```python
import uvicorn
import numpy as np
from PIL import Image
from io import BytesIO
from fastapi import FastAPI
from fastapi import UploadFile, File
from tensorflow import keras

appi = FastAPI()

model_path = r'D:\sem 8\CS5830\Assignment 6\model_to_use.h5'

def read_image(image_obj):
    '''
        Function to read the image
        input: png format images
        output: image in the required format (1, 784) array
    '''
    img = Image.open(BytesIO(image_obj.file.read()))
    img = format_image(img)
    return img

def format_image(img):
    '''
        Function to format the image to the standard MNIST shape and size
        input: image object
        output: image in the required format (1, 784) array
    '''
    if img.size != (28, 28):
        img = img.resize((28, 28))

    img = img.convert('L')
    img = np.array(img)
    img = img.reshape(1, 784)
    img = img/255
    return img

def load_model(path:str):
    '''
        Function to load the trained data
        input: saved model
        output: model after loading
    '''
    loaded_model = keras.models.load_model('model_to_use.h5')
    return loaded_model

def predict_digit(model, data):
    '''
        Function to predict the digit based on the image provided
        input: model and the data point
        output: digit
    '''
    probs = model.predict(data, verbose=True)
    print("Predicted Digit:", np.argmax(probs))
    return str(np.argmax(probs))


@appi.get('/')
def root():
    return {'Message': 'Hello there! this the assignment 6th of the Big Data Lab where we build
a FastAPI for MNIST handwritten digit image classification.'}

@appi.post('/predict')
def predict_image(file: UploadFile = File(...)):
    # Read and format the file
    image = read_image(file)
    # load model
```

```
    model = load_model(model_path)
    # Predict
    pred = predict_digit(model, image)
    return pred

if __name__ == '__main__':
    uvicorn.run(appi, host = '123.4.5.6', port = 8080)
```