Team ID	PNT2022TMID41253
Project Name	A Novel Method For Handwritten Digit Recognition System

SPRINT – 2

load.py

```
import os
import random
import string
from pathlib import Path
import numpy as np
from tensorflow.keras.models import load_model
from PIL import Image, ImageOps
import cv2
def recognize(image: bytes) -> int:
model=load_model(Path("./model/mnist.h5"))
 image = cv2.imread(image)
 grey = cv2.cvtColor(image.copy(), cv2.COLOR_BGR2GRAY)
 ret, thresh = cv2.threshold(grey.copy(), 75, 255, cv2.THRESH_BINARY_INV)
 contours, _ = cv2.findContours(thresh.copy(), cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)
 preprocessed_digits = []
 if not contours:
     return 'not recognized', '', 'error'
 for c in contours:
     x,y,w,h = cv2.boundingRect(c)
     cv2.rectangle(image, (x,y), (x+w, y+h), color=(0, 255, 0), thickness=2)
     digit = thresh[y:y+h, x:x+w]
     resized_digit = cv2.resize(digit, (18,18))
     padded_digit = np.pad(resized_digit, ((5,5),(5,5)), "constant", constant_values=0)
     preprocessed_digits.append(padded_digit)
 for digit in preprocessed_digits:
     prediction = model.predict(digit.reshape(1, 28, 28, 1))
     best= np.argmax(prediction)
 return best, "1.jpg", max(prediction[0])
result = []
for index in range(10):
    a, b, c = recognize('data/'+str(20+index)+'.jpg')
    result.append([index,'->',a,'=',c])
print(*result, sep="\n")
```