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 with batch normalization, no batch normalization, and added another block (4 instead of 3)

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.

MODEL

Model 1 :

- No batch normalization
- Epoch = 20
- Batch = 164

Model 2 :

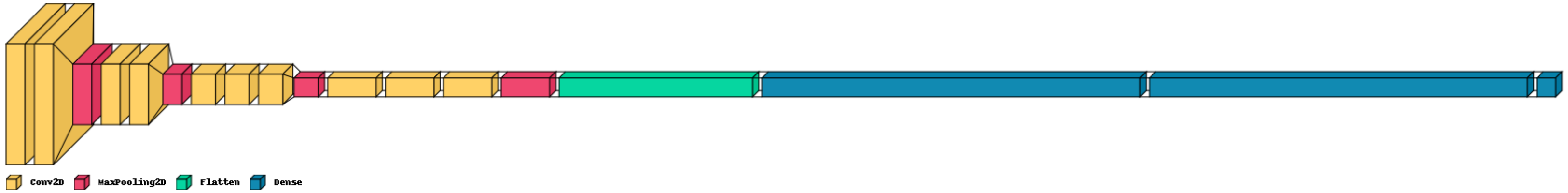
- Batch normalization
- Epoch = 20
- Batch = 164

Model 3 :

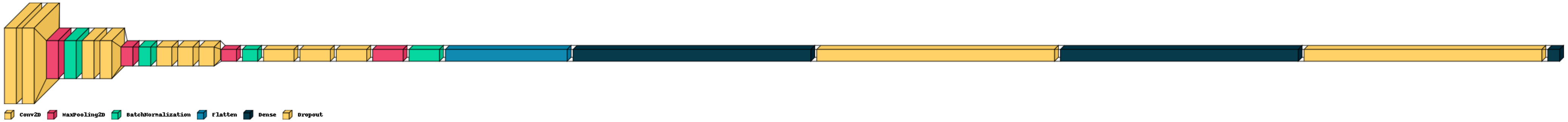
- Another block of 3 convolutional layers with 512 filters each
- Epoch = 20
- Batch = 164

KERAS VISUAL

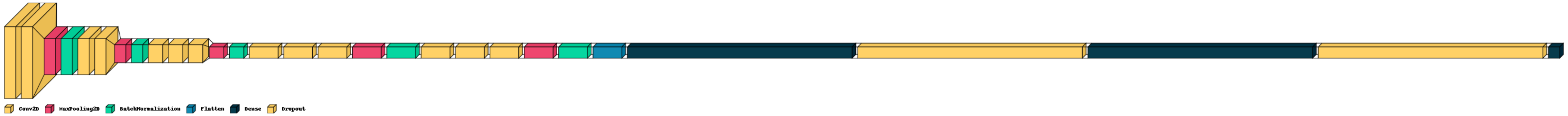
Without batch normalization



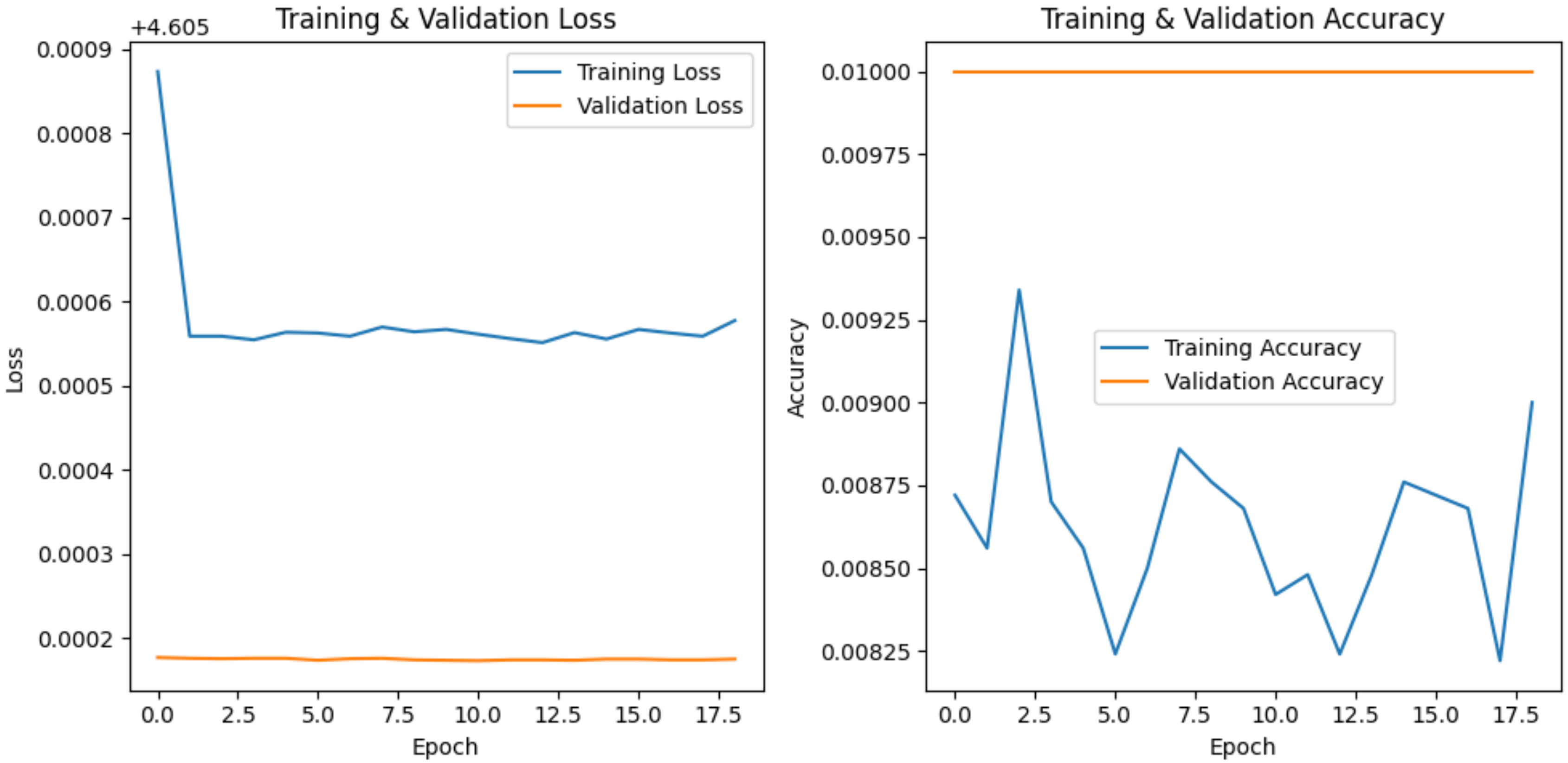
With batch normalization



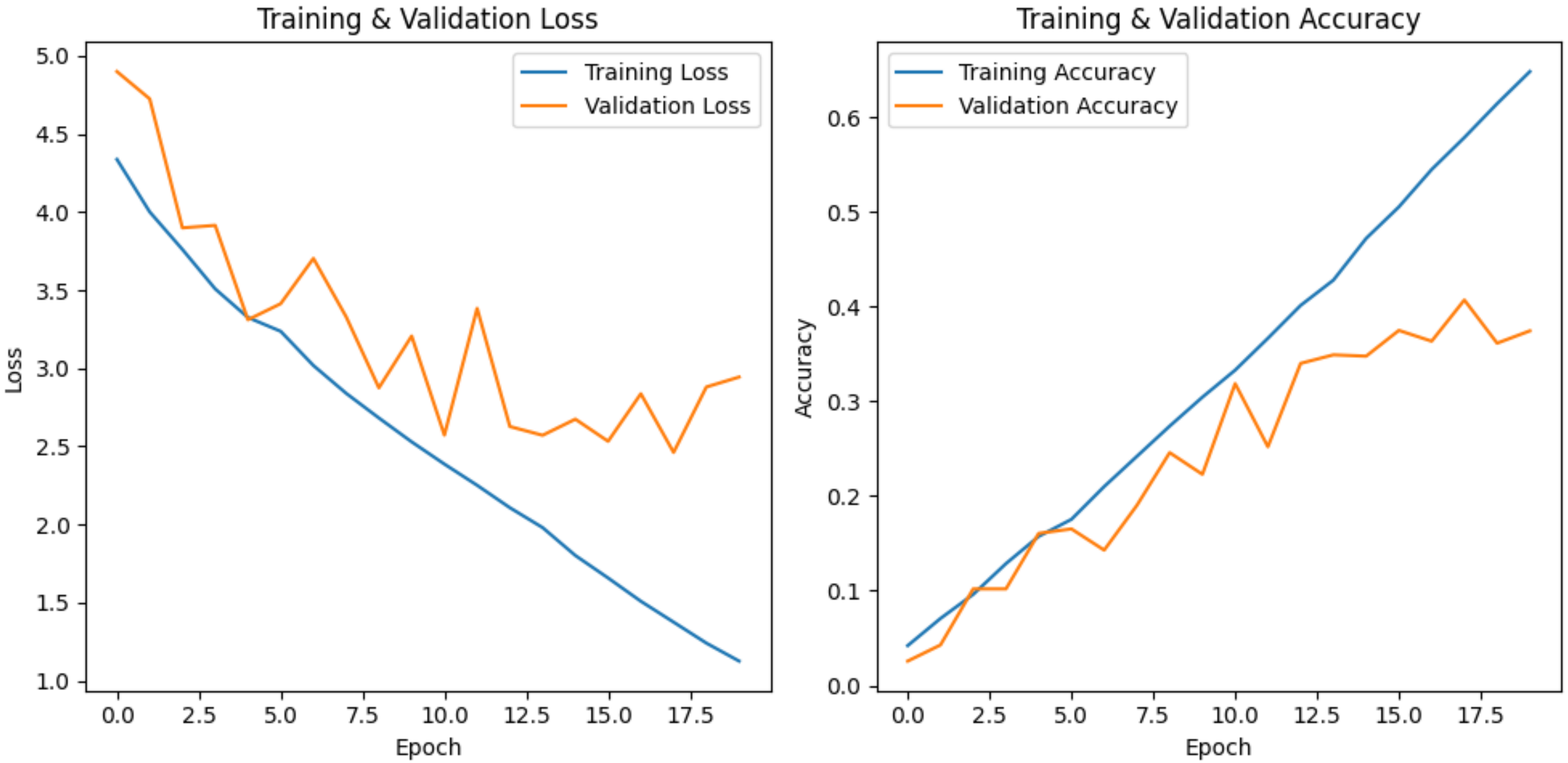
With another block of 3 convolutional layers(512 filters)



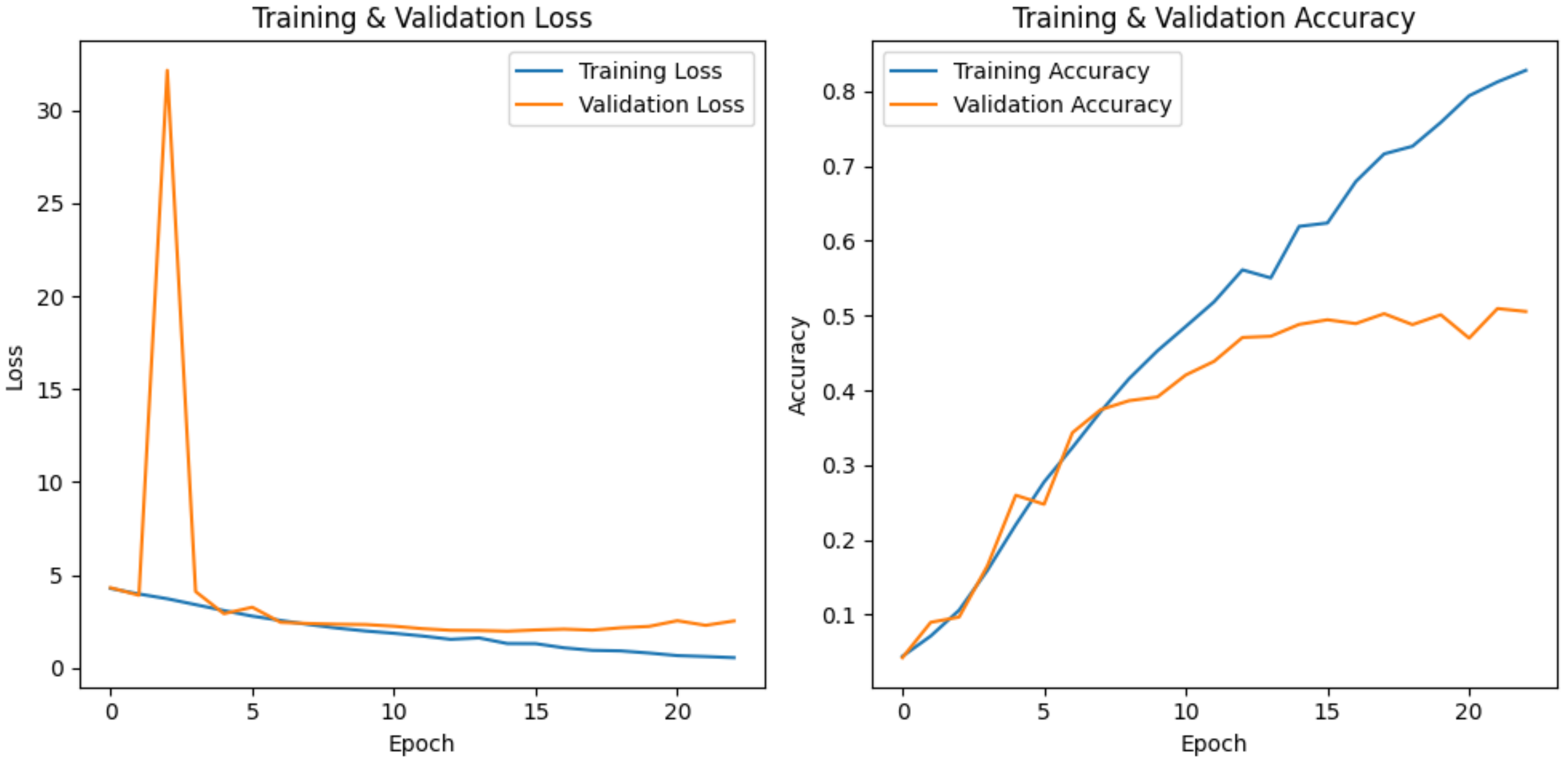
MODEL 1


Training & Validation Loss
+4.605
0.0009
0.0008
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
Loss
Training Loss
Validation Loss
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch
Training & Validation Accuracy
0.01000
0.00975
0.00950
0.00925
0.00900
0.00875
0.00850
0.00825
Accuracy
Training Accuracy
Validation Accuracy
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
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
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
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch

MODEL 3


Training & Validation Loss
Training Loss
Validation Loss
Loss
0
5
10
15
20
25
30
Epoch
0
5
10
15
20
Training & Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Epoch
0
5
10
15
20

RESULTS

- Best
- Model 2

  - training accuracy: 0.62
  - Test loss: 1.98
  - Test accuracy: 0.49

- Model 1

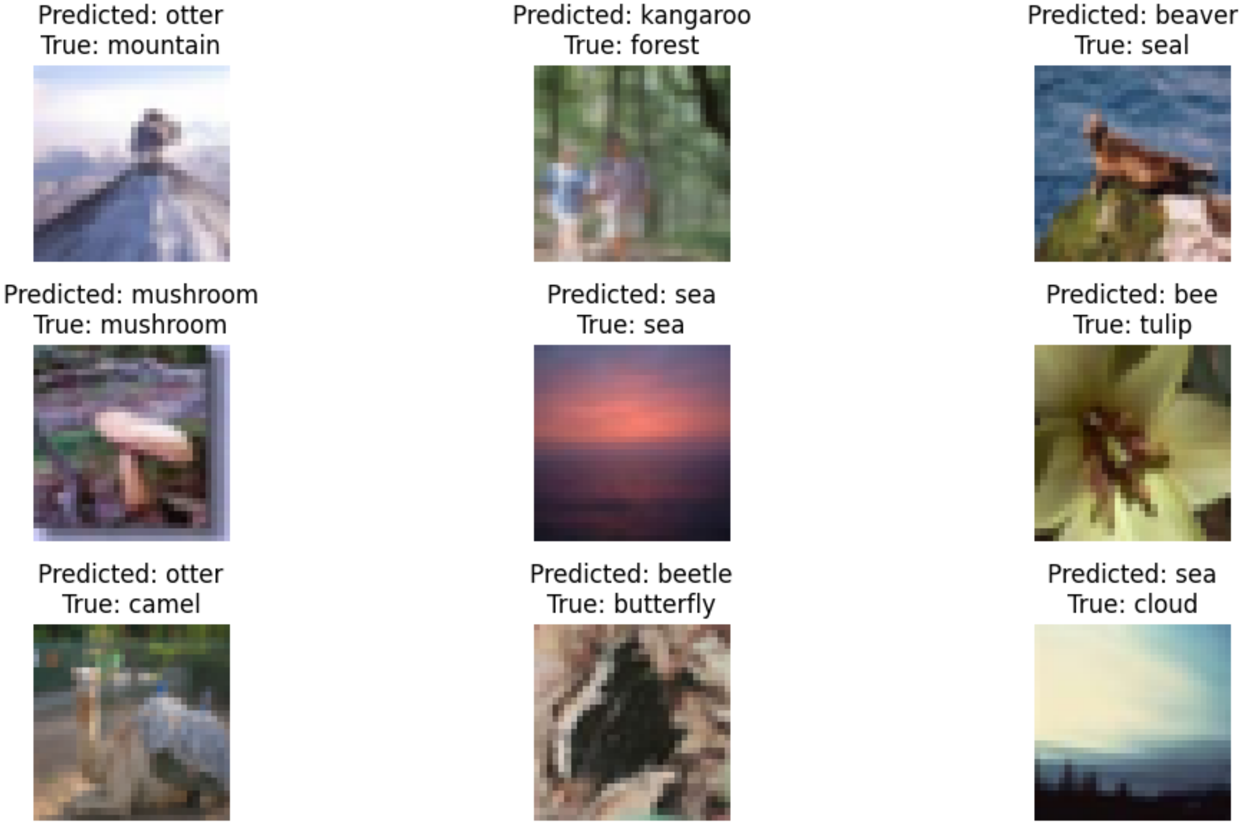
  - training accuracy: 0.0084
  - Test loss: 4.61
  - Test accuracy: 0.01

- Model 3

  - training accuracy: 0.54
  - Test loss: 2.94
  - Test accuracy: 0.37

IMAGE LABELING

- display of test images, only 2 correctly labeled



Predicted: otter
True: mountain

Predicted: kangaroo
True: forest

Predicted: beaver
True: seal

Predicted: mushroom
True: mushroom

Predicted: sea
True: sea

Predicted: bee
True: tulip

Predicted: otter
True: camel

Predicted: beetle
True: butterfly

Predicted: sea
True: cloud

IMAGE LABELING

- Selection of correctly labeled images

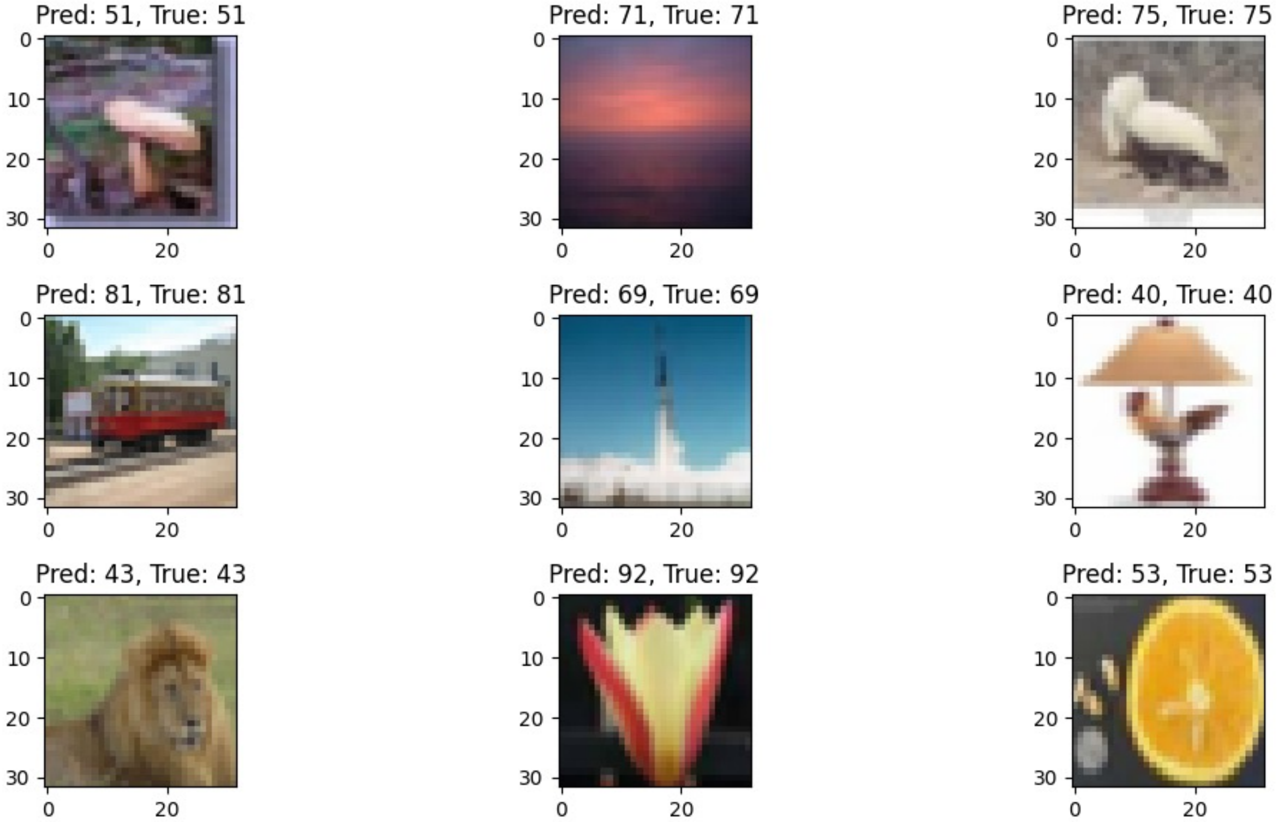


IMAGE LABELING

- Selection of incorrectly labeled images

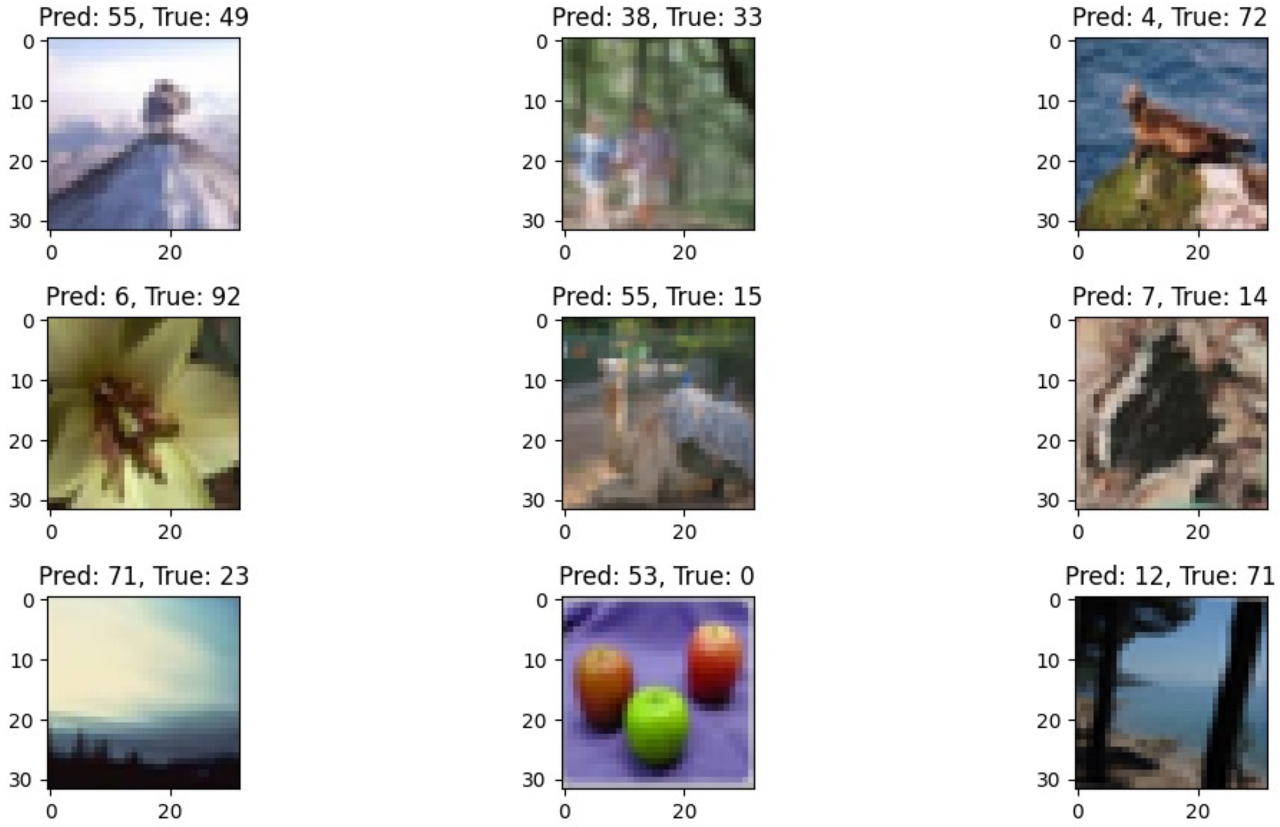
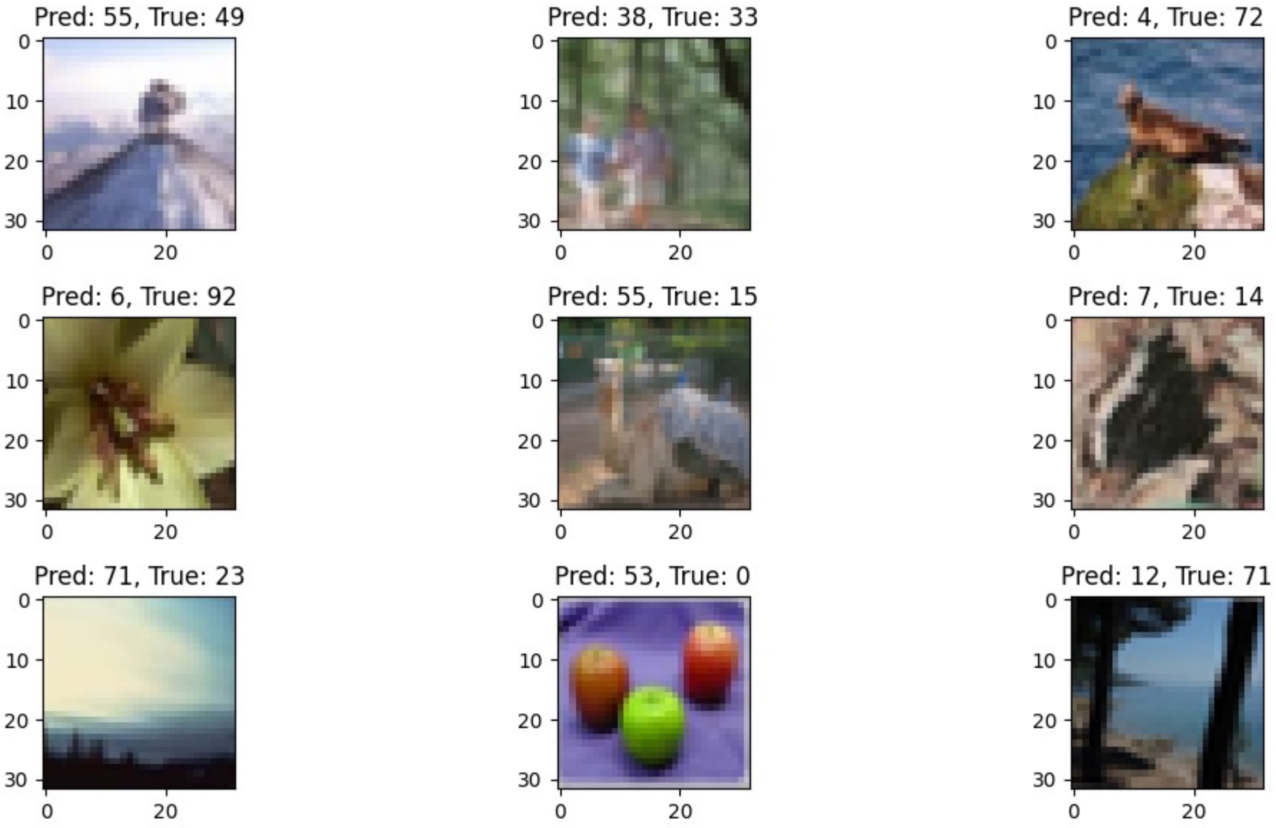


IMAGE LABELING

- Selection of incorrectly labeled images



Pred: 55, True: 49

Pred: 38, True: 33

Pred: 4, True: 72

Pred: 6, True: 92

Pred: 55, True: 15

Pred: 7, True: 14

Pred: 71, True: 23

Pred: 53, True: 0

Pred: 12, True: 71

CONCLUSION

- Model 2 had the best test accuracy compared to the other models with an accuracy of 0.49.
- Model 2 with epoch 20 and batch size 164 was stopped early to prevent overfitting which ended up with a training accuracy of 0.62. If the early stop was not there, the training for epoch 20 was up to 0.82. I could improve the training and testing accuracy using another optimizer or architecture.