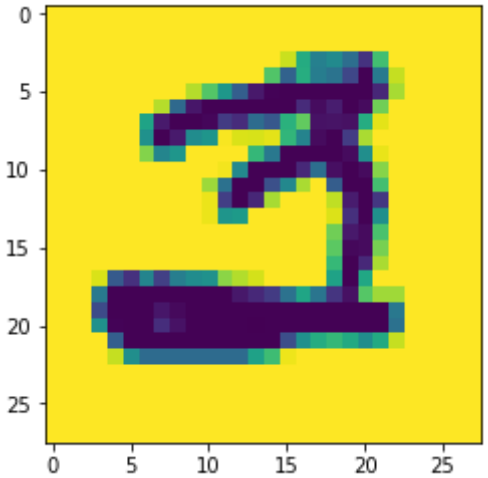
In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

from sklearn.neighbors import KNeighborsClassifier
```

提炼样本数据

In [2]:
```
img_arr = plt.imread('./num_img/3/3_100.bmp')
```

In [3]:
```
plt.imshow(img_arr)
```

Out[3]: <matplotlib.image.AxesImage at 0x1edf7d23940>



In [4]:
```
all_imgs_list = []
target_list = []
for i in range(10):
    for j in range(500):
        img_path = './num_img/'+str(i)+'/'+str(i)+'_'+str(j+1)+'.bmp'
        img_arr = plt.imread(img_path)
        all_imgs_list.append(img_arr)
        target_list.append(i)
```

In [5]:
```
len(all_imgs_list)
```

Out[5]: 5000

In [6]:
```
feature = np.array(all_imgs_list)
```

In [7]:
```
feature.shape
```

Out[7]: (5000, 28, 28)

In [8]:
```
#feature是一个三维数组（执行将维操作）
feature = feature.reshape(5000,28*28)
```

In [9]:
```
feature.shape
```

Out[9]: (5000, 784)

In [10]:
```
target = np.array(target_list)
```

将样本打乱

In [11]:
```
np.random.seed(3)
np.random.shuffle(feature)
np.random.seed(3)
np.random.shuffle(target)
```

获取训练数据 和 测试数据

In [12]:
```
x_train = feature[:4950]
y_train = target[:4950]
x_test = feature[-50:]
y_test = target[-50:]
```

实例化模型对象,训练

In [13]:
```
knn = KNeighborsClassifier(n_neighbors=30)
knn.fit(x_train,y_train)
knn.score(x_train,y_train)
```

Out[13]:
```
0.9195959595959596
```

In [14]:
```
# 预测 分类
knn.predict(x_test)
```

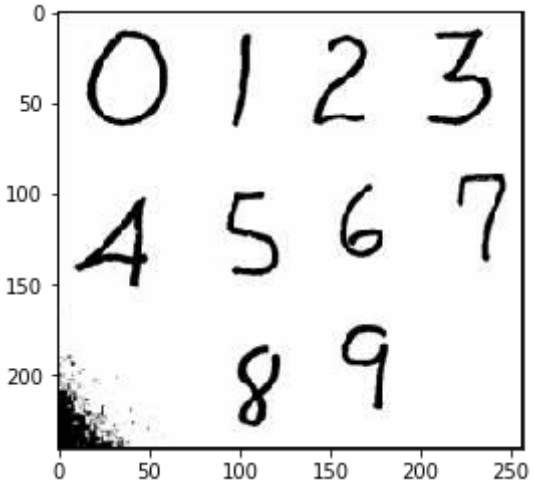
Out[14]:
```
array([4, 5, 7, 9, 7, 5, 7, 6, 8, 6, 1, 1, 3, 4, 8, 4, 1, 0, 1, 2, 0, 5,
       8, 6, 5, 9, 3, 9, 1, 8, 9, 6, 4, 1, 5, 0, 8, 7, 7, 1, 5, 3, 5, 5,
       6, 1, 1, 3, 6, 3])
```

In [15]:
```
# 真实数据
y_test
```

Out[15]:
```
array([4, 5, 7, 9, 7, 5, 7, 6, 8, 6, 4, 1, 3, 4, 8, 4, 2, 0, 1, 2, 0, 5,
       8, 6, 5, 9, 3, 9, 1, 8, 9, 6, 4, 1, 5, 2, 8, 7, 7, 2, 5, 3, 5, 5,
       6, 1, 1, 3, 6, 3])
```

In [16]:
```
# 外部图片 的识别
img_arr = plt.imread('./数字.jpg')
plt.imshow(img_arr)
```

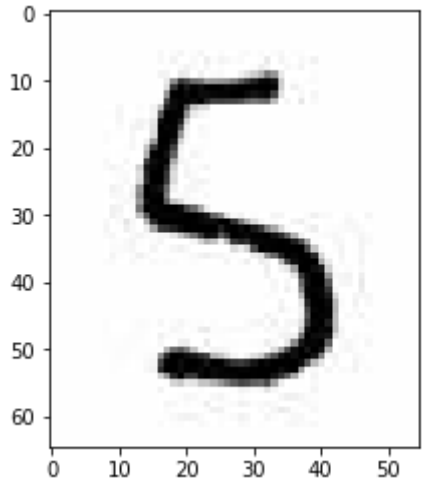
Out[16]: <matplotlib.image.AxesImage at 0x1edf8e4bcc0>



In [17]:
```
five_arr = img_arr[90:155,80:135]
```

In [18]:
```
plt.imshow(five_arr)
```

Out[18]: <matplotlib.image.AxesImage at 0x1edfb326278>



In [19]:
```
feature.shape
```

Out[19]: (5000, 784)

In [20]:
```
# five 数组是三维的,需要进行降维,舍弃第三个表示颜色的维度
five_arr = five_arr.mean(axis=2 )
```

In [21]:
```
five_arr.shape
```

Out[21]: (65, 55)

In [22]:
```
import scipy.ndimage as ndimage
five = ndimage.zoom(five_arr,zoom = (28/65,28/55))
```

```
c:\users\administrator\appdata\local\programs\python\python37\lib\site-packages\scipy\ndimage\interpolation.py:605: UserWarning: Fr
om scipy 0.13.0, the output shape of zoom() is calculated with round() instead of int() - for these inputs the size of the returned
array has changed.
  "the returned array has changed.", UserWarning)
```

In [23]:
```
five.shape
```

Out[23]: (28, 28)

In [24]:
```
feature.shape
```

Out[24]: (5000, 784)

In [25]:
```
knn.predict(five.reshape(1,784))
```

Out[25]: array([5])

In [26]:
```
from sklearn.externals import joblib  #保存 训练模型
```

In [27]:
```
joblib.dump(knn,'./digist.m')  # 保存 训练模型
```

Out[27]: ['./digist.m']

In [28]:
```
knn = joblib.load('./digist.m') # 加载训练模型
```

In [ ]:
```

```