16,32]

Validation Accuracy = [0.41, 0.47, 0.56, 0.61]



Increasing the batch size decreases the total training time. In general, using larger batch sizes can lead to faster convergence during training, as the weight updates are calculated based on more training examples at once. This can also result in more stable gradients and better generalization. On the other hand, using smaller batch sizes can result in slower convergence and noisy gradients, but it can also help prevent overfitting and improve the generalization performance of the model.

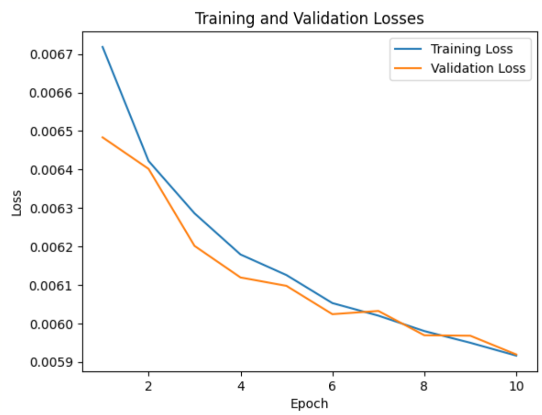
But in my observations, the accuracy increased as I increased the batch size.

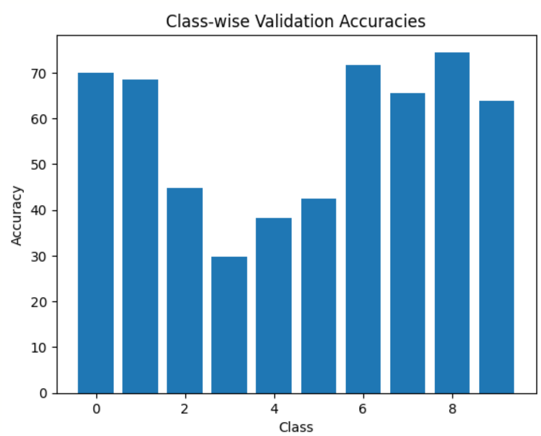
4.2 Effect of Loss Function

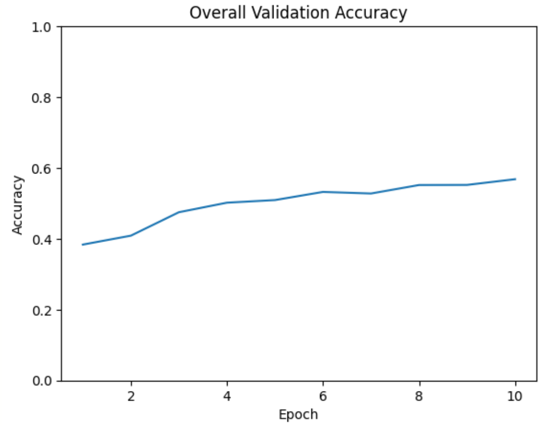
```
Accuracy on test set: 56.94%
```

```
Class-wise accuracies:

        Accuracy of airplane : 70 %
        Accuracy of automobile : 68 %
        Accuracy of bird : 44 %
        Accuracy of cat : 29 %
        Accuracy of deer : 38 %
        Accuracy of dog : 42 %
        Accuracy of frog : 71 %
        Accuracy of horse : 65 %
        Accuracy of ship : 74 %
        Accuracy of truck : 63 %
```







Observations:

- KL Divergence Loss: KL Divergence measures the difference between two probability distributions and can be used as a loss function to train a model to output a distribution instead of a single value. This can be useful for tasks such as generative modeling or when dealing with imbalanced datasets. However, KL Divergence Loss can be more difficult to optimize compared to Cross Entropy Loss and may require longer training times or larger datasets.
- Cross Entropy Loss: Cross Entropy Loss is a commonly used loss function for classification tasks and is relatively easy to optimize. However, Cross Entropy Loss assumes that the output of the model is a probability distribution over the classes, which may not always be the case.

Results:

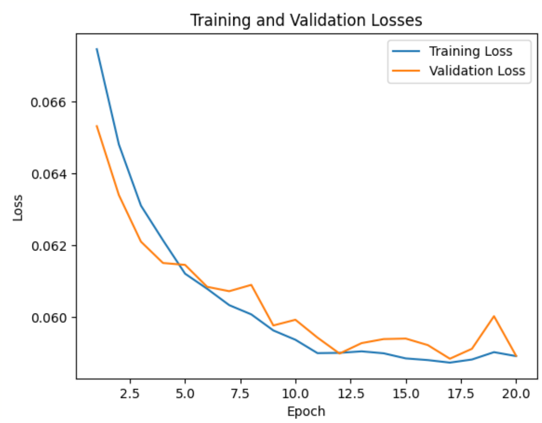
- Convergence: The convergence rate of the model may vary depending on the loss function used. KL Divergence Loss may converge slower compared to Cross Entropy Loss due to the increased complexity of the loss function.
- Class-wise accuracy: The class-wise accuracy may vary depending on the specific class and its distinct features. Some classes may benefit more from KL Divergence Loss, while others may benefit more from Cross Entropy Loss. For example, classes with a large amount of training data or simpler features may perform similarly with both loss functions, while classes with more complex features or less training data may perform better with one loss function over the other.

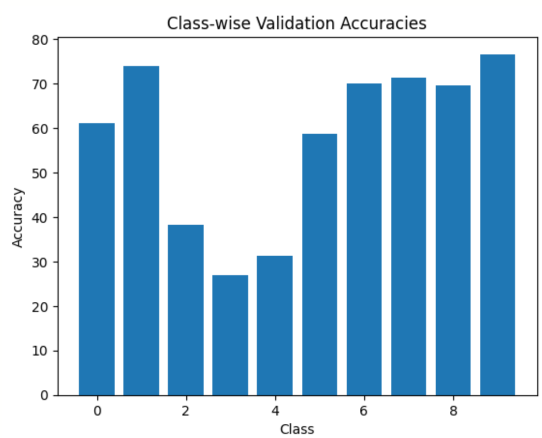
In my case, for epochs = 10, KLDivLoss had a lesser accuracy of 57% than Cross Entropy and for epochs = 20, KLDivLoss had a higher accuracy of 67% than Cross Entropy which had an accuracy of 62%.

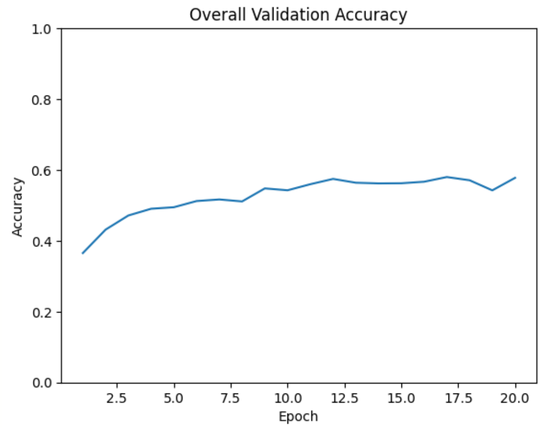
4.3 Effect of Data Augmentation

```
Accuracy on test set: 57.76%

Class-wise accuracies:
      Accuracy of airplane : 61 %
      Accuracy of automobile : 73 %
      Accuracy of  bird : 38 %
      Accuracy of   cat : 27 %
      Accuracy of  deer : 31 %
      Accuracy of   dog : 58 %
      Accuracy of  frog : 70 %
      Accuracy of horse : 71 %
      Accuracy of  ship : 69 %
      Accuracy of truck : 76 %
```







Observations:

- Turning off data augmentation: Without data augmentation, the model may not be exposed to a wide variety of image transformations and may not be able to learn robust features that generalize well to unseen data. This can lead to overfitting to the training data, which can be seen by a higher training accuracy and lower validation accuracy compared to training with data augmentations.
- Class-wise accuracy: Without data augmentation, the class-wise accuracy may vary depending on the specific class and its distinct features. Classes with simpler or more distinct features may still perform well, while classes with more complex features or less training data may suffer from overfitting and perform poorly.

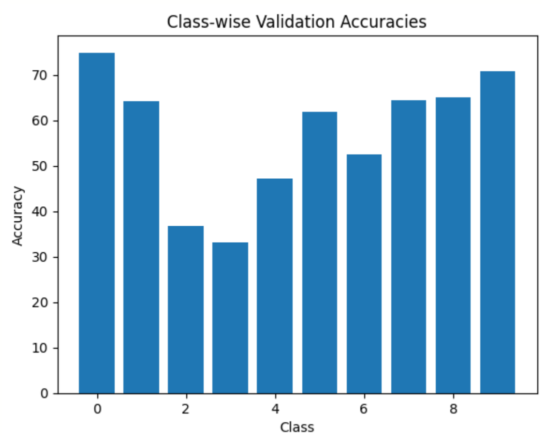
Results:

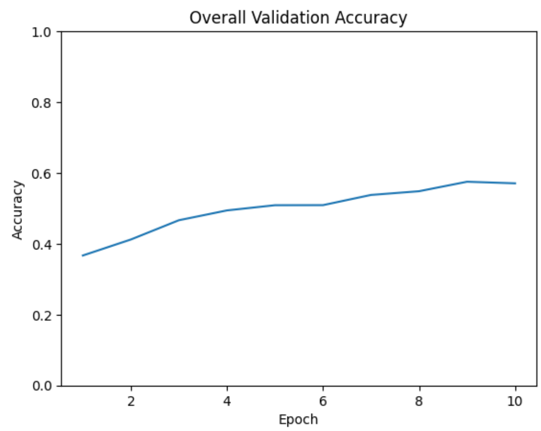
- Overall accuracy: Training with data augmentations may lead to a higher overall accuracy due to the increased exposure to a wide range of image transformations and robust feature learning. Without data augmentations, the model may overfit to the training data and have a lower validation accuracy, leading to a lower overall accuracy.
- Training accuracy: Without data augmentations, the training accuracy may be higher compared to training with data augmentations due to the overfitting of the model to the training data.

PART 1

```
Accuracy on test set: 57.06%
    Class-wise accuracies:
```

```
Accuracy of airplane: 74 %
Accuracy of automobile: 64 %
Accuracy of bird: 36 %
Accuracy of cat: 33 %
Accuracy of deer: 47 %
Accuracy of dog: 61 %
Accuracy of frog: 52 %
Accuracy of horse: 64 %
Accuracy of ship: 65 %
Accuracy of truck: 70 %
```




Overall Validation Accuracy
Accuracy
Epoch
0.0
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10