

```
import gzip
import numpy as np

from sklearn.datasets import fetch_openml
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, recall_score, f1_score
```

```
# 读取图像数据
def load_images(filename):
    with gzip.open(filename, 'rb') as f:
        magic_number = int.from_bytes(f.read(4), 'big')
        num_images = int.from_bytes(f.read(4), 'big')
        num_rows = int.from_bytes(f.read(4), 'big')
        num_cols = int.from_bytes(f.read(4), 'big')
        images = np.frombuffer(f.read(), dtype=np.uint8)
        images = images.reshape(num_images, num_rows * num_cols)
        # 像素值范围缩放
    return images/255.0
```

```
# 读取标签数据
def load_labels(filename):
    with gzip.open(filename, 'rb') as f:
        magic_number = int.from_bytes(f.read(4), 'big')
        num_labels = int.from_bytes(f.read(4), 'big')
        labels = np.frombuffer(f.read(), dtype=np.uint8)
    return labels
```

```
# 加载训练集数据和标签
X_train = load_images('train-images-idx3-ubyte.gz')
y_train = load_labels('train-labels-idx1-ubyte.gz')
```

```
# 加载测试集数据和标签
X_test = load_images('t10k-images-idx3-ubyte.gz')
y_test = load_labels('t10k-labels-idx1-ubyte.gz')
```

```
print('X_train shape:', X_train.shape)
print('y_train shape:', y_train.shape)
print('X_test shape:', X_test.shape)
print('y_test shape:', y_test.shape)
```

```
X_train shape: (60000, 784)
y_train shape: (60000,)
X_test shape: (10000, 784)
y_test shape: (10000,)
```

```
# 训练拟合模型，核函数为linear
svm1 = SVC(kernel='linear', C=1.0)
svm1.fit(X_train, y_train)
```

```
# 训练拟合模型，核函数为poly
svm2 = SVC(kernel='poly', C=1.0)
svm2.fit(X_train, y_train)
```

```
# 训练拟合模型，核函数为sigmoid
svm3 = SVC(kernel='sigmoid', C=1.0)
svm3.fit(X_train, y_train)
```

```
# 训练拟合模型，核函数为rbf
svm4 = SVC(kernel='rbf', C=1.0)
svm4.fit(X_train, y_train)
```

```
# 在测试集上进行预测并计算准确率、召回率及F1
y_pred = svm4.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
recall = recall_score(y_test, y_pred, average='macro')
f1 = f1_score(y_test, y_pred, average='macro')
print('Accuracy:', accuracy)
print('Recall:', recall)
print('F1:', f1)
```

```
Accuracy: 0.9792
Recall: 0.9790919842945065
F1: 0.9791298259748042
```

经过测试和实验，验证了当选择核函数为rbf时，效果最好，模型各指标如上所示。