In [17]:

```
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPool2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Dense
import matplotlib.pyplot as plt
```

In [2]:

```
(X_train,y_train) , (X_test,y_test)=mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-dat
asets/mnist.npz (https://storage.googleapis.com/tensorflow/tf-keras-dataset
s/mnist.npz)
11493376/11490434 [==============================] - 35s 3us/step
11501568/11490434 [==============================] - 35s 3us/step
```

In [3]:

```
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))
```

In [5]:

```
print(X_train.shape)
print(X_test.shape)
```

```
(60000, 28, 28, 1)
(10000, 28, 28, 1)
```

In [6]:

```
X_train=X_train/255
X_test=X_test/255
```

In [7]:

```
model=Sequential()
```

In [8]:

```
model.add(Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
```

In [9]:

```
model.add(MaxPool2D(2,2))
```

In [10]:

```
model.add(Flatten())
```

In [11]:

```
model.add(Dense(100,activation='relu'))
```

In [12]:

```
model.add(Dense(10,activation='softmax'))
```

In [13]:

```
model.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

In [14]:

```
model.fit(X_train,y_train,epochs=10)
```

```
Epoch 1/10
1875/1875 [==============================] - 30s 12ms/step - loss: 0.1682 -
accuracy: 0.9504
Epoch 2/10
1875/1875 [==============================] - 21s 11ms/step - loss: 0.0569 -
accuracy: 0.9830
Epoch 3/10
1875/1875 [==============================] - 21s 11ms/step - loss: 0.0373 -
accuracy: 0.9886
Epoch 4/10
1875/1875 [==============================] - 23s 12ms/step - loss: 0.0249 -
accuracy: 0.9923
Epoch 5/10
1875/1875 [==============================] - 24s 13ms/step - loss: 0.0173 -
accuracy: 0.9942
Epoch 6/10
1875/1875 [==============================] - 25s 13ms/step - loss: 0.0136 -
accuracy: 0.9955
Epoch 7/10
1875/1875 [==============================] - 23s 12ms/step - loss: 0.0098 -
accuracy: 0.9969
Epoch 8/10
1875/1875 [==============================] - 26s 14ms/step - loss: 0.0071 -
accuracy: 0.9976
Epoch 9/10
1875/1875 [==============================] - 26s 14ms/step - loss: 0.0063 -
accuracy: 0.9981
Epoch 10/10
1875/1875 [==============================] - 25s 13ms/step - loss: 0.0051 -
accuracy: 0.9983
```

Out[14]:

```
<keras.callbacks.History at 0x2bdba4be3b0>
```

In [15]:

```
model.evaluate(X_test,y_test)
```

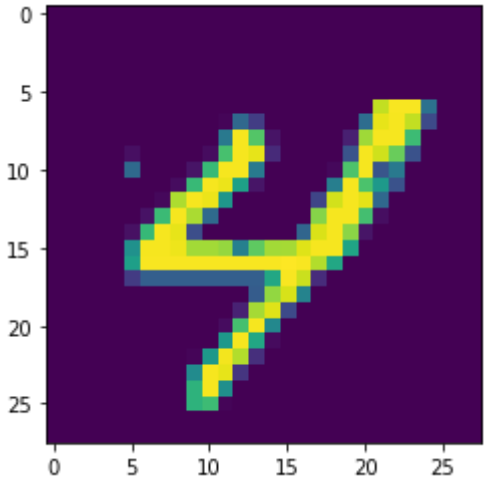
```
313/313 [==============================] - 2s 5ms/step - loss: 0.0532 - accu
racy: 0.9858
```

Out[15]:

```
[0.05316377058625221, 0.98580002784729]
```

In [20]:

```
plt.imshow(X_train[89, :])
plt.show()
```



In [26]:

```
y_pred=model.predict(X_test[89, :].reshape(1,28,28,1))
y_pred
```

Out[26]:

```
array([[9.6334700e-09, 9.9998164e-01, 3.5556809e-06, 1.1106345e-06,
        2.0665028e-07, 2.1412344e-10, 2.0225615e-09, 9.0603744e-06,
        4.4473359e-06, 2.2387640e-08]], dtype=float32)
```

In [28]:

```
plt.imshow(y_pred)
```

Out[28]:

```
<matplotlib.image.AxesImage at 0x2bdd7705360>
```

In [ ]:

```

```