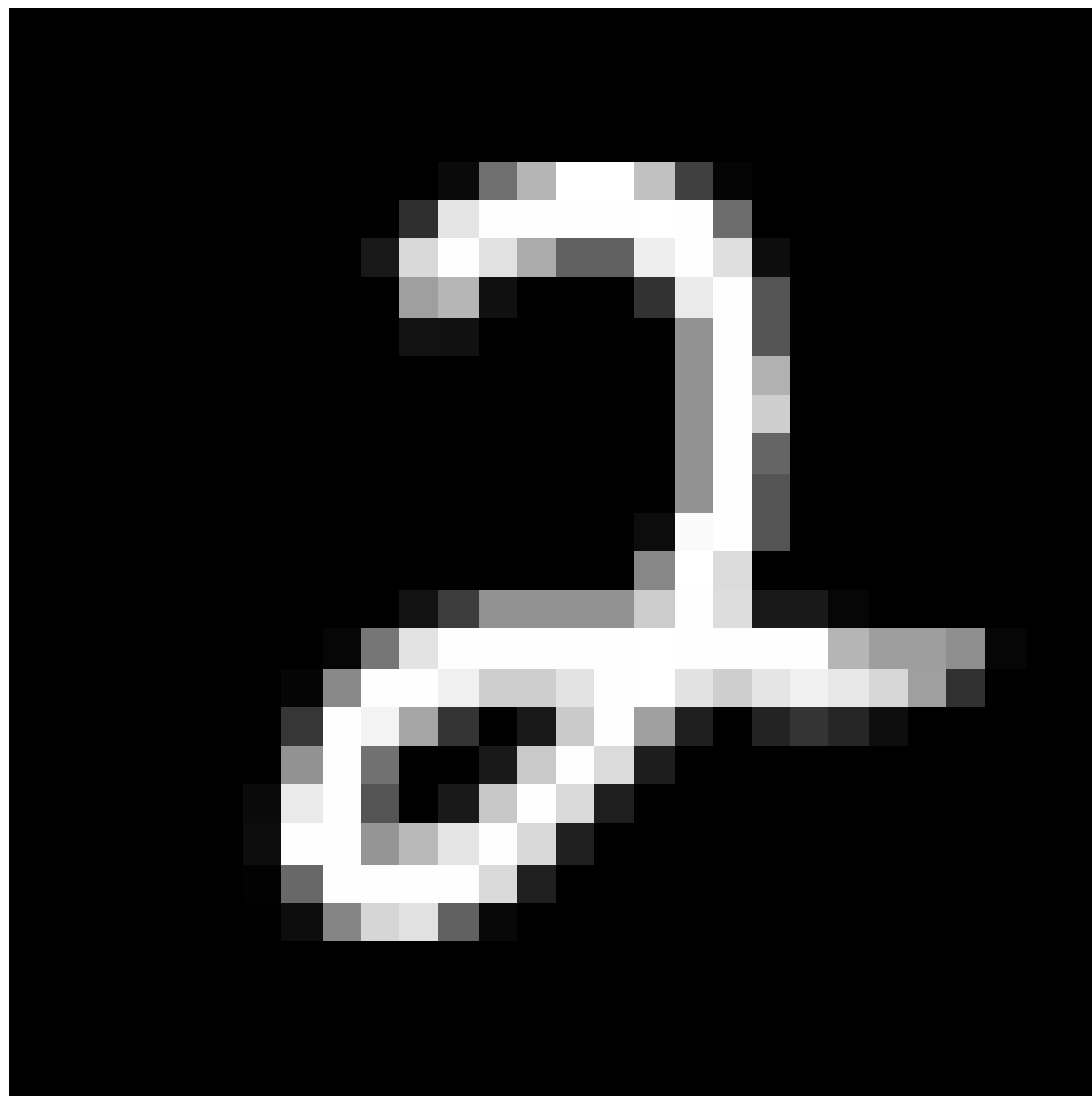
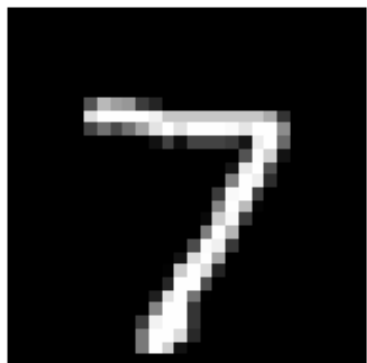


Predicted Digit: 2



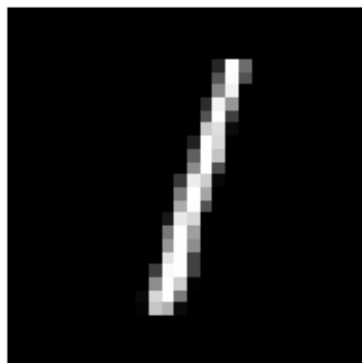
Pred: 7



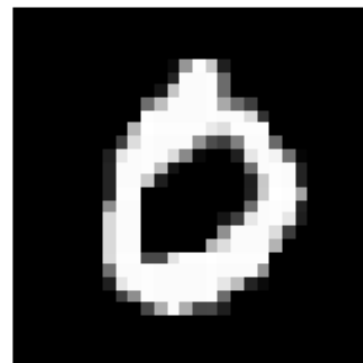
Pred: 2



Pred: 1



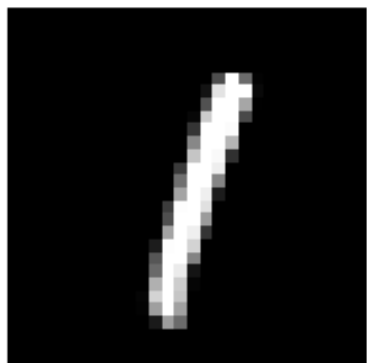
Pred: 0



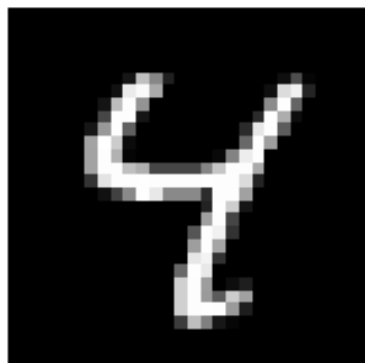
Pred: 4



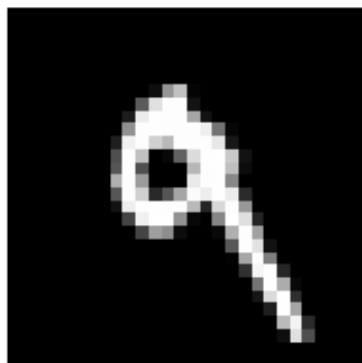
Pred: 1



Pred: 4



Pred: 9

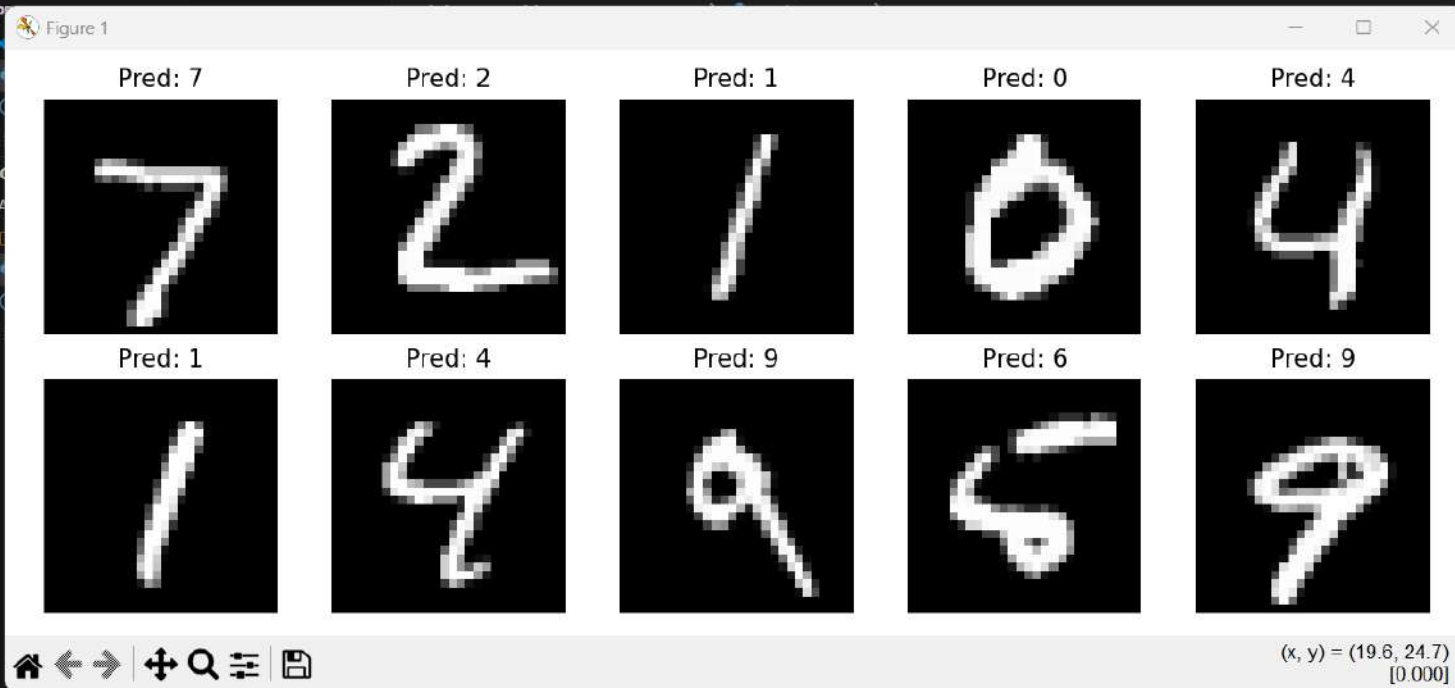


Pred: 6



Pred: 9





```
val_accuracy: 0.9913
Epoch 4/5
844/844 [=====] - 30s 35ms/step - loss: 0.0260 - accuracy: 0.9920 - val_loss: 0.0357 -
val_accuracy: 0.9910
Epoch 5/5
844/844 [=====] - 30s 36ms/step - loss: 0.0188 - accuracy: 0.9942 - val_loss: 0.0387 -
val_accuracy: 0.9888
313/313 [=====] - 3s 8ms/step - loss: 0.0297 - accuracy: 0.9902
Test Accuracy: 0.9901999831199646
313/313 [=====] - 2s 7ms/step
Accuracy Score: 0.9902
```

Build with Agent

AI responses may be inaccurate.
[Generate Agent Instructions](#) to onboard AI
onto your codebase.

+ mnist_cnn.py

Describe what to build next

Agent v Auto v



VS Code interface showing a Digit Recognition project. The Explorer pane on the left lists files: `mnist_cnn.py`, `mnist.py`, `README.md`, and `requirements.txt`. The main editor displays the `mnist.py` file with the following code:

```
Digit Recognition (MNIST Dataset) > mnist.py > ...
25     metrics=['accuracy']
26 )
27
28 model.fit(x_train, y_train, epochs=5, batch_size=32)
```

A "Figure 1" window is overlaid, displaying a handwritten digit "2" on a black background. The text "Predicted Digit: 2" is visible above the image.

The bottom status bar shows the current file is `mnist.py`, line 42, column 1, with a Python interpreter (3.14.2) and Prettier formatting.

On the right side, the "Build with Agent" panel is visible, indicating AI responses may be inaccurate and providing a link to "Generate Agent Instructions". Below this, a chat input area is shown with the text "Describe what to build next" and buttons for "Agent", "Auto", and a send button.

The bottom taskbar shows the system clock at 12:46 AM on 31-01-2026, with 86% battery and English as the system language.

File Edit Selection View Go Run Terminal Help

Digit Recognition

CHAT

EXPLORER

OPEN EDITORS

DIGIT RECOGNITION

OUTLINE

TIMELINE

Welcome

mnist_cnn.py 3, U

mnist.py 2, U

README.md U

requirements.txt U

mnist.py

25 metrics=['accuracy']

26)

27

28 model.fit(x_train, y_train, epochs=5, batch_size=32)

29


30 test_loss, test_accuracy = model.evaluate(x_test, y_test)

31 print("Test Accuracy:", test_accuracy)

32

Figure 1

Predicted Digit: 2



JavaSE-22

0.9316

0.9494

0.9613

0.9692

0.9255

Describe what to build next

Agent Auto

20°C Clear

12:48 AM 31-01-2026

Digit Recognition (MNIST)



Clear Predict

Predicted Digit: 2

Elements Console Sources Network >> 1

top Filter Default levels No Issues

1 hidden

```
Uncaught (in promise) TypeError: tf.data.mnist is not a function
    at loadMNIST (main.js:63:31)
    at trainModel (main.js:77:28)
    at main.js:87:1
```

Digit Recognition System

MNIST Dataset using CNN

TensorFlow.js • Web Based

- ✓ Train a **CNN** on MNIST
- ✓ 5 Epochs Training
- ✓ Accuracy shown in **Console**

Open **Console (F12)** to view training output

Elements Console Sources Network >>

top Filter Default levels No Issues

```
[null,5,5,32] 0 tfjs@latest:17  
  
flatten_Flatten1 (Flatten) [[null,5,5,32]] tfjs@latest:17  
[null,800] 0 tfjs@latest:17  
  
dense_Dense1 (Dense) [[null,800]] tfjs@latest:17  
[null,128] 102528 tfjs@latest:17  
  
dense_Dense2 (Dense) [[null,128]] tfjs@latest:17  
[null,10] 1290 tfjs@latest:17  
===== tfjs@latest:17  
===== tfjs@latest:17  
  
Total params: 108618 tfjs@latest:17  
Trainable params: 108618 tfjs@latest:17  
Non-trainable params: 0 tfjs@latest:17  
  
Training started... main.js:97  
Epoch 1: Accuracy = 91.80% main.js:105  
Epoch 2: Accuracy = 97.68% main.js:105  
Epoch 3: Accuracy = 98.33% main.js:105  
Epoch 4: Accuracy = 98.68% main.js:105  
Epoch 5: Accuracy = 98.88% main.js:105  
Tensor tfjs@latest:17  
0.98499995470047  
Training completed main.js:115  
> |
```

Console AI assistance What's new

Assignment 1

Train CNN on MNIST Dataset

Epochs: 5

Check browser console for accuracy

TensorFlow.js | CO2 – Analyze

```
max_pooling2d_MaxPooling2D1 [[null,26,26,8]] tfjs@latest:17
[null,13,13,8] 0
tfjs@latest:17

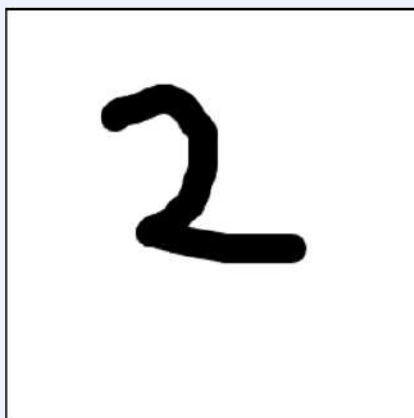
flatten_Flatten1 (Flatten) [[null,13,13,8]] tfjs@latest:17
[null,1352] 0
tfjs@latest:17

dense_Dense1 (Dense) [[null,1352]] tfjs@latest:17
[null,64] 86592
tfjs@latest:17

dense_Dense2 (Dense) [[null,64]] tfjs@latest:17
[null,10] 650
tfjs@latest:17
===== tfjs@latest:17
=====
Total params: 87322 tfjs@latest:17
Trainable params: 87322 tfjs@latest:17
Non-trainable params: 0 tfjs@latest:17

Training Assignment-1 CNN... main.js:95
Epoch 1 Accuracy: 61.00% main.js:103
Epoch 2 Accuracy: 86.04% main.js:103
Epoch 3 Accuracy: 89.60% main.js:103
Epoch 4 Accuracy: 90.90% main.js:103
Epoch 5 Accuracy: 92.56% main.js:103
Assignment-1 Completed main.js:110
```

Assignment 2: Draw & Predict Digit



Clear

Predict

Prediction: 2

Live reload enabled. index.html:64

> |



Live reload enabled. index.html:137

>

CNN vs Dense Network

Machine Learning Tools Lab – Assignment 3

Train Dense Model

Train CNN Model

Training CNN Model...
CNN Epoch 1 Accuracy: 0.1180
CNN Epoch 2 Accuracy: 0.1260
CNN Epoch 3 Accuracy: 0.1360
CNN Epoch 4 Accuracy: 0.1370
CNN Epoch 5 Accuracy: 0.1440

CNN Model Training Complete

TensorFlow.js | Image Classification Demo