BHARAT INTERN

Task 2

Handwritten Number Recognition using MNIST dataset

```
#Import necessary libraries
import tensorflow
from tensorflow import keras
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense,Flatten
```

```
# Load MNIST dataset
(X_train,y_train),(X_test,y_test) = keras.datasets.mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 0s 0us/step
```

```
# Display the shape of the test dataset
X_test.shape
```

```
(10000, 28, 28)
```

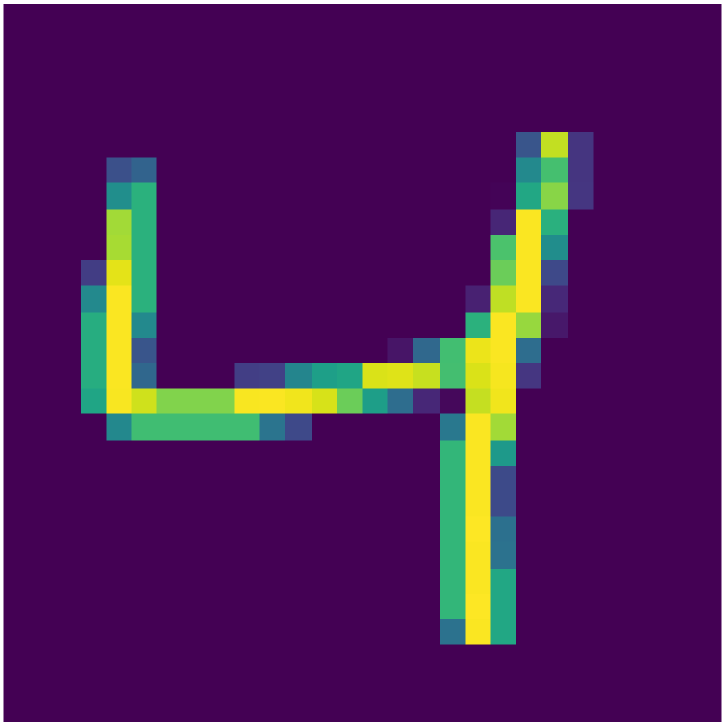
```
# Display the label in the training data set
y_train
```

```
array([5, 0, 4, ..., 5, 6, 8], dtype=uint8)
```

```
# Import matplotlib for visualization
import matplotlib.pyplot as plt
```

```
# Display an image from the training set
plt.imshow(X_train[2])
plt.axis('off')
```

```
(-0.5, 27.5, 27.5, -0.5)
```



```
#Normalize the pixel values to a range between 0 and 1
X_train = X_train/255
X_test = X_test/255

# Display normalized pixel values of the first image
X_train[0]
```

```
array([[0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ],
       [0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        , 0.        , 0.        ,
```

```
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.01176471, 0.07058824, 0.07058824,
         0.07058824, 0.49411765, 0.53333333, 0.68627451, 0.10196078,
         0.65098039, 1.        , 0.96862745, 0.49803922, 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.11764706, 0.14117647,
         0.36862745, 0.60392157, 0.66666667, 0.99215686, 0.99215686,
         0.99215686, 0.99215686, 0.99215686, 0.88235294, 0.6745098 ,
         0.99215686, 0.94901961, 0.76470588, 0.25098039, 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.19215686, 0.93333333, 0.99215686,
         0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.99215686,
         0.99215686, 0.99215686, 0.98431373, 0.36470588, 0.32156863,
         0.32156863, 0.21960784, 0.15294118, 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.07058824, 0.85882353, 0.99215686,
         0.99215686, 0.99215686, 0.99215686, 0.99215686, 0.77647059,
         0.71372549, 0.96862745, 0.94509804, 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.31372549, 0.61176471,
         0.41960784, 0.99215686, 0.99215686, 0.80392157, 0.04313725,
         0.        , 0.16862745, 0.60392157, 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.05490196,
         0.00392157, 0.60392157, 0.99215686, 0.35294118, 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.54509804, 0.99215686, 0.74509804, 0.00784314,
```

```
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.04313725, 0.74509804, 0.99215686, 0.2745098 ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.1372549 , 0.94509804, 0.88235294,
         0.62745098, 0.42352941, 0.00392157, 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.31764706, 0.94117647,
         0.99215686, 0.99215686, 0.46666667, 0.09803922, 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.17647059,
         0.72941176, 0.99215686, 0.99215686, 0.58823529, 0.10588235,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.0627451 , 0.36470588, 0.98823529, 0.99215686, 0.73333333,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.97647059, 0.99215686, 0.97647059,
         0.25098039, 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.18039216,
         0.50980392, 0.71764706, 0.99215686, 0.99215686, 0.81176471,
         0.00784314, 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.15294118, 0.58039216, 0.89803922,
         0.99215686, 0.99215686, 0.99215686, 0.98039216, 0.71372549,
```

```
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.09411765, 0.44705882, 0.86666667, 0.99215686, 0.99215686,
         0.99215686, 0.99215686, 0.78823529, 0.30588235, 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.09019608, 0.25882353,
         0.83529412, 0.99215686, 0.99215686, 0.99215686, 0.99215686,
         0.77647059, 0.31764706, 0.00784314, 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.07058824, 0.67058824, 0.85882353, 0.99215686,
         0.99215686, 0.99215686, 0.99215686, 0.76470588, 0.31372549,
         0.03529412, 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.21568627,
         0.6745098 , 0.88627451, 0.99215686, 0.99215686, 0.99215686,
         0.99215686, 0.95686275, 0.52156863, 0.04313725, 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.53333333,
         0.99215686, 0.99215686, 0.99215686, 0.83137255, 0.52941176,
         0.51764706, 0.0627451 , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        ],
        [0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
         0.        , 0.        , 0.        , 0.        , 0.        ,
```

```
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.        , 0.        ]])

# Create a Sequential model
model = Sequential()

# Add layers to the model
model.add(Flatten(input_shape=(28,28)))
model.add(Dense(128,activation='relu'))
model.add(Dense(32,activation='relu'))
model.add(Dense(10,activation='softmax'))

# Display model summary
model.summary()

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten (Flatten)           (None, 784)               0

 dense (Dense)               (None, 128)               100480

 dense_1 (Dense)             (None, 32)                4128

 dense_2 (Dense)             (None, 10)                330

=================================================================
Total params: 104938 (409.91 KB)
Trainable params: 104938 (409.91 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________

# Compile the model
model.compile(loss='sparse_categorical_crossentropy',optimizer='Adam',
metrics=['accuracy'])

# Train the model and store training history
history = model.fit(X_train,y_train,epochs=25,validation_split=0.2)

Epoch 1/25
1500/1500 [==============================] - 10s 6ms/step - loss:
0.2842 - accuracy: 0.9162 - val_loss: 0.1490 - val_accuracy: 0.9556
Epoch 2/25
1500/1500 [==============================] - 7s 5ms/step - loss:
0.1217 - accuracy: 0.9630 - val_loss: 0.1050 - val_accuracy: 0.9674
Epoch 3/25
1500/1500 [==============================] - 8s 6ms/step - loss:
0.0826 - accuracy: 0.9748 - val_loss: 0.1003 - val_accuracy: 0.9675
Epoch 4/25
1500/1500 [==============================] - 9s 6ms/step - loss:
```

```
0.0625 - accuracy: 0.9802 - val_loss: 0.0965 - val_accuracy: 0.9722
Epoch 5/25
1500/1500 [==============================] - 8s 6ms/step - loss:
0.0474 - accuracy: 0.9845 - val_loss: 0.1171 - val_accuracy: 0.9660
Epoch 6/25
1500/1500 [==============================] - 8s 6ms/step - loss:
0.0380 - accuracy: 0.9877 - val_loss: 0.0935 - val_accuracy: 0.9735
Epoch 7/25
1500/1500 [==============================] - 7s 5ms/step - loss:
0.0300 - accuracy: 0.9898 - val_loss: 0.0995 - val_accuracy: 0.9724
Epoch 8/25
1500/1500 [==============================] - 7s 5ms/step - loss:
0.0267 - accuracy: 0.9908 - val_loss: 0.1161 - val_accuracy: 0.9730
Epoch 9/25
1500/1500 [==============================] - 8s 6ms/step - loss:
0.0212 - accuracy: 0.9933 - val_loss: 0.1005 - val_accuracy: 0.9761
Epoch 10/25
1500/1500 [==============================] - 7s 5ms/step - loss:
0.0182 - accuracy: 0.9943 - val_loss: 0.1170 - val_accuracy: 0.9726
Epoch 11/25
1500/1500 [==============================] - 9s 6ms/step - loss:
0.0159 - accuracy: 0.9950 - val_loss: 0.1211 - val_accuracy: 0.9728
Epoch 12/25
1500/1500 [==============================] - 10s 7ms/step - loss:
0.0176 - accuracy: 0.9942 - val_loss: 0.1172 - val_accuracy: 0.9747
Epoch 13/25
1500/1500 [==============================] - 8s 6ms/step - loss:
0.0134 - accuracy: 0.9955 - val_loss: 0.1325 - val_accuracy: 0.9726
Epoch 14/25
1500/1500 [==============================] - 9s 6ms/step - loss:
0.0126 - accuracy: 0.9959 - val_loss: 0.1269 - val_accuracy: 0.9756
Epoch 15/25
1500/1500 [==============================] - 9s 6ms/step - loss:
0.0131 - accuracy: 0.9958 - val_loss: 0.1175 - val_accuracy: 0.9775
Epoch 16/25
1500/1500 [==============================] - 7s 5ms/step - loss:
0.0109 - accuracy: 0.9966 - val_loss: 0.1361 - val_accuracy: 0.9743
Epoch 17/25
1500/1500 [==============================] - 8s 5ms/step - loss:
0.0108 - accuracy: 0.9965 - val_loss: 0.1426 - val_accuracy: 0.9758
Epoch 18/25
1500/1500 [==============================] - 8s 5ms/step - loss:
0.0117 - accuracy: 0.9963 - val_loss: 0.1399 - val_accuracy: 0.9749
Epoch 19/25
1500/1500 [==============================] - 9s 6ms/step - loss:
0.0086 - accuracy: 0.9972 - val_loss: 0.1536 - val_accuracy: 0.9732
Epoch 20/25
1500/1500 [==============================] - 10s 6ms/step - loss:
0.0100 - accuracy: 0.9965 - val_loss: 0.1370 - val_accuracy: 0.9762
```

```
Epoch 21/25
1500/1500 [==============================] - 8s 5ms/step - loss:
0.0079 - accuracy: 0.9975 - val_loss: 0.1566 - val_accuracy: 0.9725
Epoch 22/25
1500/1500 [==============================] - 8s 6ms/step - loss:
0.0097 - accuracy: 0.9970 - val_loss: 0.1345 - val_accuracy: 0.9778
Epoch 23/25
1500/1500 [==============================] - 9s 6ms/step - loss:
0.0067 - accuracy: 0.9978 - val_loss: 0.1383 - val_accuracy: 0.9783
Epoch 24/25
1500/1500 [==============================] - 8s 5ms/step - loss:
0.0097 - accuracy: 0.9971 - val_loss: 0.1415 - val_accuracy: 0.9774
Epoch 25/25
1500/1500 [==============================] - 9s 6ms/step - loss:
0.0046 - accuracy: 0.9983 - val_loss: 0.1581 - val_accuracy: 0.9750
```

```
# Make predictions on the test data
y_prob = model.predict(X_test)
```

```
313/313 [==============================] - 1s 2ms/step
```

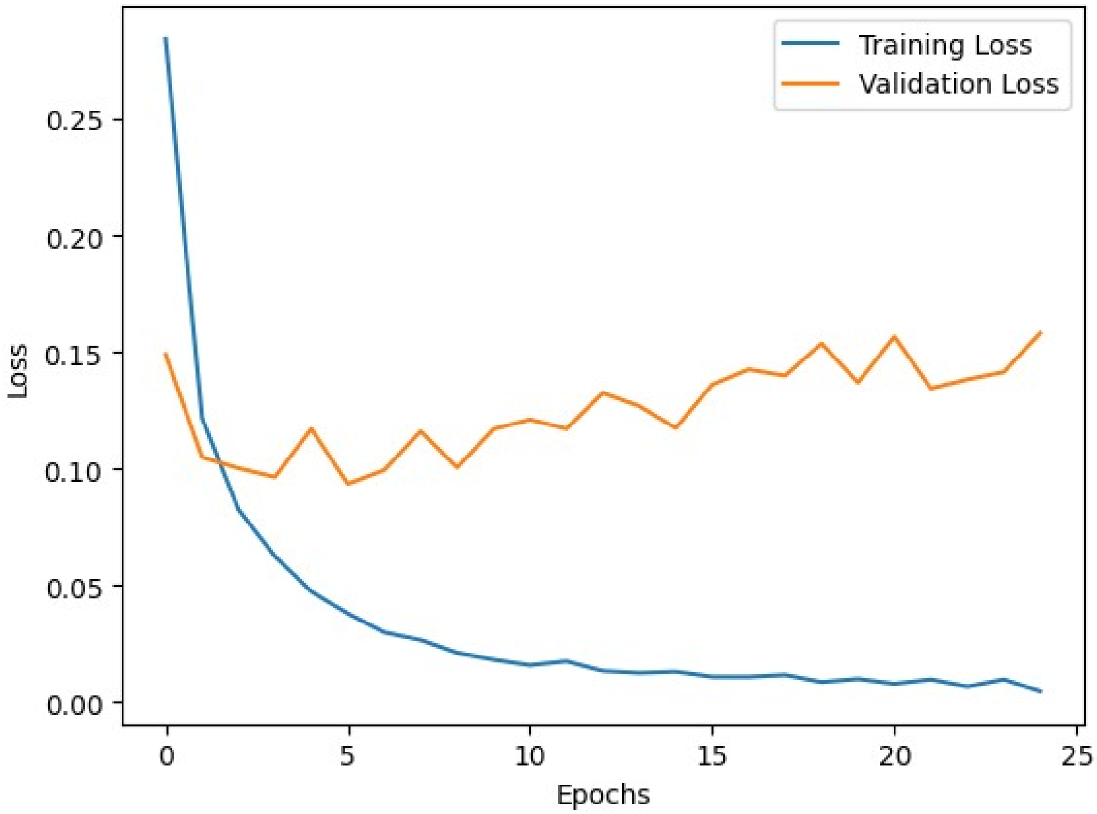
```
# Get the predicted labels by selecting the class with the highest
probability
y_pred = y_prob.argmax(axis=1)
```

```
# Import accuracy_score for calculating accuracy
from sklearn.metrics import accuracy_score
```

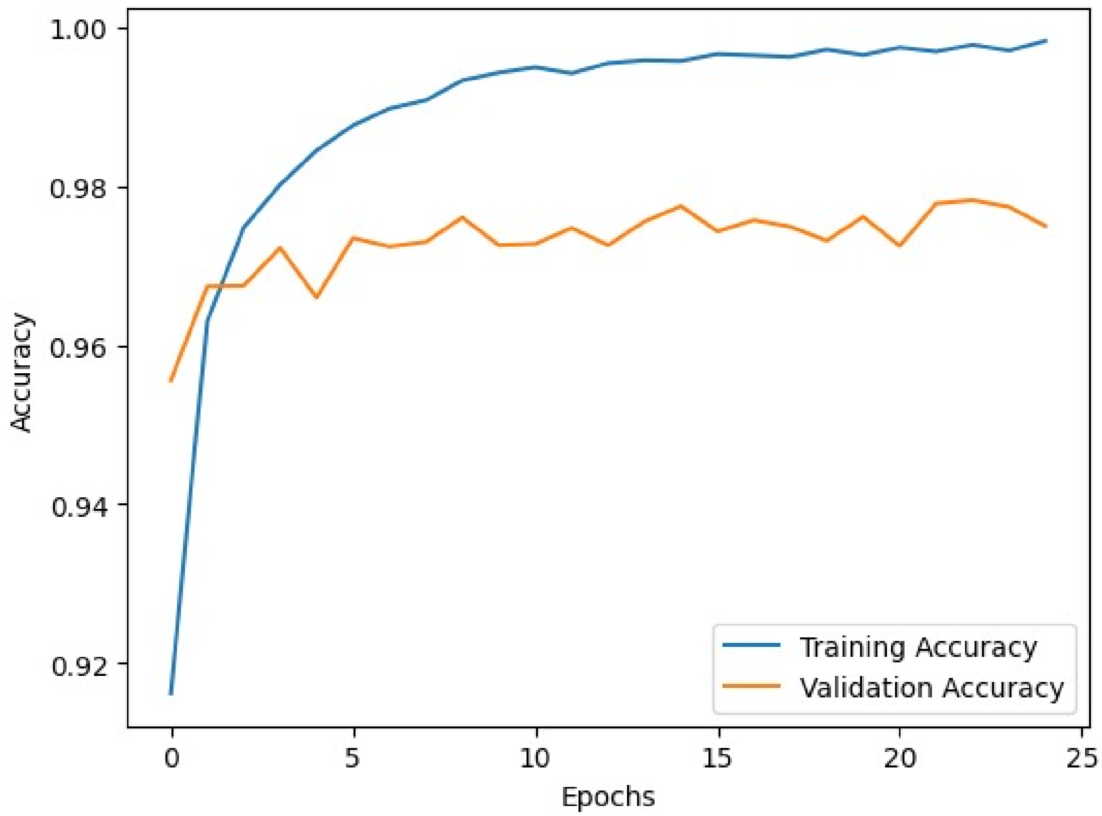
```
# Calculate and display accuracy score
accuracy_score(y_test,y_pred)
```

```
0.9773
```

```
# Plot the training and validation loss over epochs
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend(['Training Loss', 'Validation Loss'])
plt.show()
```

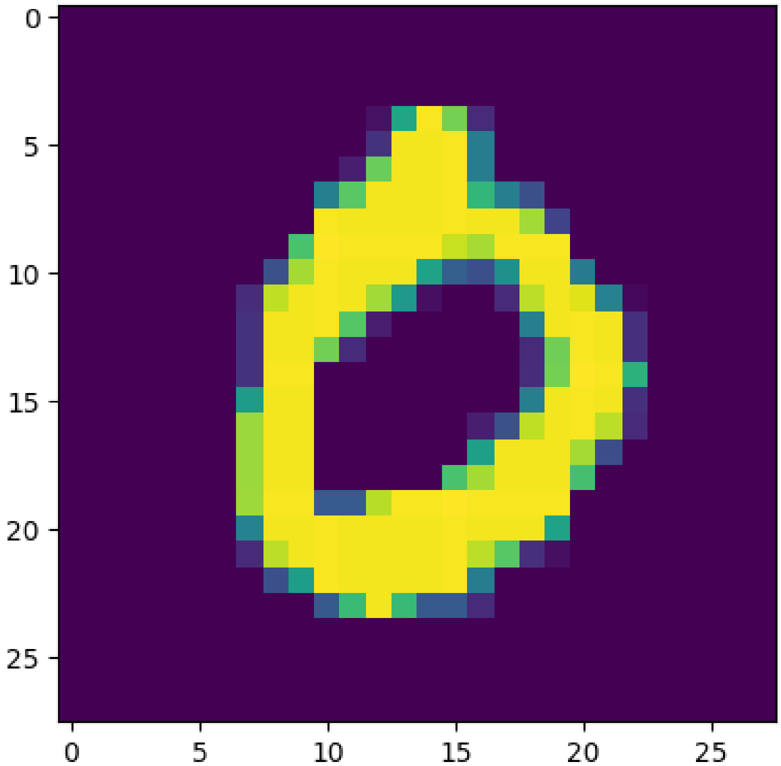



```
# Plot the training and validation accuracy over epochs
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend(['Training Accuracy', 'Validation Accuracy'])
plt.show()
```




```
# Display an image from the test set
plt.imshow(X_test[3])
```

```
<matplotlib.image.AxesImage at 0x788c6e119240>
```




```
# Make a prediction on a single test image and display the predicted
label
model.predict(X_test[3].reshape(1,28,28)).argmax(axis=1)
```

```
1/1 [==============================] - 0s 23ms/step
```

```
array([0])
```