

BAHRIA UNIVERSITY, ISLAMABAD
Department of Computer Science

CEN 444
Digital Image Processing
Lab Journal 12

Student Name: M MUNEEB AHMED KIANI

Enrolment No.: 01-135212-063

Title: Digit Recognition, Pattern Recognition, Classification

Objectives: to understand template matching, Digit Recognition, Pattern Recognition, and solve a classification problem

Tools Used: Python

Procedure: Open IDLE and perform the following tasks

Use the template matching algorithm (function) you wrote. Make a small dataset of digits (0 to 9) with 5 samples in each class. You can use test images with only single digits and of the same size as the template, for simplicity and ease, if you want. This means you don't need to move the template window over test image. Both are of same size! Use KNN algorithm to find the digit in the test image.

Digit recognition is a classical pattern recognition problem. Each digit 0-9 represents a class. Features extracted from images of the digits can be used to generate a reference base (training data). A query digit image presented to the system can then be recognized using a classifier.

For this task, you are provided with multiple images of each digit as training data. Load each image, extract features and store the features of each image along with class label. Using k-nn classification

scheme, classify each of the images in the test data set into one of the digit classes. Examples of digits are illustrated in the following table.

DATASET:

	0	194	226	224	207	213	228	21
1	1	217	231	224	228	226	227	22
2	10	255	255	255	255	255	255	23
3	11	255	255	254	253	253	82	24
4	12	255	255	255	254	255	251	25
5	13	255	255	255	254	252	255	26
6	14	255	255	255	254	253	252	27
7	15	255	255	255	254	249	164	28
8	16	255	255	255	255	255	97	29
9	17	255	255	255	255	252	249	30
10	18	255	255	255	250	252	72	31
11	19	255	255	255	255	254	254	32
12	2	247	249	201	245	247	250	33
13	20	255	255	255	255	254	255	34
14	3	205	229	220	203	220	229	35
15	4	232	219	232	187	229	222	36
16	5	209	218	211	212	215	218	37
17	6	212	216	217	199	221	210	38
18	7	220	208	216	214	214	217	39
19	8	211	236	235	216	228	237	40
20	9	239	230	236	202	237	230	41

IMAGE PREPARATION:

```
import cv2
import numpy as np

def create_digit_image(digit, output_filename):
    img = np.ones((28, 28), dtype=np.uint8) * 255
    font = cv2.FONT_HERSHEY_SIMPLEX
    font_scale = 1.0
    color = (0, 0, 0)
    thickness = 2
    (w, h), _ = cv2.getTextSize(str(digit), font, font_scale, thickness)
    x = (img.shape[1] - w) // 2
    y = (img.shape[0] + h) // 2
    cv2.putText(img, str(digit), (x, y), font, font_scale, color, thickness)
    cv2.imwrite(output_filename, img)
    print(f"Image saved as {output_filename}")
create_digit_image(3, 'testimage1.png')
```

Image saved as testimage1.png

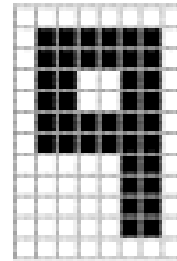
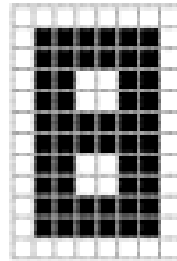
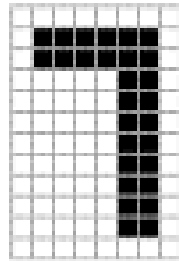
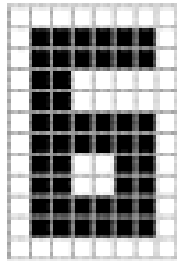
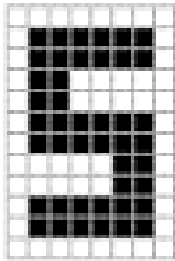
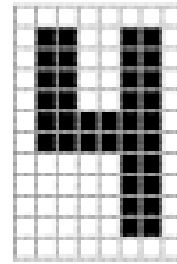
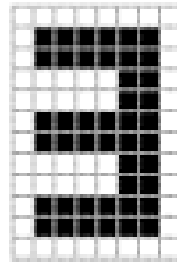
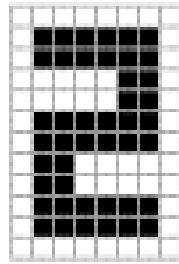
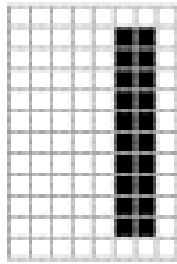
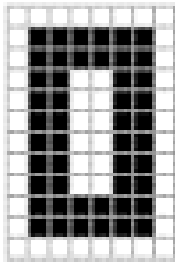
TESTIMAGE 1:

3

OUTPUT:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
import cv2
import numpy as np
def train_knn(csv_filename):
    data = pd.read_csv(csv_filename, header=None)
    X = data.iloc[:, 1:].values
    y = data.iloc[:, 0].values
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    knn = KNeighborsClassifier(n_neighbors=3)
    knn.fit(X_train, y_train)
    return knn
def predict_digit(image_path, model):
    img = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    img = cv2.resize(img, (28, 28))
    img = img.flatten().reshape(1, -1)
    prediction = model.predict(img)
    return prediction[0]
def main():
    csv_filename = 'digit_dataset.csv'
    knn_model = train_knn(csv_filename)
    test_image_path = 'testimage1.png'
    predicted_digit = predict_digit(test_image_path, knn_model)
    print(f"The predicted digit is: {predicted_digit}")
if __name__ == "__main__":
    main()
```

The predicted digit is: 3



Submission Date:

Signature Ms. Umarah Qaseem