ASSIGNMENT 1: *CIFAR100 IMAGE CLASSIFICATION WITH KERAS*

INTRODUCTION

CIFAR-100

- *100 CLASSES WITH 600 IMAGES EACH*

Process:

- Load and preprocess dataset
  - normalizing pixel values and converting class labels into one-hot encoded vectors
- construct a CNN architecture
  - Used established model e.g. VGG-16
  - trained it on the preprocessed dataset

Training:

- Determined loss and accuracy for training and validation

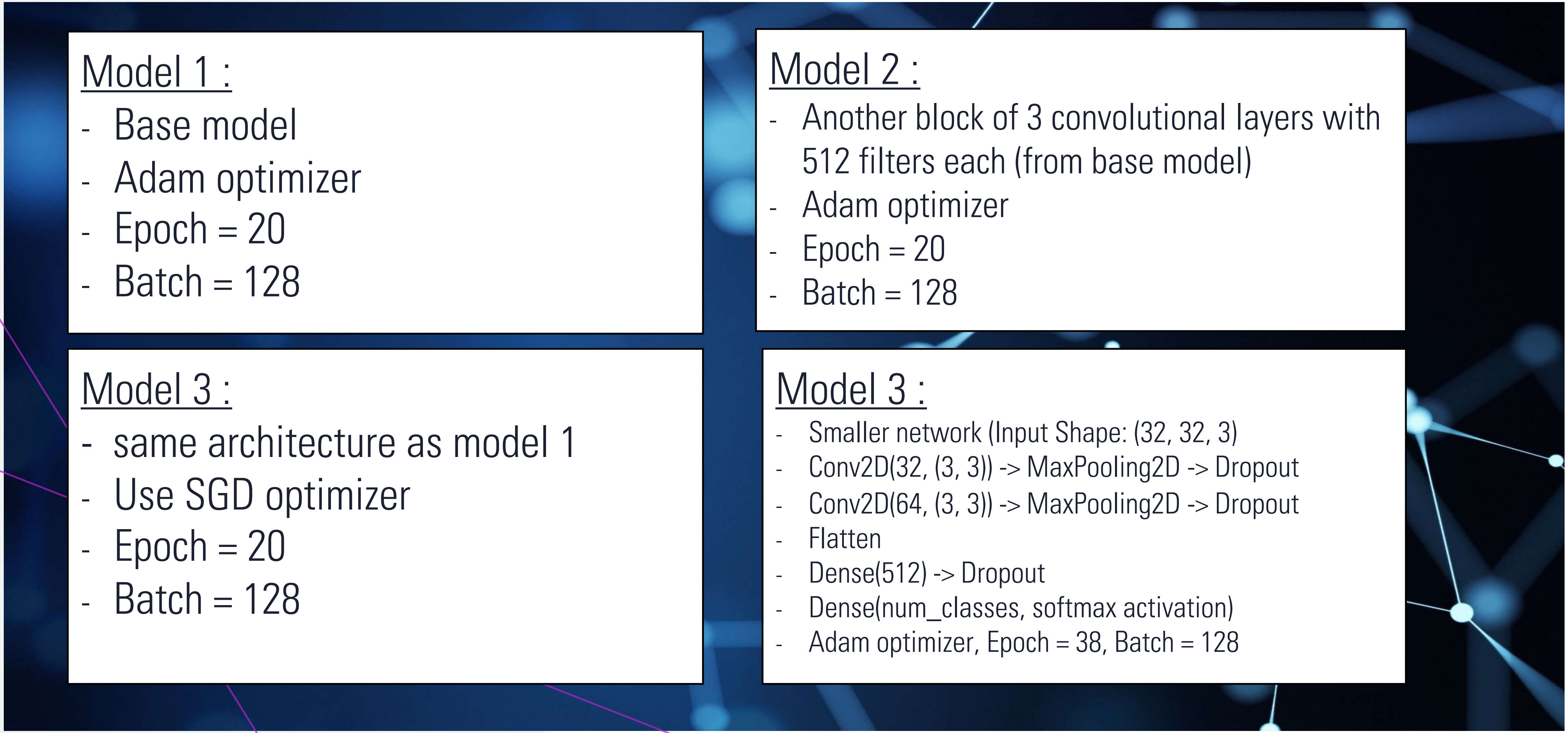
Experiment:

- Trained different models using base vgg-16 model, extra block of layers, use SGD optimization, and a smaller architecture

MODEL ARCHITECTURE

- VGG-16 :
- Input Layer:
- - Input dimensions: (32, 32, 3) for CIFAR-100 dataset.
- Convolutional Layers:
- - 2 convolutional layers with 64 filters each, kernel size of (3, 3), and ReLU activation function.
- - 2 convolutional layers with 128 filters each, kernel size of (3, 3), and ReLU activation function.
- - 3 convolutional layers with 256 filters each, kernel size of (3, 3), and ReLU activation function.
- - 3 convolutional layers with 512 filters each, kernel size of (3, 3), and ReLU activation function.
- Max Pooling Layers:
- - After every two convolutional layers, there's a max-pooling layer with a pool size of (2, 2) and stride of (2, 2).
- Fully Connected Layers:
- - After the convolutional layers, there are three fully connected layers with ReLU activation function.
- - 2 fully connected layers have 4096 units each.
- - The last fully connected layer has 100 units with softmax activation, corresponding to the 100 classes in the CIFAR-100 dataset.
- Regularization:
- - Dropout layers with a dropout rate of 0.5 are added after the first two fully connected layers.
- Batch Normalization:
- - Batch normalization layers are added after every max-pooling layer to stabilize and accelerate the training process.

MODELS



Model 1 :

- Base model
- Adam optimizer
- Epoch = 20
- Batch = 128

Model 2 :

- Another block of 3 convolutional layers with 512 filters each (from base model)
- Adam optimizer
- Epoch = 20
- Batch = 128

Model 3 :

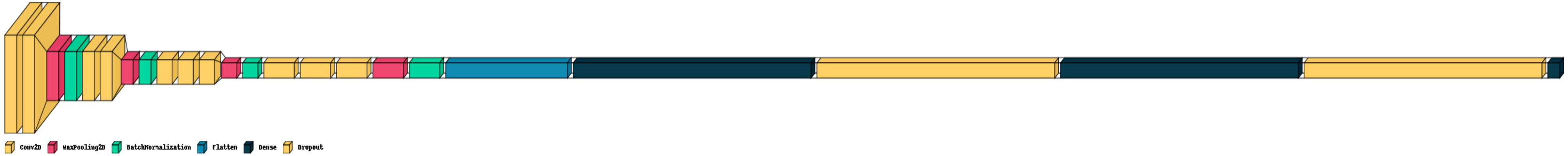
- same architecture as model 1
- Use SGD optimizer
- Epoch = 20
- Batch = 128

Model 3 :

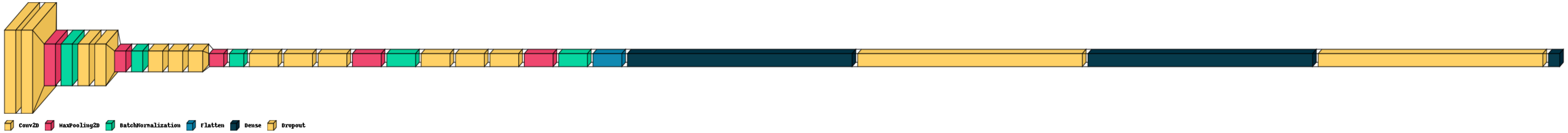
- Smaller network (Input Shape: (32, 32, 3)
- Conv2D(32, (3, 3)) -> MaxPooling2D -> Dropout
- Conv2D(64, (3, 3)) -> MaxPooling2D -> Dropout
- Flatten
- Dense(512) -> Dropout
- Dense(num_classes, softmax activation)
- Adam optimizer, Epoch = 38, Batch = 128

KERAS VISUAL

Model 1: Base model

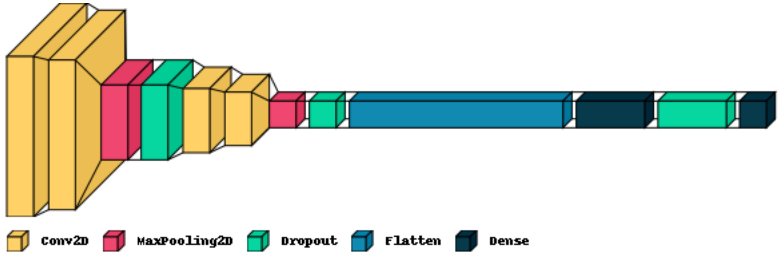


Model 2: With another block of 3 convolutional layers (512 filters)

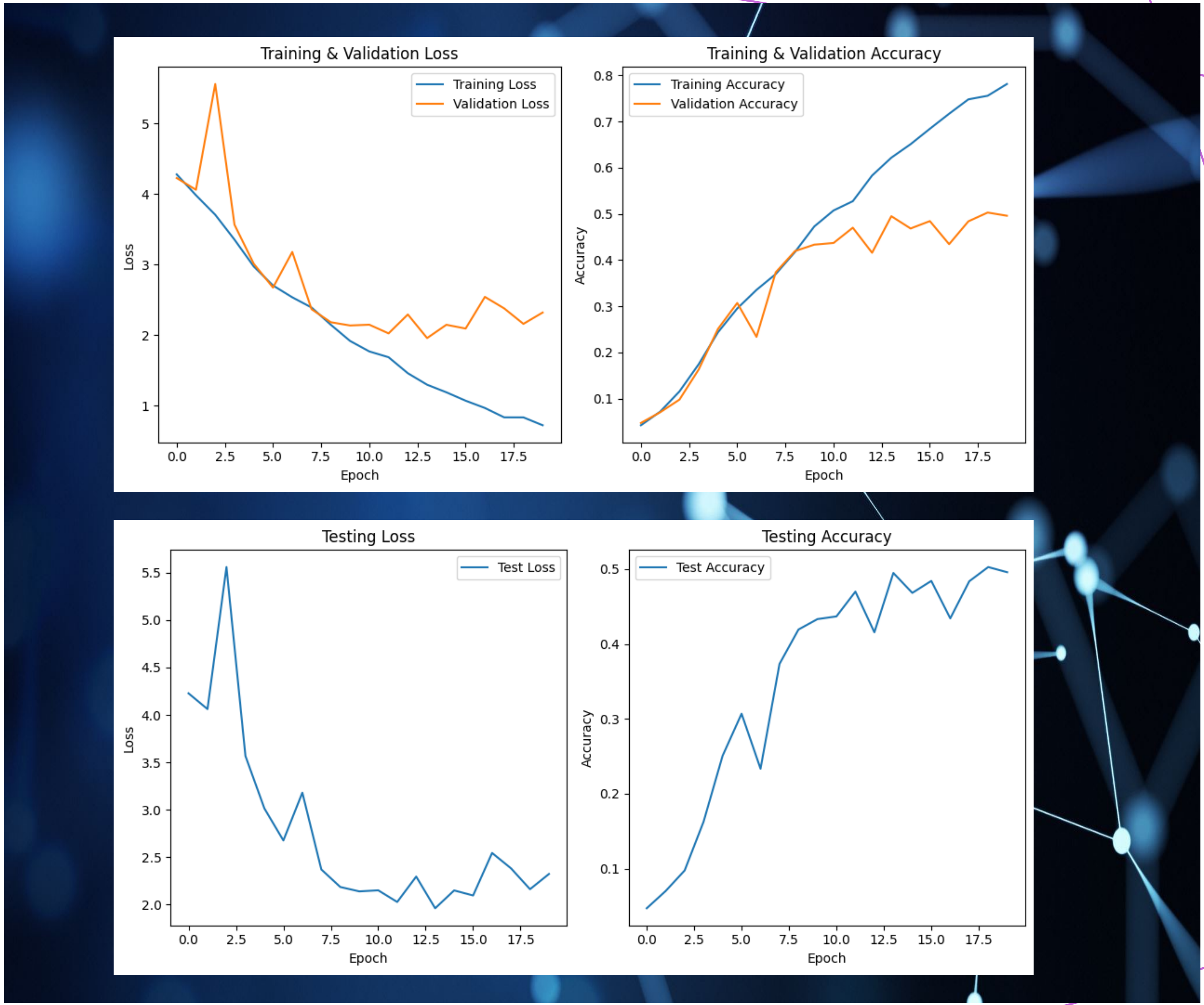


Model 3: same as model 2

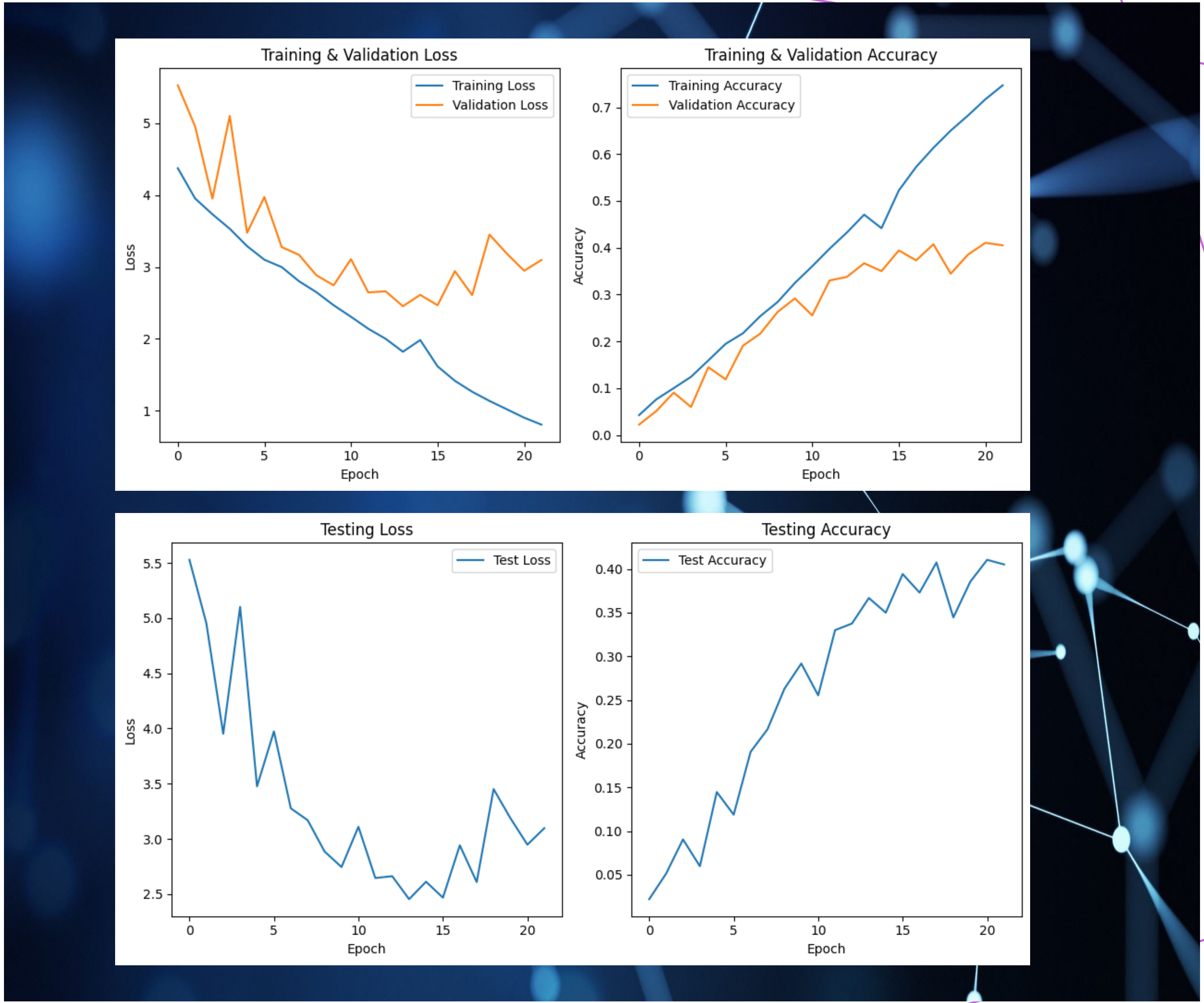
Model 4: With another block of 3 convolutional layers(512 filters)



MODEL 1

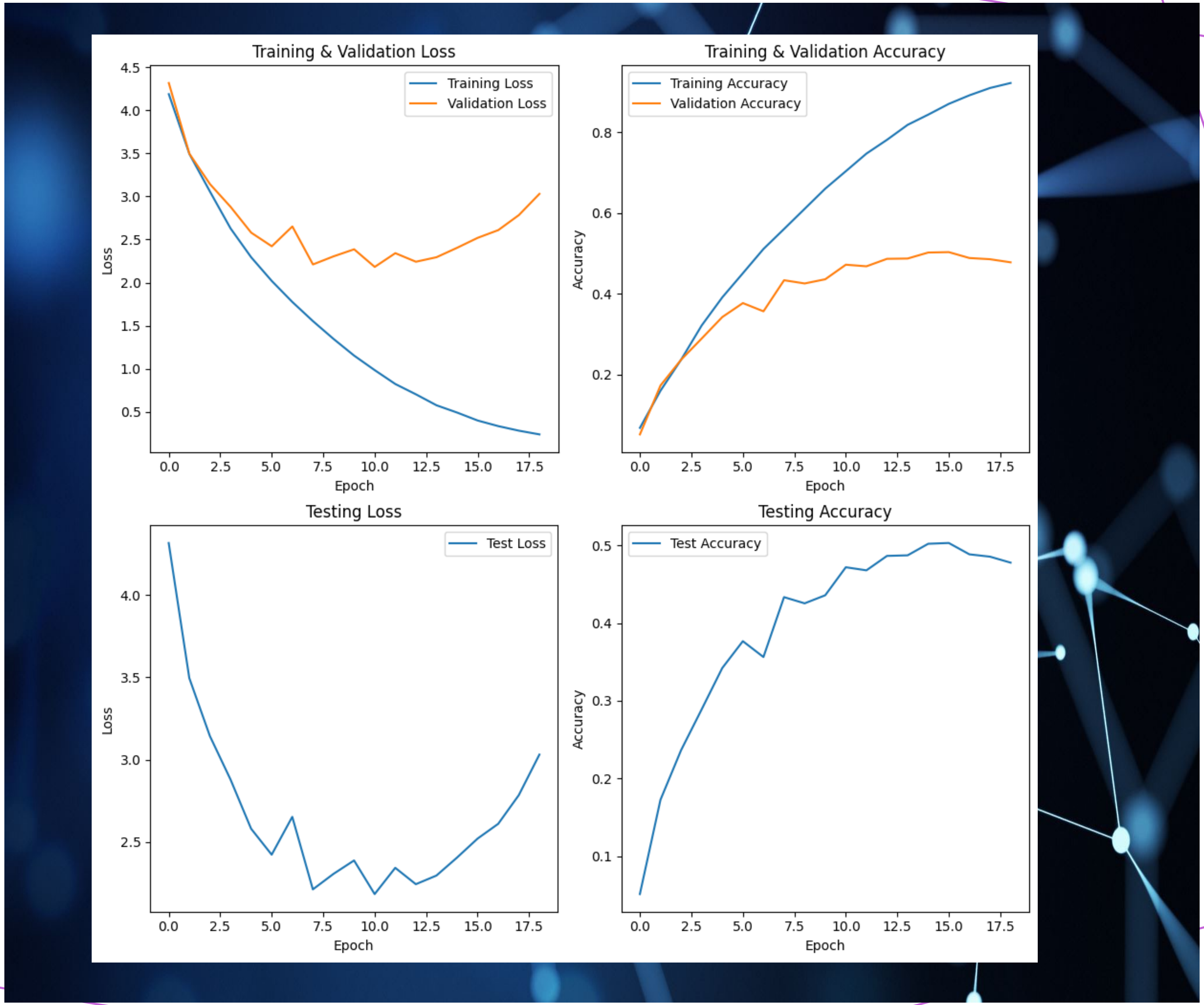

Training & Validation Loss
Training Loss
Validation Loss
Loss
1
2
3
4
5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch
Training & Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch
Testing Loss
Test Loss
Loss
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch
Testing Accuracy
Test Accuracy
Accuracy
0.1
0.2
0.3
0.4
0.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch

MODEL 2



Training & Validation Loss
Training Loss
Validation Loss
Loss
1
2
3
4
5
0
5
10
15
20
Epoch
Training & Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
5
10
15
20
Epoch
Testing Loss
Test Loss
Loss
2.5
3.0
3.5
4.0
4.5
5.0
5.5
0
5
10
15
20
Epoch
Testing Accuracy
Test Accuracy
Accuracy
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0
5
10
15
20
Epoch

MODEL 3



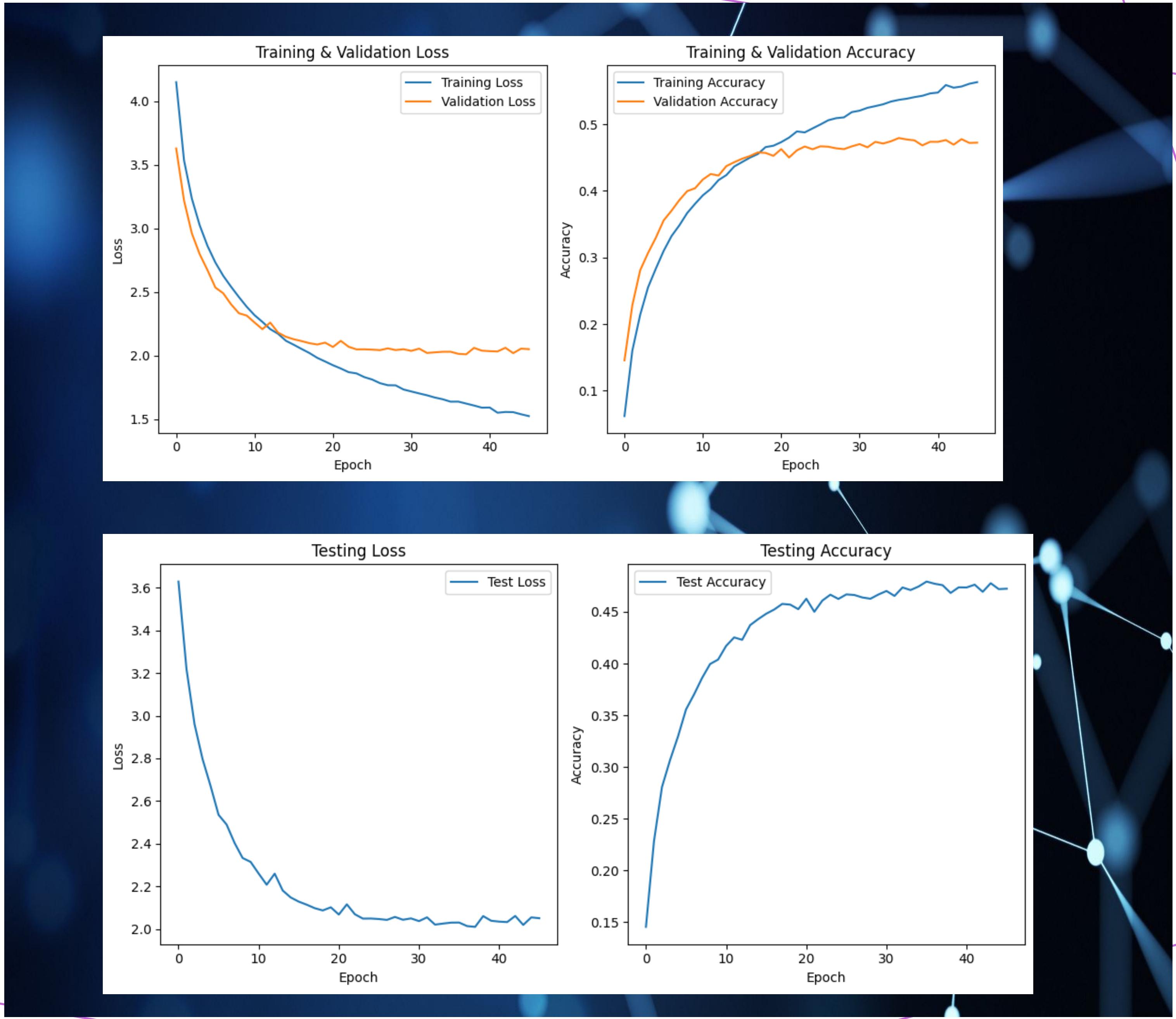
Training & Validation Loss

Training & Validation Accuracy

Testing Loss

Testing Accuracy

MODEL 4



Training & Validation Loss
Training Loss
Validation Loss
Loss
4.0
3.5
3.0
2.5
2.0
1.5
0
10
20
30
40
Epoch
Training & Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
0.5
0.4
0.3
0.2
0.1
0
10
20
30
40
Epoch
Testing Loss
Test Loss
Loss
3.6
3.4
3.2
3.0
2.8
2.6
2.4
2.2
2.0
0
10
20
30
40
Epoch
Testing Accuracy
Test Accuracy
Accuracy
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0
10
20
30
40
Epoch

MODEL RESULTS

- Model 1 (BEST)

 - Training accuracy: 0.78
- Test loss: 2.32
- Test accuracy: 0.50

- Model 2

 - Training accuracy: 0.47
- Test loss: 2.45
- Test accuracy: 0.37

- Model 3

 - Training accuracy: 0.70
- Test loss: 2.18
- Test accuracy: 0.47

- Model 4

 - Training accuracy: 0.54
- Test loss: 2.01
- Test accuracy: 0.48

IMAGE LABELING

- display of test images, only 2 correctly labeled

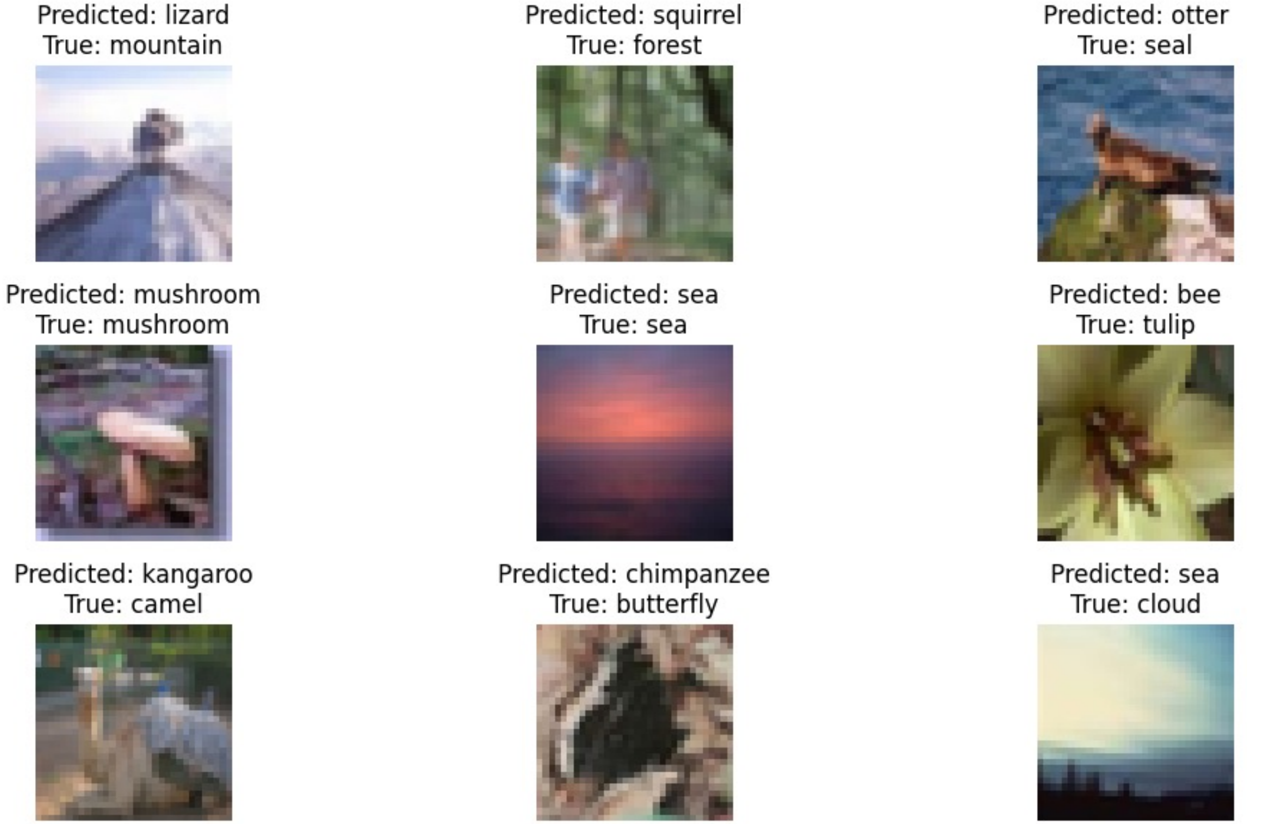


IMAGE LABELING

- Selection of correctly labeled images

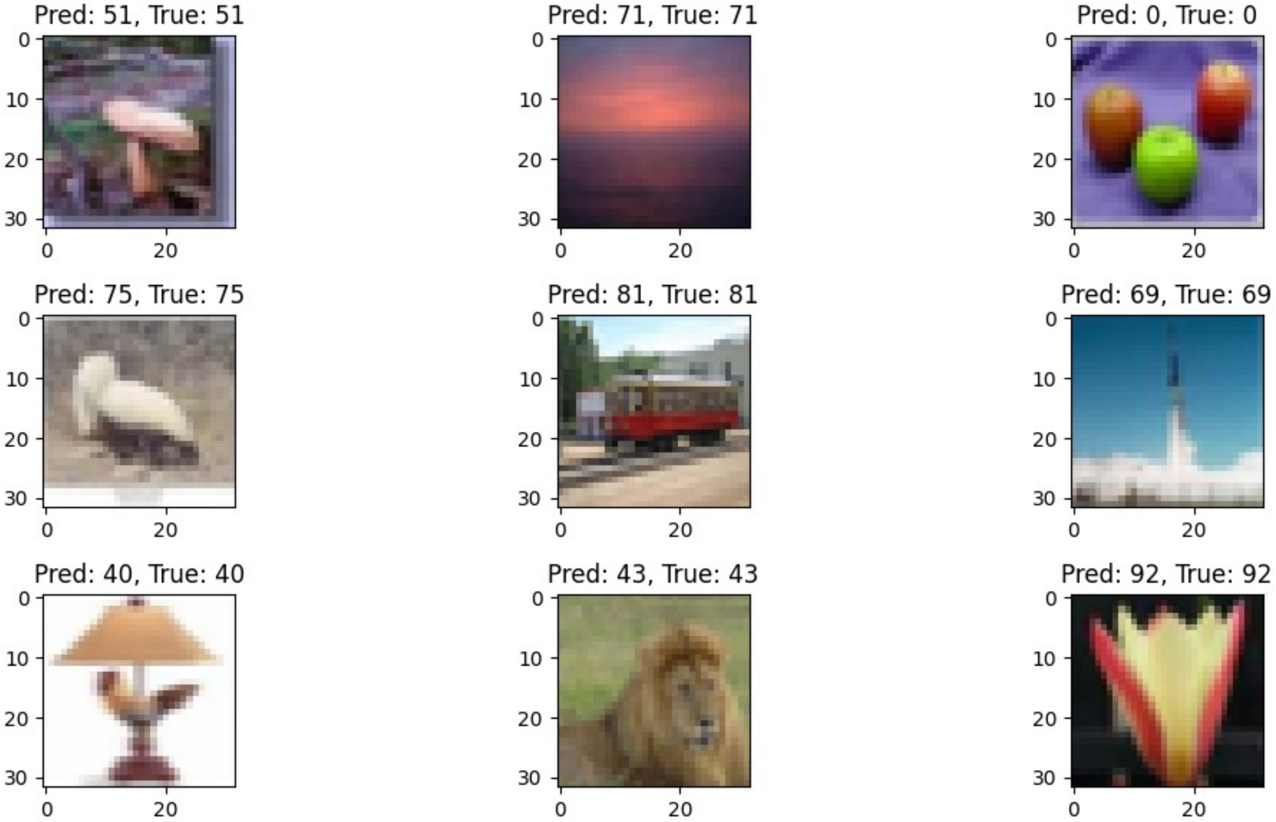
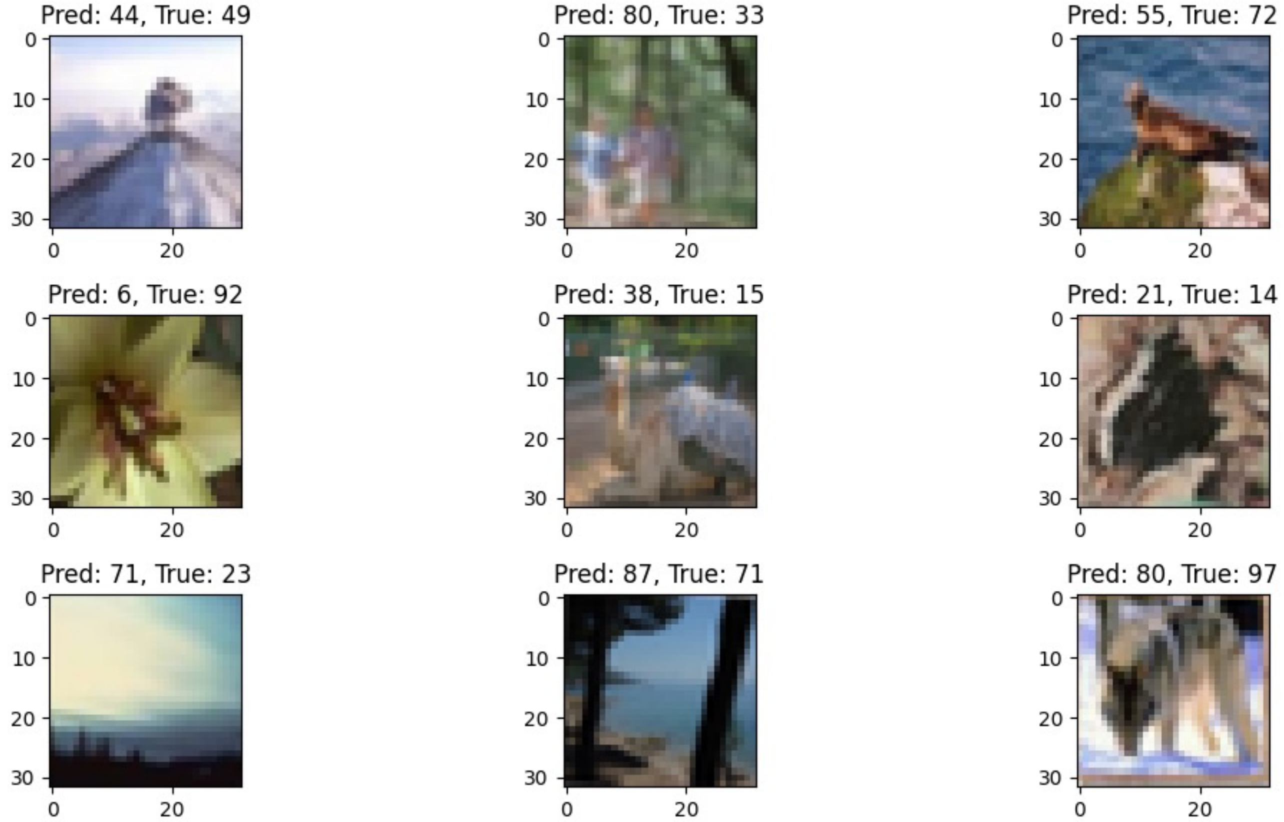


IMAGE LABELING

- Selection of incorrectly labeled images



CONCLUSION

- Model 1 had the best test accuracy compared to the other models with a test accuracy of 0.50.
- I could improve the training and testing accuracy using another optimizer or architecture.