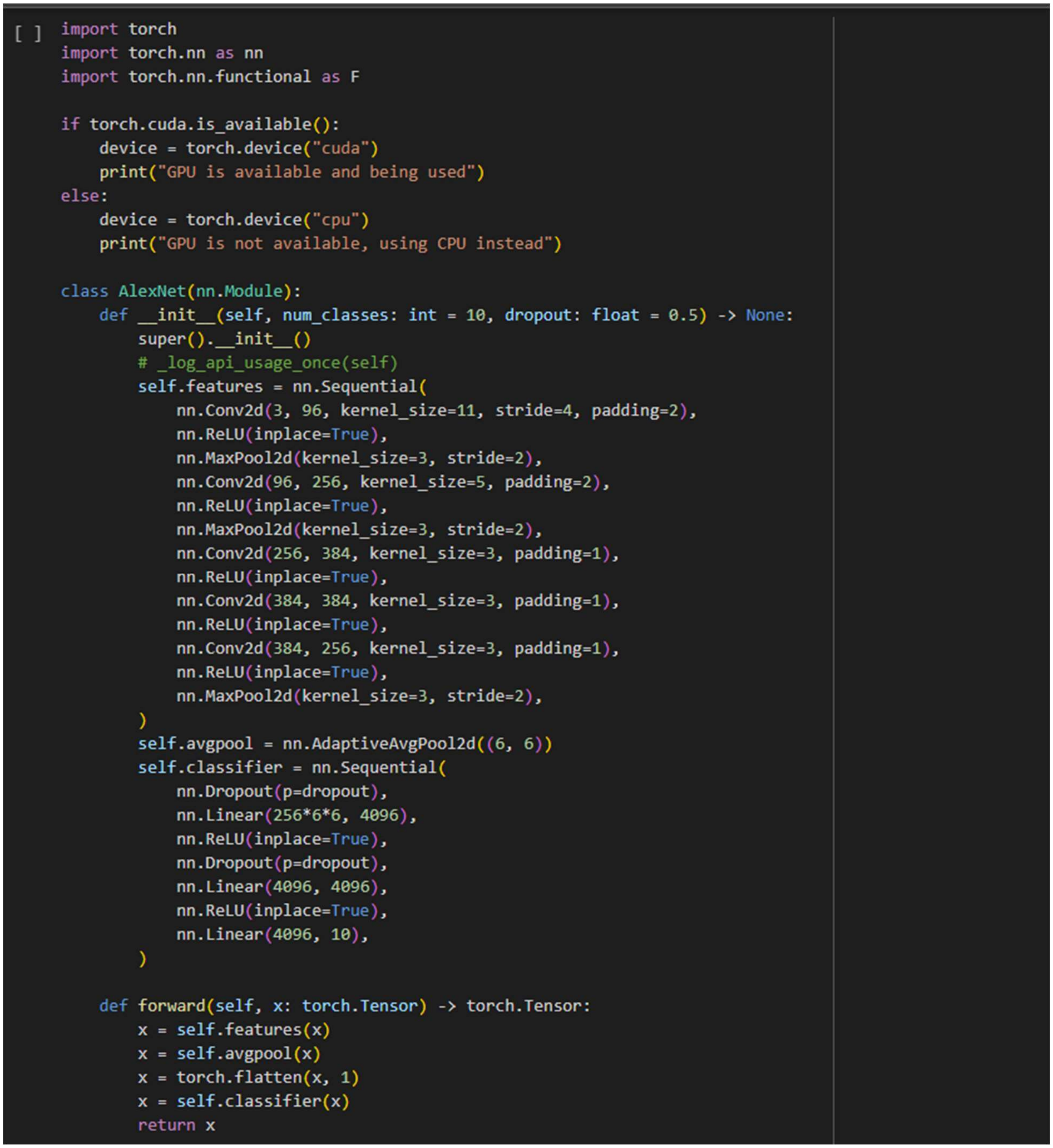
Lab 3 Report

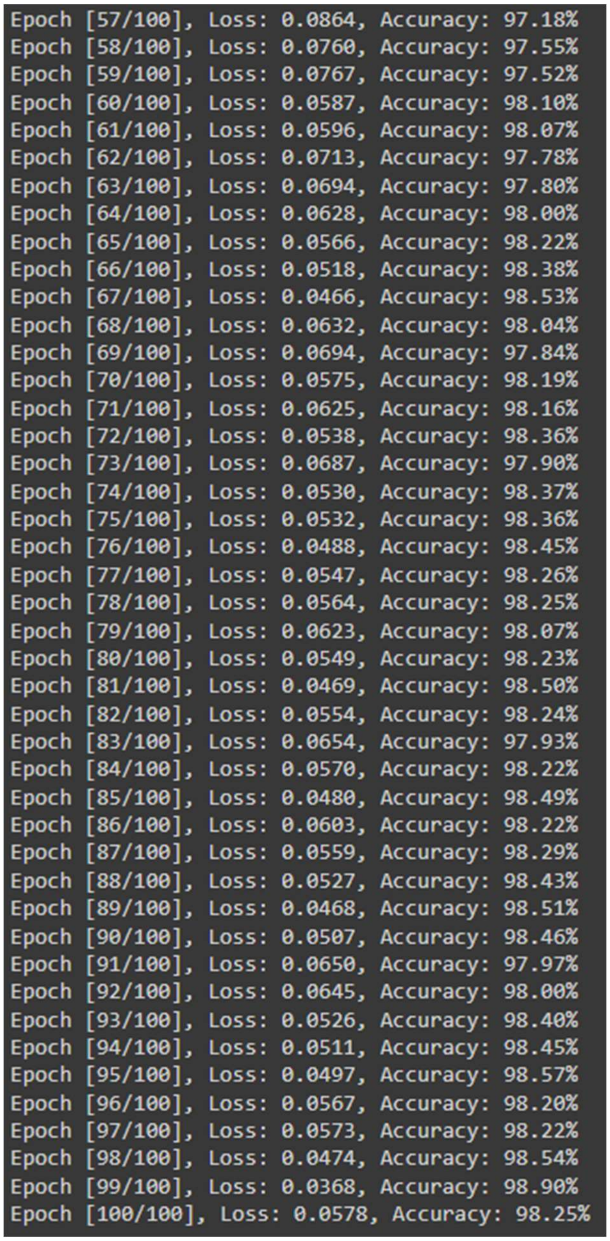
Aaryan Agarwal

Task 1

Alexnet Architecture

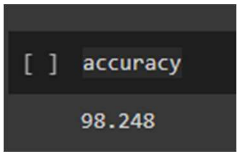


```
import torch
import torch.nn as nn
import torch.nn.functional as F

if torch.cuda.is_available():
    device = torch.device("cuda")
    print("GPU is available and being used")
else:
    device = torch.device("cpu")
    print("GPU is not available, using CPU instead")

class AlexNet(nn.Module):
    def __init__(self, num_classes: int = 10, dropout: float = 0.5) -> None:
        super().__init__()
        # _log_api_usage_once(self)
        self.features = nn.Sequential(
            nn.Conv2d(3, 96, kernel_size=11, stride=4, padding=2),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
            nn.Conv2d(96, 256, kernel_size=5, padding=2),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
            nn.Conv2d(256, 384, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(384, 384, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.Conv2d(384, 256, kernel_size=3, padding=1),
            nn.ReLU(inplace=True),
            nn.MaxPool2d(kernel_size=3, stride=2),
        )
        self.avgpool = nn.AdaptiveAvgPool2d((6, 6))
        self.classifier = nn.Sequential(
            nn.Dropout(p=dropout),
            nn.Linear(256*6*6, 4096),
            nn.ReLU(inplace=True),
            nn.Dropout(p=dropout),
            nn.Linear(4096, 4096),
            nn.ReLU(inplace=True),
            nn.Linear(4096, 10),
        )

    def forward(self, x: torch.Tensor) -> torch.Tensor:
        x = self.features(x)
        x = self.avgpool(x)
        x = torch.flatten(x, 1)
        x = self.classifier(x)
        return x
```

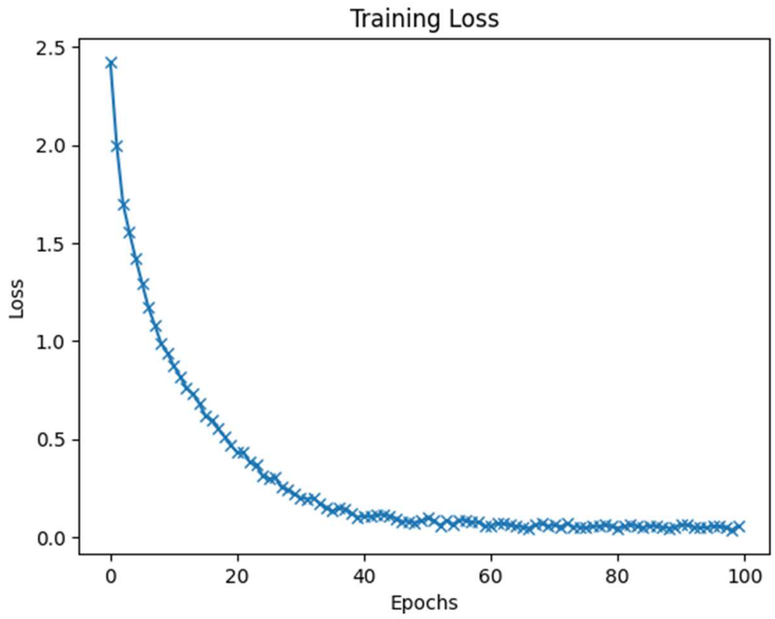


```
Epoch [57/100], Loss: 0.0864, Accuracy: 97.18%
Epoch [58/100], Loss: 0.0760, Accuracy: 97.55%
Epoch [59/100], Loss: 0.0767, Accuracy: 97.52%
Epoch [60/100], Loss: 0.0587, Accuracy: 98.10%
Epoch [61/100], Loss: 0.0596, Accuracy: 98.07%
Epoch [62/100], Loss: 0.0713, Accuracy: 97.78%
Epoch [63/100], Loss: 0.0694, Accuracy: 97.80%
Epoch [64/100], Loss: 0.0628, Accuracy: 98.00%
Epoch [65/100], Loss: 0.0566, Accuracy: 98.22%
Epoch [66/100], Loss: 0.0518, Accuracy: 98.38%
Epoch [67/100], Loss: 0.0466, Accuracy: 98.53%
Epoch [68/100], Loss: 0.0632, Accuracy: 98.04%
Epoch [69/100], Loss: 0.0694, Accuracy: 97.84%
Epoch [70/100], Loss: 0.0575, Accuracy: 98.19%
Epoch [71/100], Loss: 0.0625, Accuracy: 98.16%
Epoch [72/100], Loss: 0.0538, Accuracy: 98.36%
Epoch [73/100], Loss: 0.0687, Accuracy: 97.90%
Epoch [74/100], Loss: 0.0530, Accuracy: 98.37%
Epoch [75/100], Loss: 0.0532, Accuracy: 98.36%
Epoch [76/100], Loss: 0.0488, Accuracy: 98.45%
Epoch [77/100], Loss: 0.0547, Accuracy: 98.26%
Epoch [78/100], Loss: 0.0564, Accuracy: 98.25%
Epoch [79/100], Loss: 0.0623, Accuracy: 98.07%
Epoch [80/100], Loss: 0.0549, Accuracy: 98.23%
Epoch [81/100], Loss: 0.0469, Accuracy: 98.50%
Epoch [82/100], Loss: 0.0554, Accuracy: 98.24%
Epoch [83/100], Loss: 0.0654, Accuracy: 97.93%
Epoch [84/100], Loss: 0.0570, Accuracy: 98.22%
Epoch [85/100], Loss: 0.0480, Accuracy: 98.49%
Epoch [86/100], Loss: 0.0603, Accuracy: 98.22%
Epoch [87/100], Loss: 0.0559, Accuracy: 98.29%
Epoch [88/100], Loss: 0.0527, Accuracy: 98.43%
Epoch [89/100], Loss: 0.0468, Accuracy: 98.51%
Epoch [90/100], Loss: 0.0507, Accuracy: 98.46%
Epoch [91/100], Loss: 0.0650, Accuracy: 97.97%
Epoch [92/100], Loss: 0.0645, Accuracy: 98.00%
Epoch [93/100], Loss: 0.0526, Accuracy: 98.40%
Epoch [94/100], Loss: 0.0511, Accuracy: 98.45%
Epoch [95/100], Loss: 0.0497, Accuracy: 98.57%
Epoch [96/100], Loss: 0.0567, Accuracy: 98.20%
Epoch [97/100], Loss: 0.0573, Accuracy: 98.22%
Epoch [98/100], Loss: 0.0474, Accuracy: 98.54%
Epoch [99/100], Loss: 0.0368, Accuracy: 98.90%
Epoch [100/100], Loss: 0.0578, Accuracy: 98.25%
```

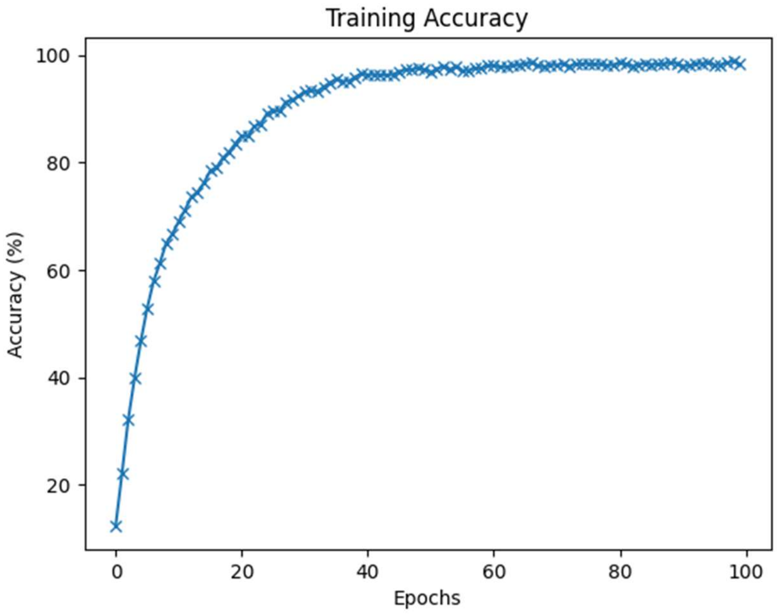
Final accuracy



```
[ ]  accuracy

     98.248
```



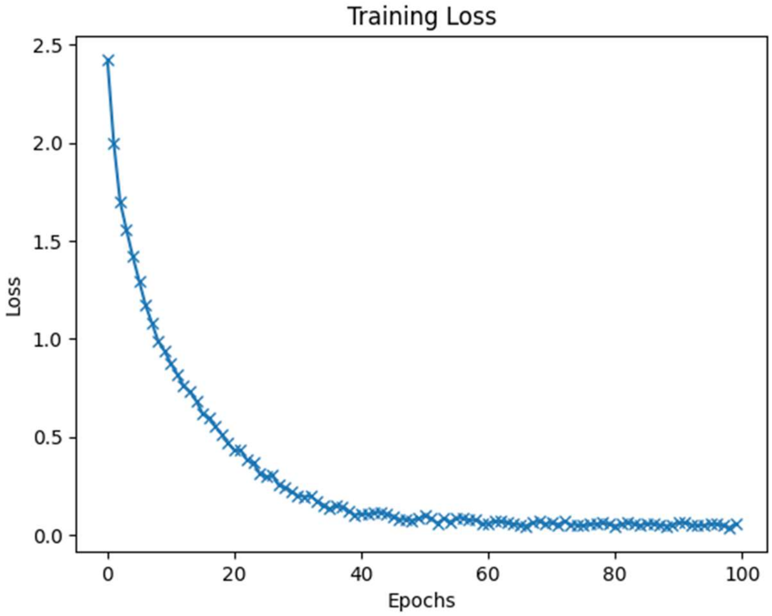
Training Loss

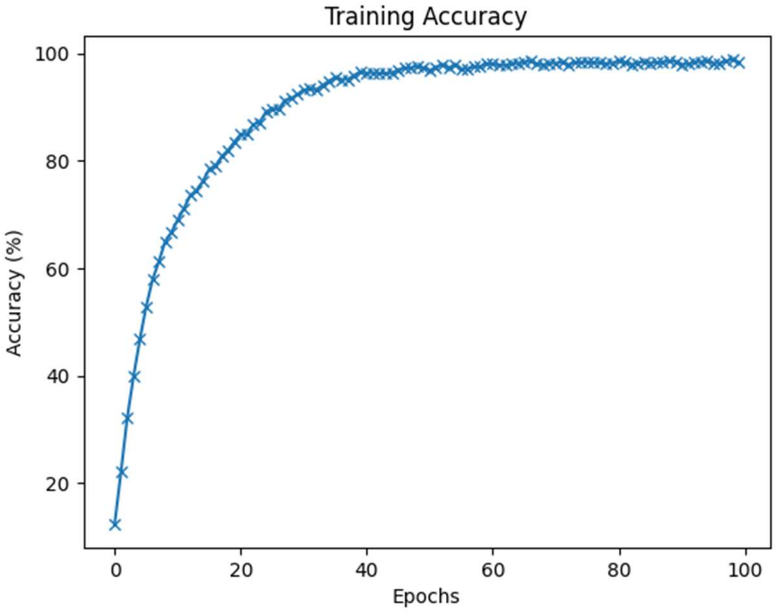


Training Accuracy

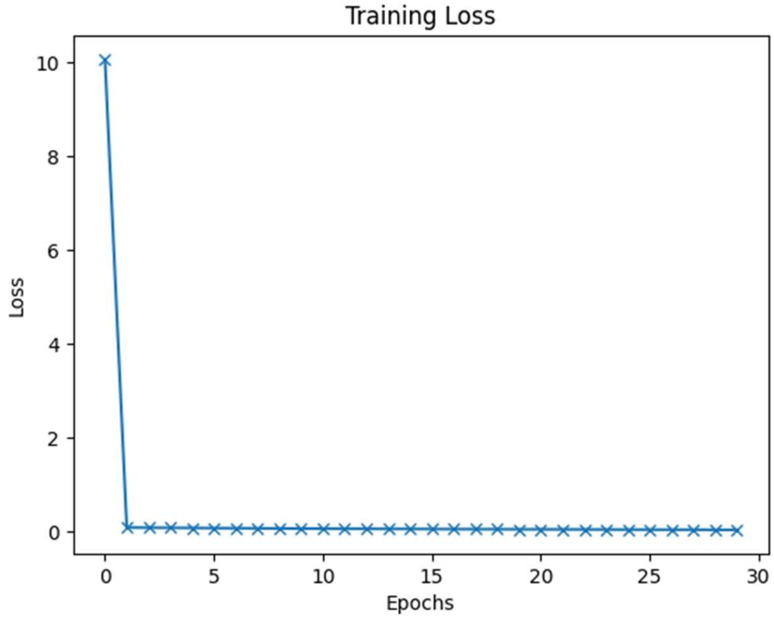
Task 2

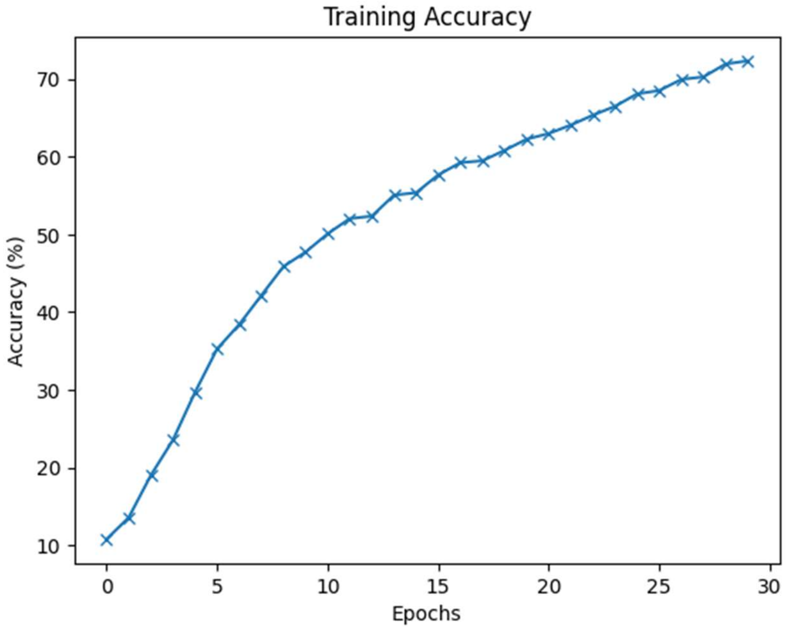
Cross Entropy Loss





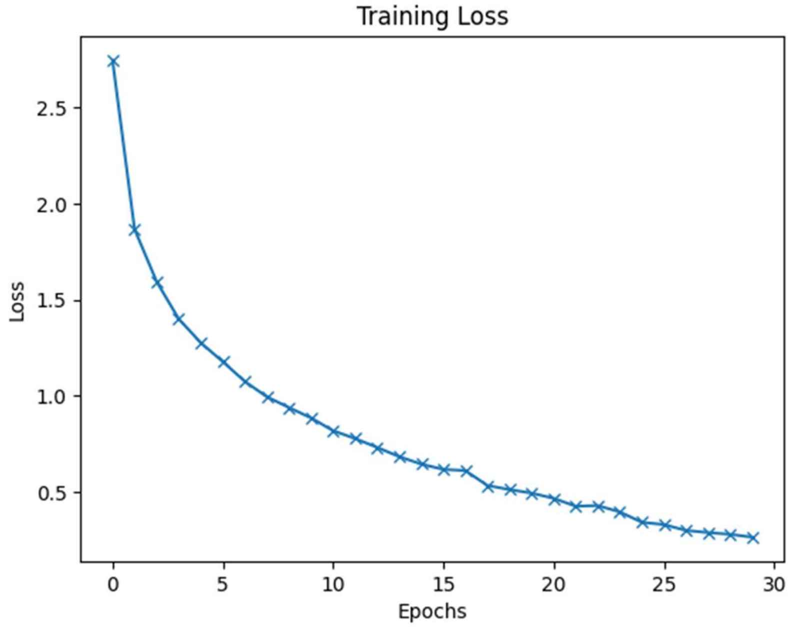
MSE Loss

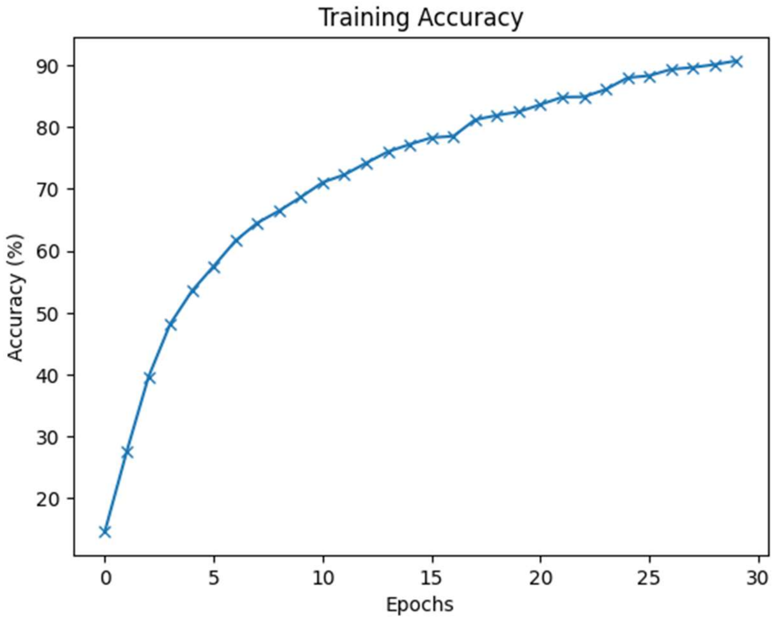




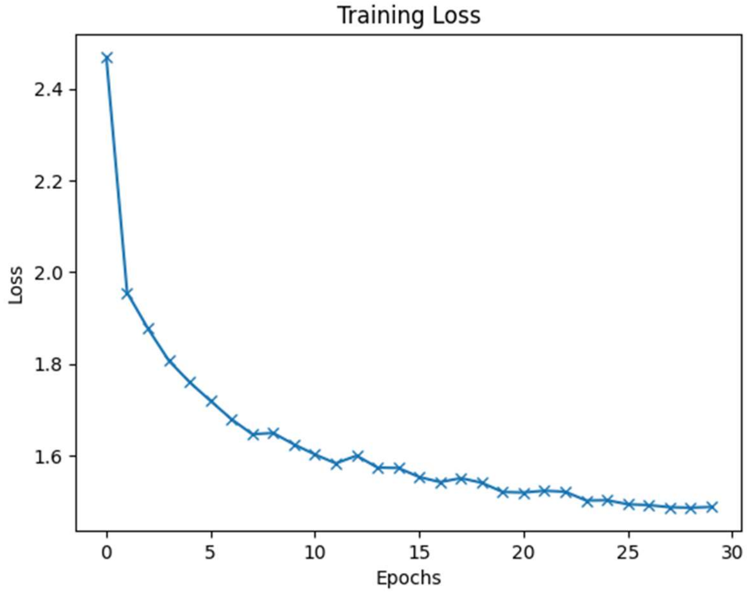
The loss is lesser in MSE but the accuracy is greater in CrossEntropyLoss.

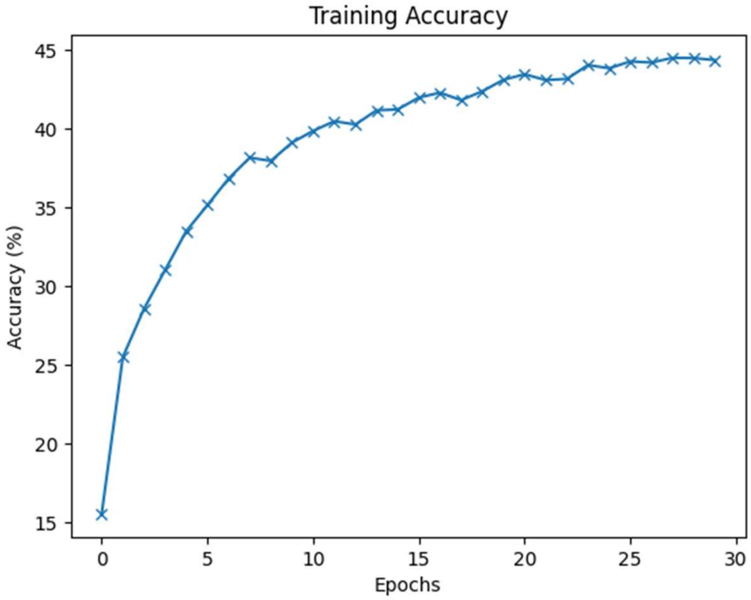
4 convoltion layers





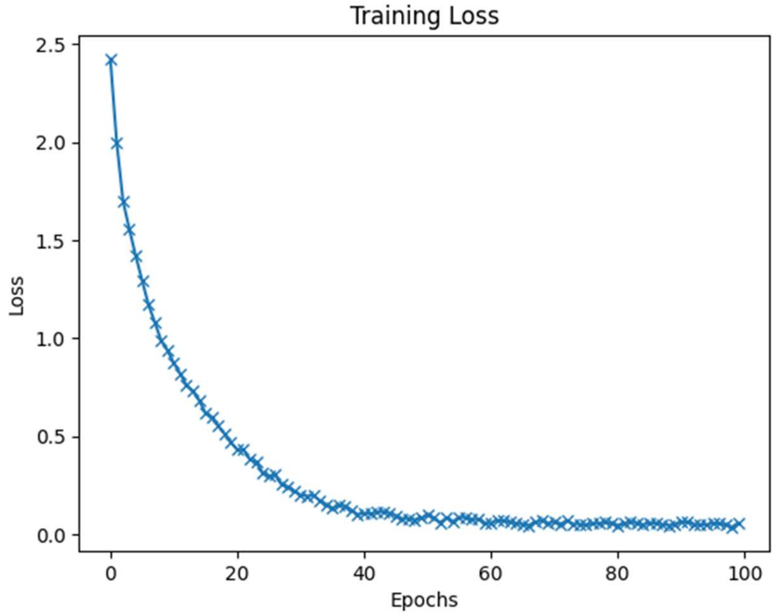
2 convolution layers

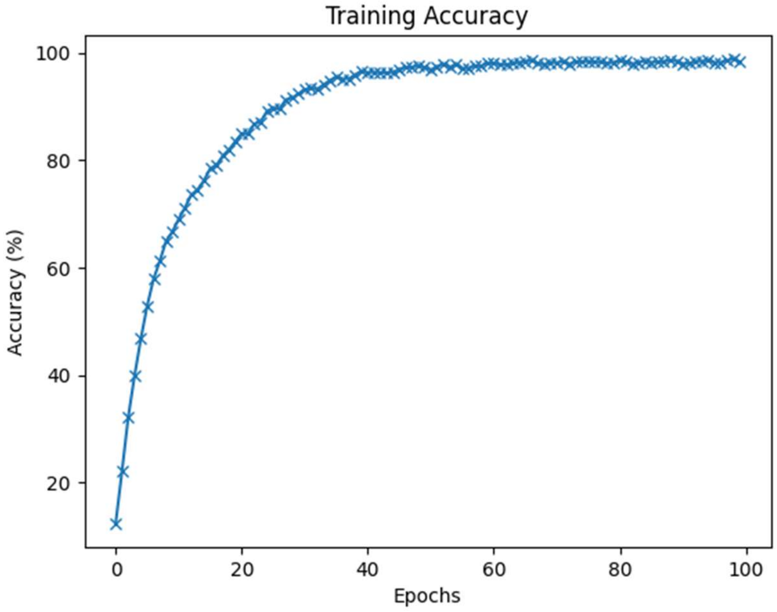




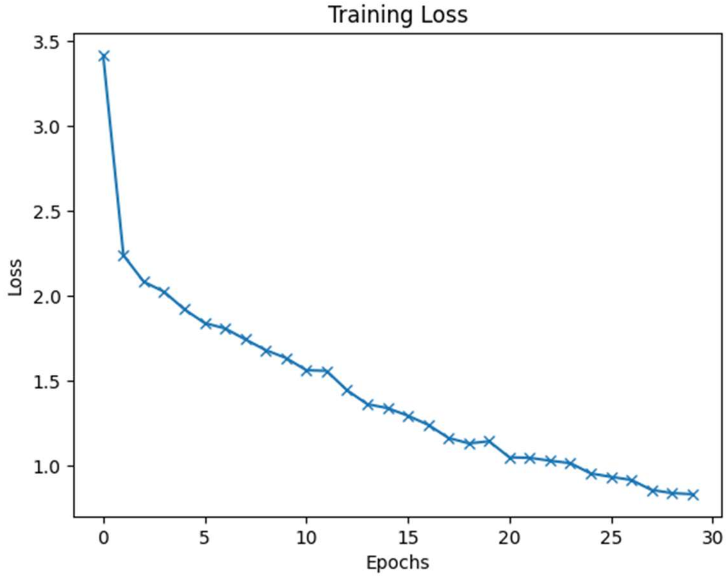
The accuracy decreases with lesser number of convolution layers.

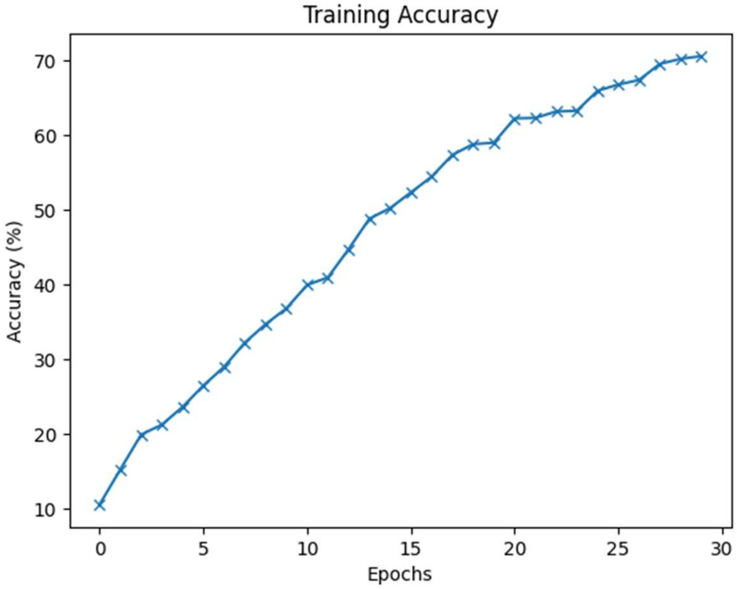
Using Relu


Training Loss
Loss
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100
Epochs


Training Accuracy
Accuracy (%)
100
80
60
40
20
0
20
40
60
80
100
Epochs

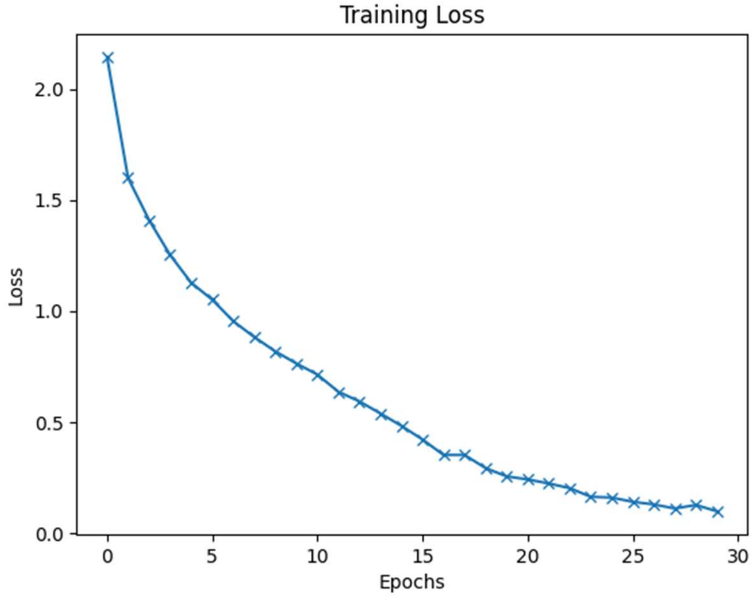
Tanh

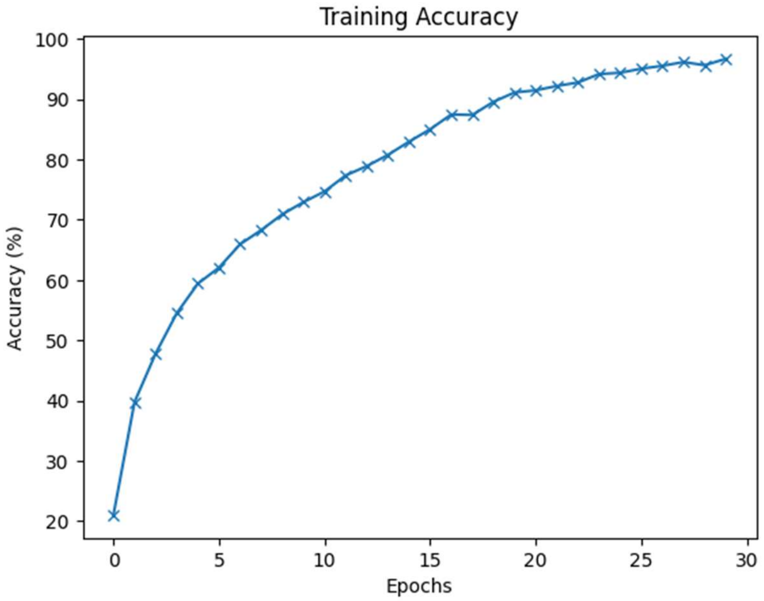




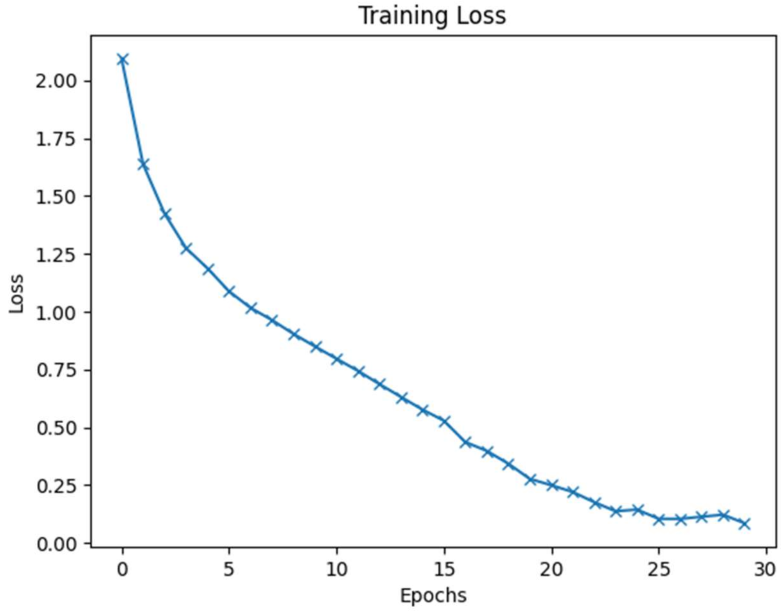
Tanh has a lesser accuracy than ReLU.

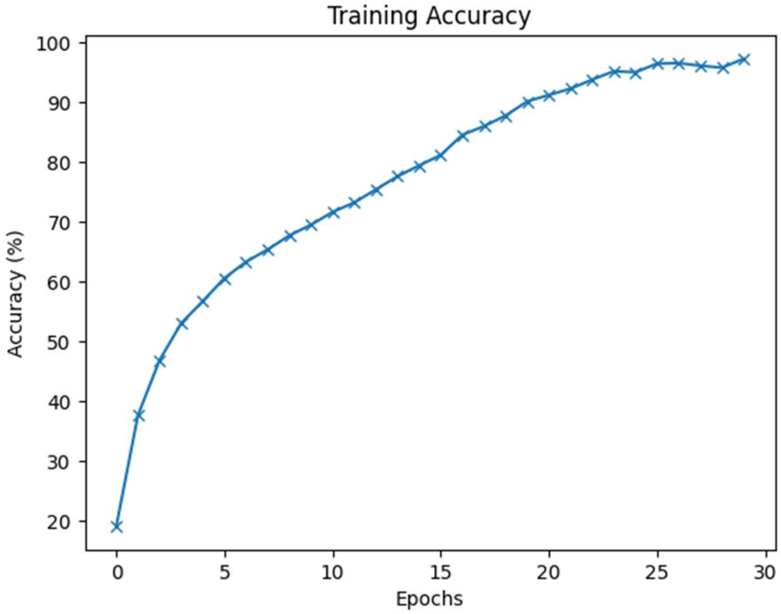
2 FC Layers


Training Loss
Loss
Epochs
0.0
0.5
1.0
1.5
2.0
0
5
10
15
20
25
30


Training Accuracy
Accuracy (%)
Epochs
20
30
40
50
60
70
80
90
100
0
5
10
15
20
25
30

1 FC Layer





Not a very significant change by altering FC Layers.