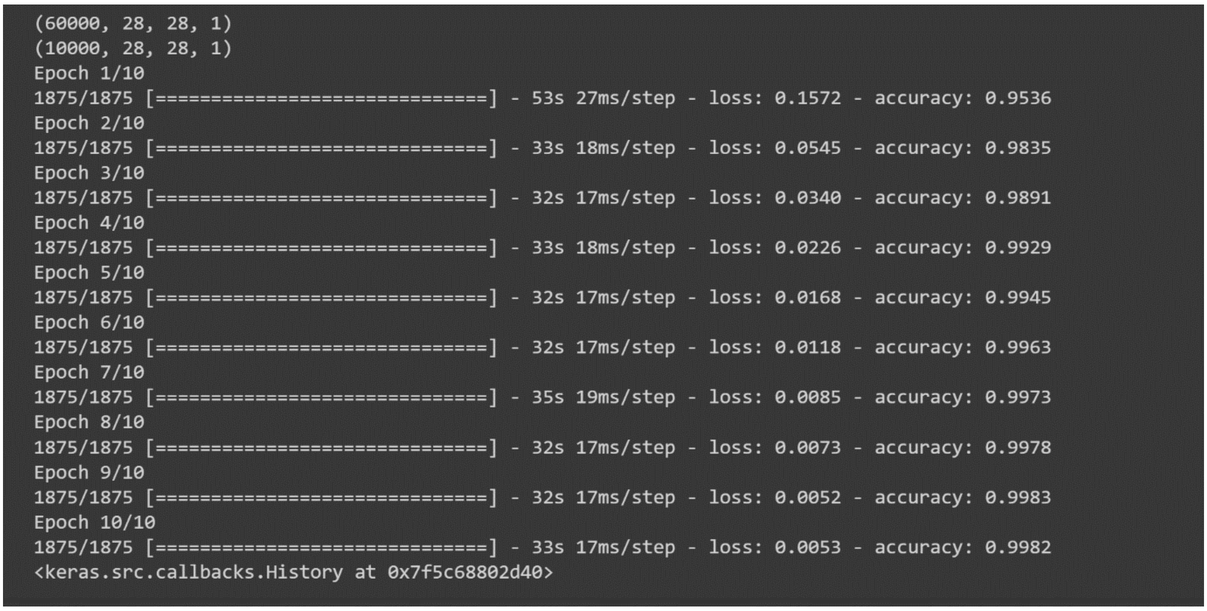
PROGRAM [15]:

```
# Importing the required libraries
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPool2D, Flatten, Dense
from tensorflow.keras.layers import Dropout
# Loading data
(X_train, y_train), (X_test, y_test) = mnist.load_data()
# Reshaping data
X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], X_train.shape[2], 1))
X_test = X_test.reshape((X_test.shape[0], X_test.shape[1], X_test.shape[2], 1))
# Checking the shape after reshaping
print(X_train.shape)
print(X_test.shape)
# Normalizing the pixel values
X_train = X_train / 255
X_test = X_test / 255
# Defining model
model = Sequential()
# Adding convolution layer
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
# Adding pooling layer
model.add(MaxPool2D(2, 2))
# Adding fully connected layer
model.add(Flatten())
model.add(Dense(100, activation='relu'))
```

OUTPUT [15]:



```
(60000, 28, 28, 1)
(10000, 28, 28, 1)
Epoch 1/10
1875/1875 [==============================] - 53s 27ms/step - loss: 0.1572 - accuracy: 0.9536
Epoch 2/10
1875/1875 [==============================] - 33s 18ms/step - loss: 0.0545 - accuracy: 0.9835
Epoch 3/10
1875/1875 [==============================] - 32s 17ms/step - loss: 0.0340 - accuracy: 0.9891
Epoch 4/10
1875/1875 [==============================] - 33s 18ms/step - loss: 0.0226 - accuracy: 0.9929
Epoch 5/10
1875/1875 [==============================] - 32s 17ms/step - loss: 0.0168 - accuracy: 0.9945
Epoch 6/10
1875/1875 [==============================] - 32s 17ms/step - loss: 0.0118 - accuracy: 0.9963
Epoch 7/10
1875/1875 [==============================] - 35s 19ms/step - loss: 0.0085 - accuracy: 0.9973
Epoch 8/10
1875/1875 [==============================] - 32s 17ms/step - loss: 0.0073 - accuracy: 0.9978
Epoch 9/10
1875/1875 [==============================] - 32s 17ms/step - loss: 0.0052 - accuracy: 0.9983
Epoch 10/10
1875/1875 [==============================] - 33s 17ms/step - loss: 0.0053 - accuracy: 0.9982
<keras.src.callbacks.History at 0x7f5c68802d40>
```

```
 Adding output layer
model.add(Dense(10, activation='softmax'))
# Compiling the model
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
# Fitting the model
model.fit(X_train, y_train, epochs=10)
```