In [ ]:
```
#  create a virtual environment
# python -m venv tfenv
# activate the virtual Environment
# tfenv\Scripts\activate
# install tensorflow under the enviromnent
# pip install tensorflow
# install matplotlib
# pip install matplotlib
# if you are under "conda" then pip will be replaced by "conda"
# run jupyter notebook
```

In [75]:
```
import tensorflow as tf
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from keras.layers import Input, Flatten, Dense
from tensorflow.keras.utils import to_categorical
```

In [77]:
```
print(tf.__version__)
```

```
2.19.0
```

In [79]:
```
import matplotlib.pyplot as plt
```

In [81]:
```
(x_train,y_train),(x_test,y_test)=mnist.load_data()
```

In [83]:
```
x_train=x_train/255.0
x_test=x_test/255.0
```

In [85]:
```
y_train
```

Out[85]:
```
array([5, 0, 4, ..., 5, 6, 8], dtype=uint8)
```

In [87]:
```
y_train=to_categorical(y_train)
y_train
```

Out[87]:
```
array([[0., 0., 0., ..., 0., 0., 0.],
       [1., 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., 1., 0.]])
```

In [89]:
```
y_test=to_categorical(y_test)
```

In [91]:
```
model = Sequential([
    Input(shape=(28, 28)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
])
```

In [93]:
```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accurac
```

In [95]:
```
model.fit(x_train,y_train,epochs=5,validation_split=0.1)
#model.fit(x_train, y_train, epochs=10, validation_data=(x_test, y_test))
```

```
Epoch 1/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - accuracy: 0.8722 - loss: 0.4493 - val_a
ccuracy: 0.9662 - val_loss: 0.1200
Epoch 2/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.9611 - loss: 0.1312 - val_a
ccuracy: 0.9728 - val_loss: 0.0993
Epoch 3/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.9751 - loss: 0.0830 - val_a
ccuracy: 0.9725 - val_loss: 0.0895
Epoch 4/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.9831 - loss: 0.0569 - val_a
ccuracy: 0.9745 - val_loss: 0.0868
Epoch 5/5
1688/1688 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - accuracy: 0.9863 - loss: 0.0437 - val_a
ccuracy: 0.9792 - val_loss: 0.0765
```

Out[95]:
```
<keras.src.callbacks.history.History at 0x193ed0d19d0>
```

In [97]:
```
test_loss,test_acc=model.evaluate(x_test,y_test)
print(f"\nTest Accuracy:{test_acc:.4f}")
```

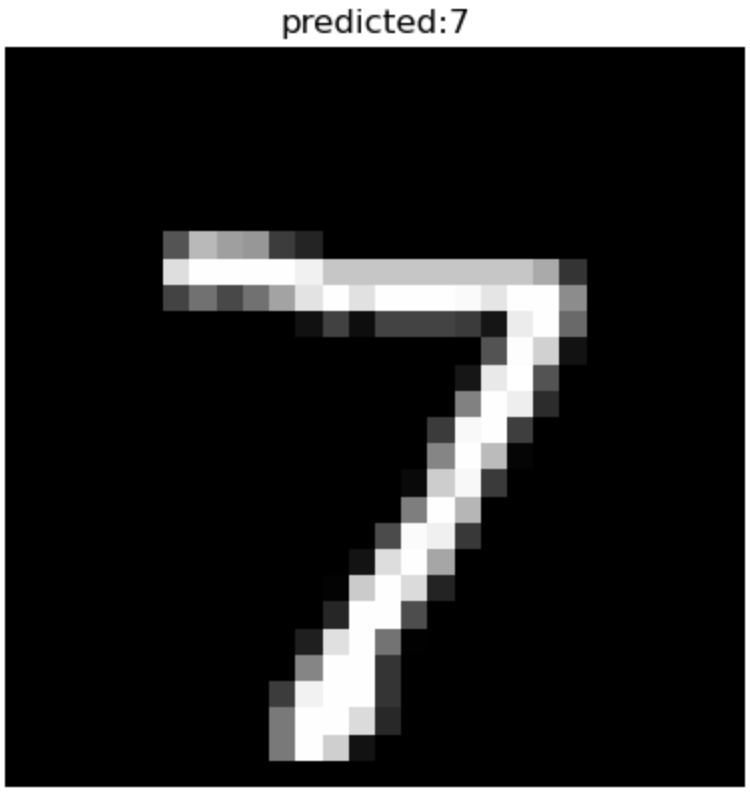
```
313/313 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - accuracy: 0.9732 - loss: 0.0842

Test Accuracy:0.9775
```

In [99]:
```
predictions=model.predict(x_test)
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 0s 875us/step
```

In [101…
```
import numpy as np
index=0
plt.imshow(x_test[index],cmap='gray')
plt.title(f"predicted:{np.argmax(predictions[index])}")
plt.axis('off')
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]: