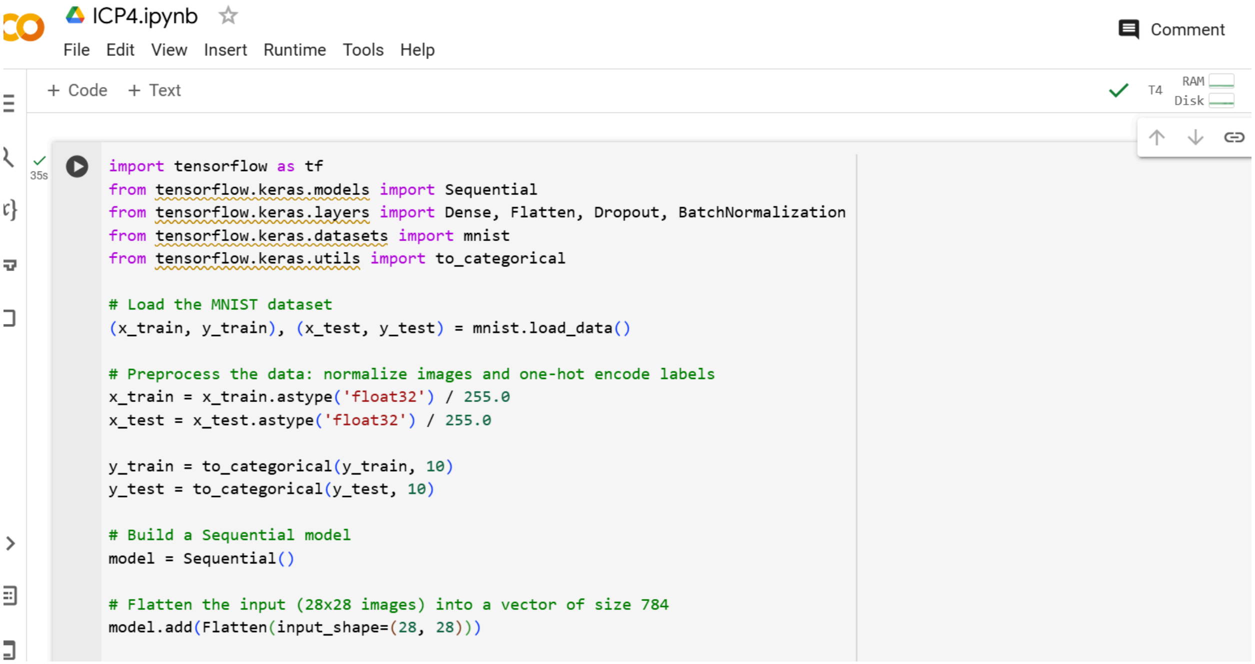
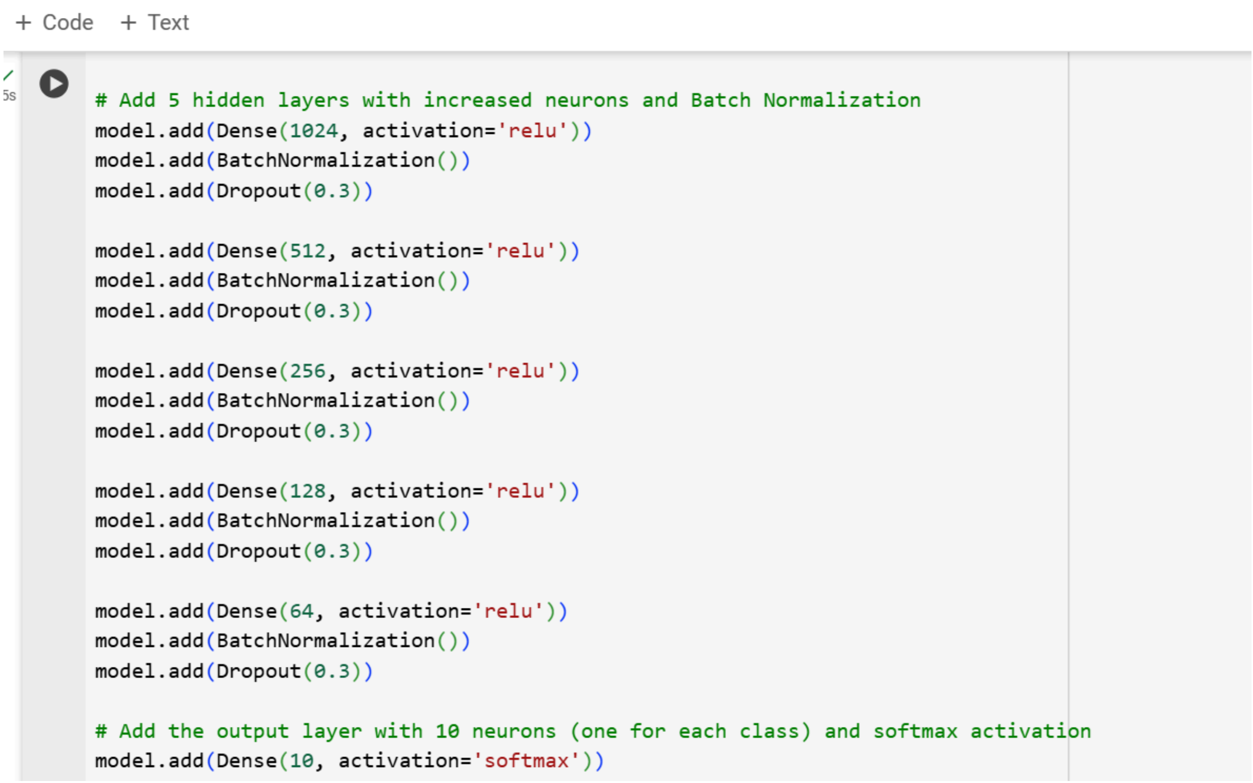
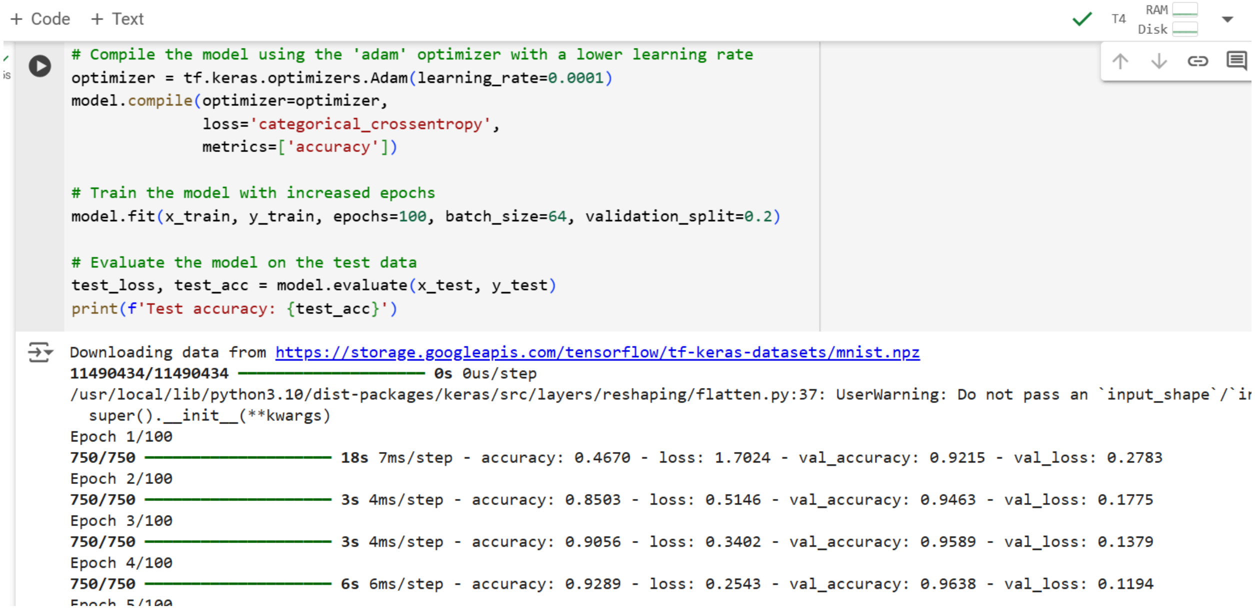
ICP4-REPORT



```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Dropout, BatchNormalization
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical

# Load the MNIST dataset
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Preprocess the data: normalize images and one-hot encode labels
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)

# Build a Sequential model
model = Sequential()

# Flatten the input (28x28 images) into a vector of size 784
model.add(Flatten(input_shape=(28, 28)))
```

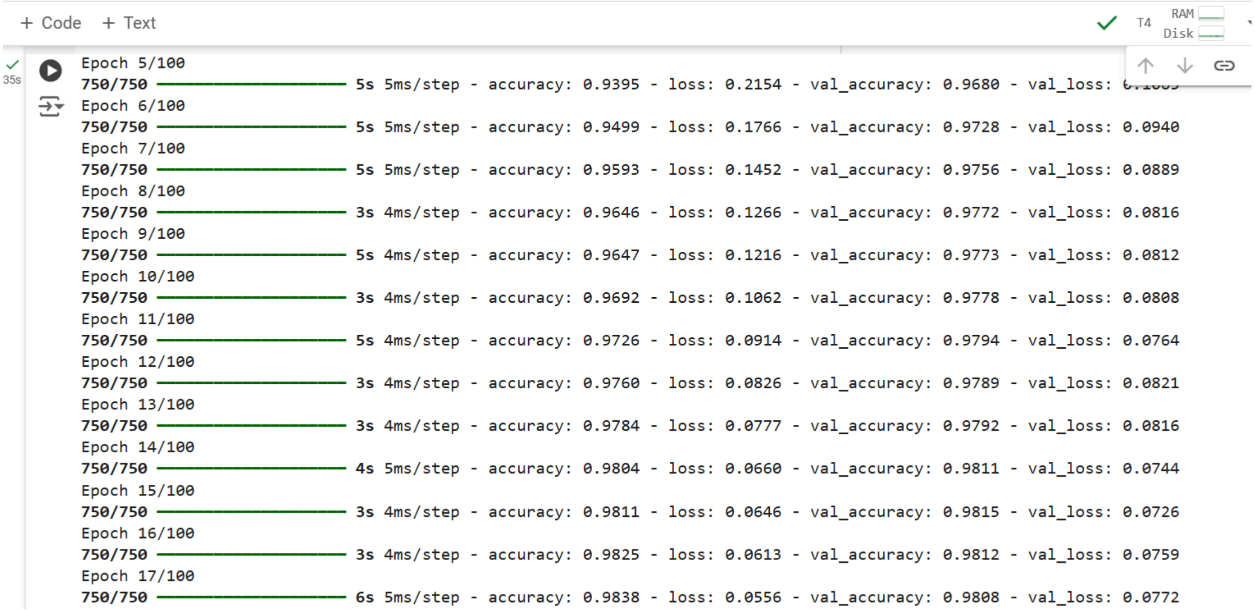


```
# Add 5 hidden layers with increased neurons and Batch Normalization
model.add(Dense(1024, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.3))

model.add(Dense(512, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.3))

model.add(Dense(256, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.3))

model.add(Dense(128, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.3))

model.add(Dense(64, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.3))

# Add the output layer with 10 neurons (one for each class) and softmax activation
model.add(Dense(10, activation='softmax'))
```

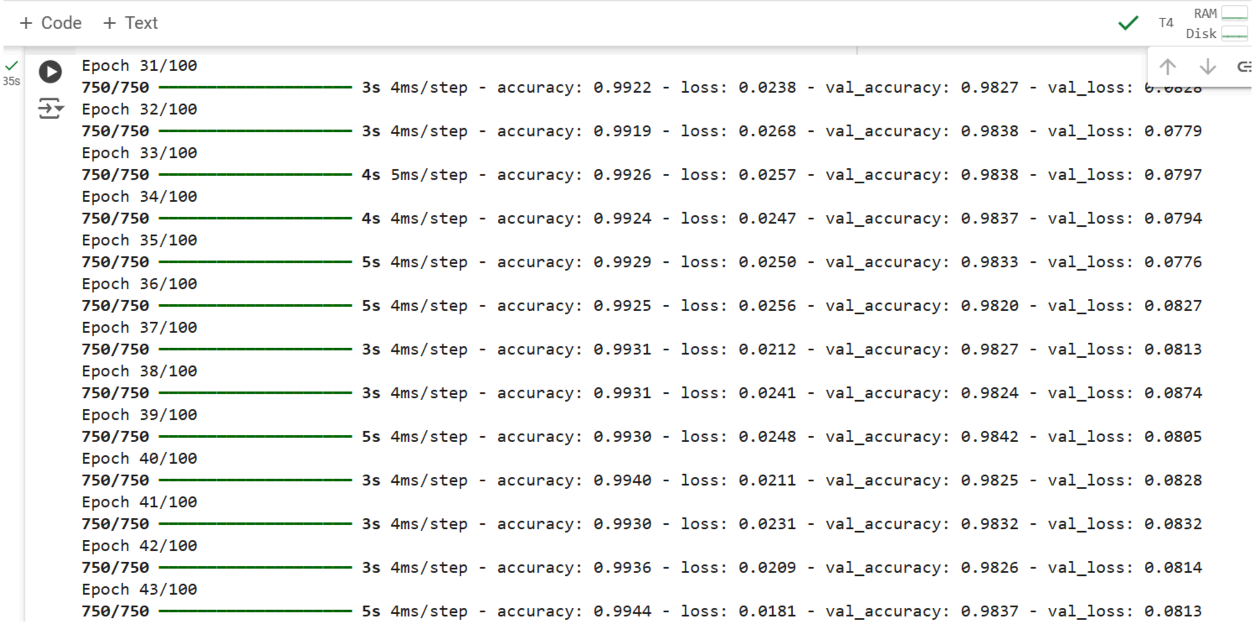


```python
# Compile the model using the 'adam' optimizer with a lower learning rate
optimizer = tf.keras.optimizers.Adam(learning_rate=0.0001)
model.compile(optimizer=optimizer,
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train the model with increased epochs
model.fit(x_train, y_train, epochs=100, batch_size=64, validation_split=0.2)

# Evaluate the model on the test data
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f'Test accuracy: {test_acc}')
```

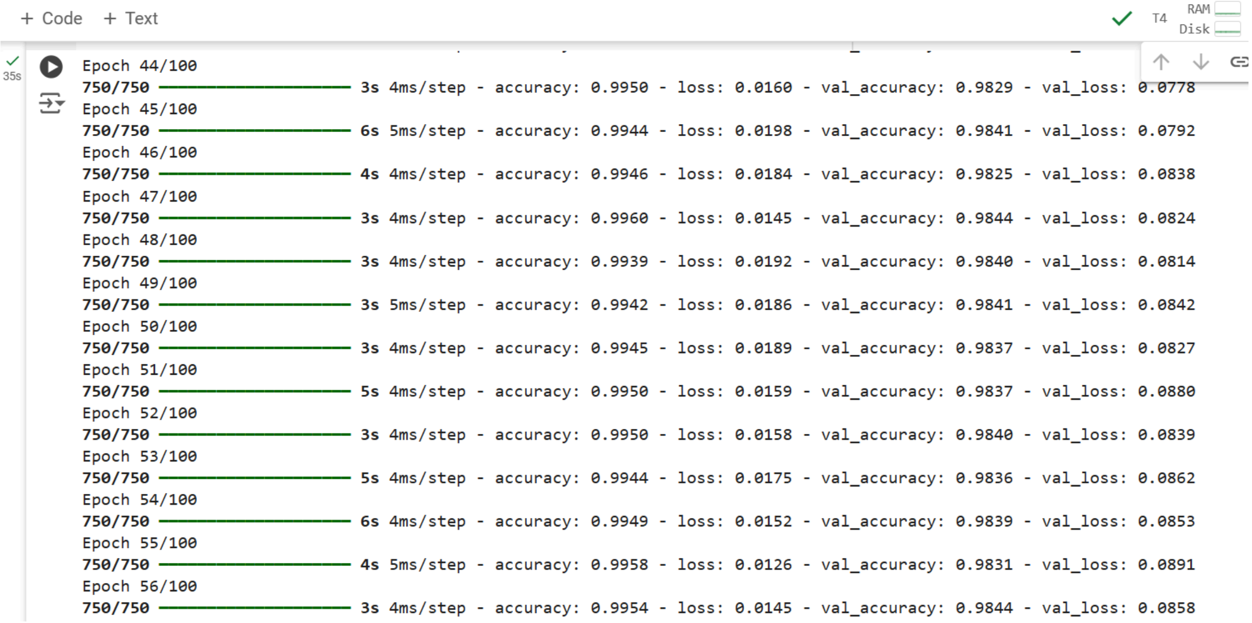
```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
/usr/local/lib/python3.10/dist-packages/keras/src/layers/reshaping/flatten.py:37: UserWarning: Do not pass an `input_shape`/`ir
  super().__init__(**kwargs)
Epoch 1/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 18s 7ms/step - accuracy: 0.4670 - loss: 1.7024 - val_accuracy: 0.9215 - val_loss: 0.2783
Epoch 2/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.8503 - loss: 0.5146 - val_accuracy: 0.9463 - val_loss: 0.1775
Epoch 3/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9056 - loss: 0.3402 - val_accuracy: 0.9589 - val_loss: 0.1379
Epoch 4/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 6s 6ms/step - accuracy: 0.9289 - loss: 0.2543 - val_accuracy: 0.9638 - val_loss: 0.1194
Epoch 5/100
```



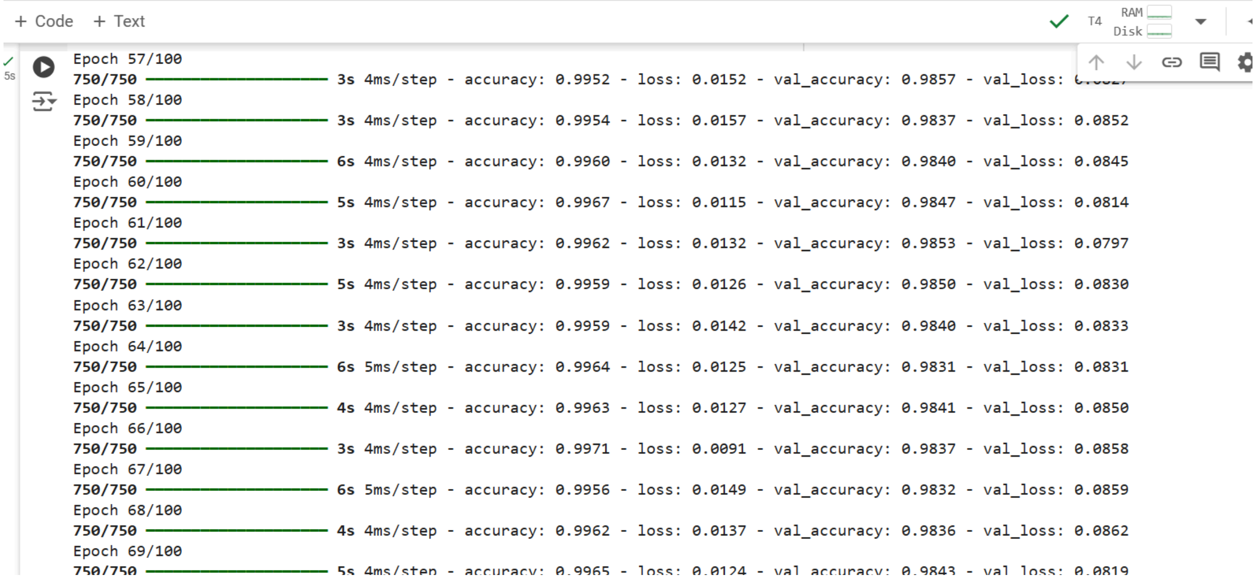
```
Epoch 5/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 5ms/step - accuracy: 0.9395 - loss: 0.2154 - val_accuracy: 0.9680 - val_loss: 0.1005
Epoch 6/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 5ms/step - accuracy: 0.9499 - loss: 0.1766 - val_accuracy: 0.9728 - val_loss: 0.0940
Epoch 7/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 5ms/step - accuracy: 0.9593 - loss: 0.1452 - val_accuracy: 0.9756 - val_loss: 0.0889
Epoch 8/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9646 - loss: 0.1266 - val_accuracy: 0.9772 - val_loss: 0.0816
Epoch 9/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9647 - loss: 0.1216 - val_accuracy: 0.9773 - val_loss: 0.0812
Epoch 10/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9692 - loss: 0.1062 - val_accuracy: 0.9778 - val_loss: 0.0808
Epoch 11/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9726 - loss: 0.0914 - val_accuracy: 0.9794 - val_loss: 0.0764
Epoch 12/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9760 - loss: 0.0826 - val_accuracy: 0.9789 - val_loss: 0.0821
Epoch 13/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9784 - loss: 0.0777 - val_accuracy: 0.9792 - val_loss: 0.0816
Epoch 14/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 5ms/step - accuracy: 0.9804 - loss: 0.0660 - val_accuracy: 0.9811 - val_loss: 0.0744
Epoch 15/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9811 - loss: 0.0646 - val_accuracy: 0.9815 - val_loss: 0.0726
Epoch 16/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9825 - loss: 0.0613 - val_accuracy: 0.9812 - val_loss: 0.0759
Epoch 17/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 6s 5ms/step - accuracy: 0.9838 - loss: 0.0556 - val_accuracy: 0.9808 - val_loss: 0.0772
```



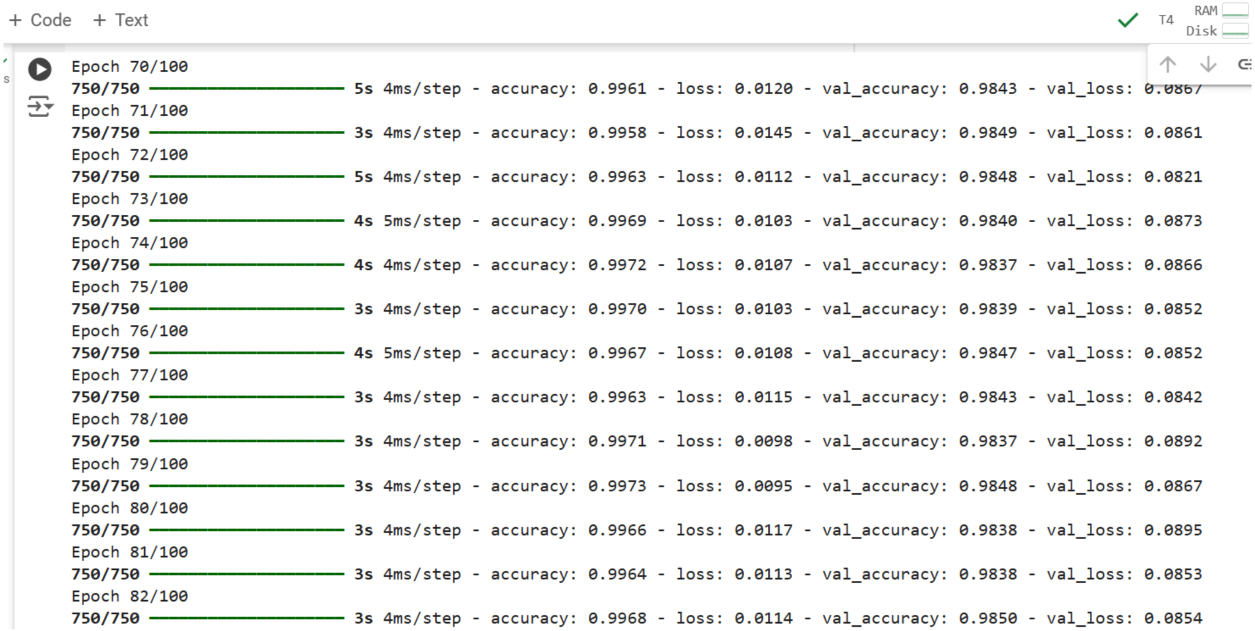
```
Epoch 31/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9922 - loss: 0.0238 - val_accuracy: 0.9827 - val_loss: 0.0828
Epoch 32/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9919 - loss: 0.0268 - val_accuracy: 0.9838 - val_loss: 0.0779
Epoch 33/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 5ms/step - accuracy: 0.9926 - loss: 0.0257 - val_accuracy: 0.9838 - val_loss: 0.0797
Epoch 34/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 4ms/step - accuracy: 0.9924 - loss: 0.0247 - val_accuracy: 0.9837 - val_loss: 0.0794
Epoch 35/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9929 - loss: 0.0250 - val_accuracy: 0.9833 - val_loss: 0.0776
Epoch 36/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9925 - loss: 0.0256 - val_accuracy: 0.9820 - val_loss: 0.0827
Epoch 37/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9931 - loss: 0.0212 - val_accuracy: 0.9827 - val_loss: 0.0813
Epoch 38/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9931 - loss: 0.0241 - val_accuracy: 0.9824 - val_loss: 0.0874
Epoch 39/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9930 - loss: 0.0248 - val_accuracy: 0.9842 - val_loss: 0.0805
Epoch 40/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9940 - loss: 0.0211 - val_accuracy: 0.9825 - val_loss: 0.0828
Epoch 41/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9930 - loss: 0.0231 - val_accuracy: 0.9832 - val_loss: 0.0832
Epoch 42/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9936 - loss: 0.0209 - val_accuracy: 0.9826 - val_loss: 0.0814
Epoch 43/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9944 - loss: 0.0181 - val_accuracy: 0.9837 - val_loss: 0.0813
```



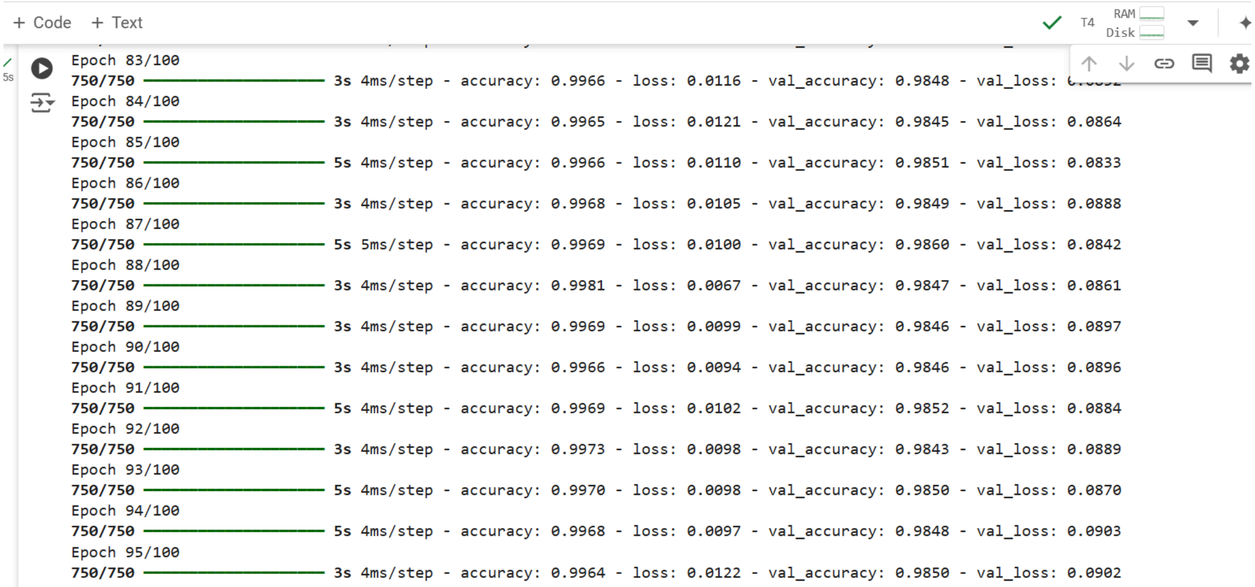
```
Epoch 44/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9950 - loss: 0.0160 - val_accuracy: 0.9829 - val_loss: 0.0778
Epoch 45/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 6s 5ms/step - accuracy: 0.9944 - loss: 0.0198 - val_accuracy: 0.9841 - val_loss: 0.0792
Epoch 46/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 4ms/step - accuracy: 0.9946 - loss: 0.0184 - val_accuracy: 0.9825 - val_loss: 0.0838
Epoch 47/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9960 - loss: 0.0145 - val_accuracy: 0.9844 - val_loss: 0.0824
Epoch 48/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9939 - loss: 0.0192 - val_accuracy: 0.9840 - val_loss: 0.0814
Epoch 49/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 5ms/step - accuracy: 0.9942 - loss: 0.0186 - val_accuracy: 0.9841 - val_loss: 0.0842
Epoch 50/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9945 - loss: 0.0189 - val_accuracy: 0.9837 - val_loss: 0.0827
Epoch 51/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9950 - loss: 0.0159 - val_accuracy: 0.9837 - val_loss: 0.0880
Epoch 52/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9950 - loss: 0.0158 - val_accuracy: 0.9840 - val_loss: 0.0839
Epoch 53/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9944 - loss: 0.0175 - val_accuracy: 0.9836 - val_loss: 0.0862
Epoch 54/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 6s 4ms/step - accuracy: 0.9949 - loss: 0.0152 - val_accuracy: 0.9839 - val_loss: 0.0853
Epoch 55/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 5ms/step - accuracy: 0.9958 - loss: 0.0126 - val_accuracy: 0.9831 - val_loss: 0.0891
Epoch 56/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9954 - loss: 0.0145 - val_accuracy: 0.9844 - val_loss: 0.0858
```



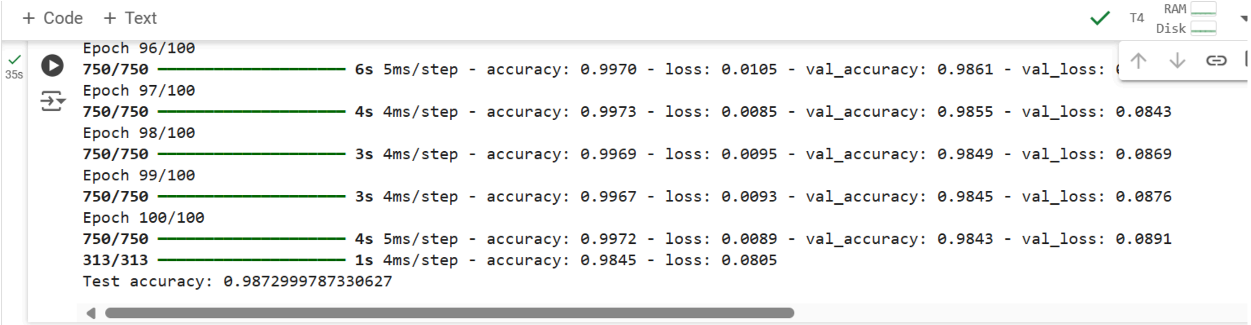
```
Epoch 57/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9952 - loss: 0.0152 - val_accuracy: 0.9857 - val_loss: 0.0827
Epoch 58/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9954 - loss: 0.0157 - val_accuracy: 0.9837 - val_loss: 0.0852
Epoch 59/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 6s 4ms/step - accuracy: 0.9960 - loss: 0.0132 - val_accuracy: 0.9840 - val_loss: 0.0845
Epoch 60/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9967 - loss: 0.0115 - val_accuracy: 0.9847 - val_loss: 0.0814
Epoch 61/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9962 - loss: 0.0132 - val_accuracy: 0.9853 - val_loss: 0.0797
Epoch 62/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9959 - loss: 0.0126 - val_accuracy: 0.9850 - val_loss: 0.0830
Epoch 63/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9959 - loss: 0.0142 - val_accuracy: 0.9840 - val_loss: 0.0833
Epoch 64/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 6s 5ms/step - accuracy: 0.9964 - loss: 0.0125 - val_accuracy: 0.9831 - val_loss: 0.0831
Epoch 65/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 4ms/step - accuracy: 0.9963 - loss: 0.0127 - val_accuracy: 0.9841 - val_loss: 0.0850
Epoch 66/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9971 - loss: 0.0091 - val_accuracy: 0.9837 - val_loss: 0.0858
Epoch 67/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 6s 5ms/step - accuracy: 0.9956 - loss: 0.0149 - val_accuracy: 0.9832 - val_loss: 0.0859
Epoch 68/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 4ms/step - accuracy: 0.9962 - loss: 0.0137 - val_accuracy: 0.9836 - val_loss: 0.0862
Epoch 69/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9965 - loss: 0.0124 - val_accuracy: 0.9843 - val_loss: 0.0819
```



```
Epoch 70/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9961 - loss: 0.0120 - val_accuracy: 0.9843 - val_loss: 0.0867
Epoch 71/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9958 - loss: 0.0145 - val_accuracy: 0.9849 - val_loss: 0.0861
Epoch 72/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9963 - loss: 0.0112 - val_accuracy: 0.9848 - val_loss: 0.0821
Epoch 73/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 5ms/step - accuracy: 0.9969 - loss: 0.0103 - val_accuracy: 0.9840 - val_loss: 0.0873
Epoch 74/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 4ms/step - accuracy: 0.9972 - loss: 0.0107 - val_accuracy: 0.9837 - val_loss: 0.0866
Epoch 75/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9970 - loss: 0.0103 - val_accuracy: 0.9839 - val_loss: 0.0852
Epoch 76/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 5ms/step - accuracy: 0.9967 - loss: 0.0108 - val_accuracy: 0.9847 - val_loss: 0.0852
Epoch 77/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9963 - loss: 0.0115 - val_accuracy: 0.9843 - val_loss: 0.0842
Epoch 78/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9971 - loss: 0.0098 - val_accuracy: 0.9837 - val_loss: 0.0892
Epoch 79/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9973 - loss: 0.0095 - val_accuracy: 0.9848 - val_loss: 0.0867
Epoch 80/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9966 - loss: 0.0117 - val_accuracy: 0.9838 - val_loss: 0.0895
Epoch 81/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9964 - loss: 0.0113 - val_accuracy: 0.9838 - val_loss: 0.0853
Epoch 82/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9968 - loss: 0.0114 - val_accuracy: 0.9850 - val_loss: 0.0854
```



```
Epoch 83/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9966 - loss: 0.0116 - val_accuracy: 0.9848 - val_loss:
Epoch 84/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9965 - loss: 0.0121 - val_accuracy: 0.9845 - val_loss: 0.0864
Epoch 85/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9966 - loss: 0.0110 - val_accuracy: 0.9851 - val_loss: 0.0833
Epoch 86/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9968 - loss: 0.0105 - val_accuracy: 0.9849 - val_loss: 0.0888
Epoch 87/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 5ms/step - accuracy: 0.9969 - loss: 0.0100 - val_accuracy: 0.9860 - val_loss: 0.0842
Epoch 88/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9981 - loss: 0.0067 - val_accuracy: 0.9847 - val_loss: 0.0861
Epoch 89/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9969 - loss: 0.0099 - val_accuracy: 0.9846 - val_loss: 0.0897
Epoch 90/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9966 - loss: 0.0094 - val_accuracy: 0.9846 - val_loss: 0.0896
Epoch 91/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9969 - loss: 0.0102 - val_accuracy: 0.9852 - val_loss: 0.0884
Epoch 92/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9973 - loss: 0.0098 - val_accuracy: 0.9843 - val_loss: 0.0889
Epoch 93/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9970 - loss: 0.0098 - val_accuracy: 0.9850 - val_loss: 0.0870
Epoch 94/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 5s 4ms/step - accuracy: 0.9968 - loss: 0.0097 - val_accuracy: 0.9848 - val_loss: 0.0903
Epoch 95/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9964 - loss: 0.0122 - val_accuracy: 0.9850 - val_loss: 0.0902
```



```
Epoch 96/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 6s 5ms/step - accuracy: 0.9970 - loss: 0.0105 - val_accuracy: 0.9861 - val_loss:
Epoch 97/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 4ms/step - accuracy: 0.9973 - loss: 0.0085 - val_accuracy: 0.9855 - val_loss: 0.0843
Epoch 98/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9969 - loss: 0.0095 - val_accuracy: 0.9849 - val_loss: 0.0869
Epoch 99/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9967 - loss: 0.0093 - val_accuracy: 0.9845 - val_loss: 0.0876
Epoch 100/100
750/750 ━━━━━━━━━━━━━━━━━━━━ 4s 5ms/step - accuracy: 0.9972 - loss: 0.0089 - val_accuracy: 0.9843 - val_loss: 0.0891
313/313 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - accuracy: 0.9845 - loss: 0.0805
Test accuracy: 0.9872999787330627
```

GITHUB REPO LINK:- https://github.com/akshaykumarpathem/bda.git

YOUTUBE LINK:-https://youtu.be/lHW6thslX14