

MNIST Handwritten Digits Recognition Use-case

01 November 2025 10:01

Modified National Institute of Standards
and Technology **database**

Using [TensorFlow & Keras]

MNIST database

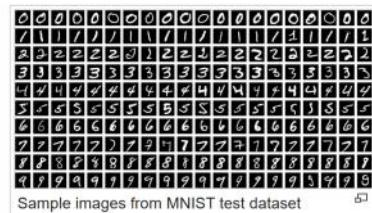
14 languages

Article Talk

Read Edit View history Tools

From Wikipedia, the free encyclopedia

The **MNIST database** (*Modified National Institute of Standards and Technology database*)^[1] is a large database of handwritten digits that is commonly used for training various image processing systems.^{[2][3]} The database is also widely used for training and testing in the field of machine learning.^{[4][5]} It was created by "re-mixing" the samples from NIST's original datasets.^[6] The creators felt that since NIST's training dataset was taken from American Census Bureau employees, while the testing dataset was taken from American high school students, it was not well-suited for machine learning experiments.^[7] Furthermore, the black and white images from NIST were normalized to fit into a **28x28 pixel bounding box** and anti-aliased, which introduced grayscale levels.^[7]



Sample images from MNIST test dataset

The MNIST database contains **60,000 training images** and **10,000 testing images**.^[8] Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset.^[9] The original creators of the database keep a list of some of the methods tested on it.^[7] In their original paper, they use a **support-vector machine** to get an error rate of 0.8%.^[10]

The original MNIST dataset contains at least 4 wrong labels.^[11]

Training dataset # 60,000 images → dataset #1

Testing dataset # 10,000 images → dataset #2

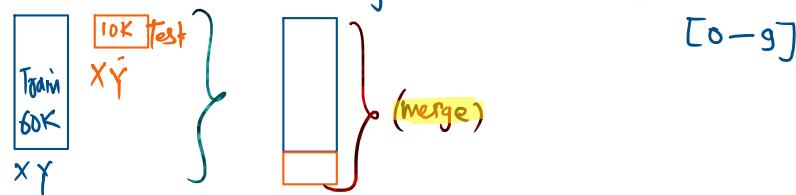
Total images → 70,000 images

Testing proportion stands at = $10000/70000 = 0.1429 * 100 = 14.29\%$

To get a custom train and test split

Training 70% Testing 30%

Step #1 Merge train and test dataset along with the appropriate labels



Step #2 Using scikit-learn, apply 'train-test-split()' and get the train and test of the required size using 'test_size' parameter

and test of the required size using `test_size` parameter

20 samples

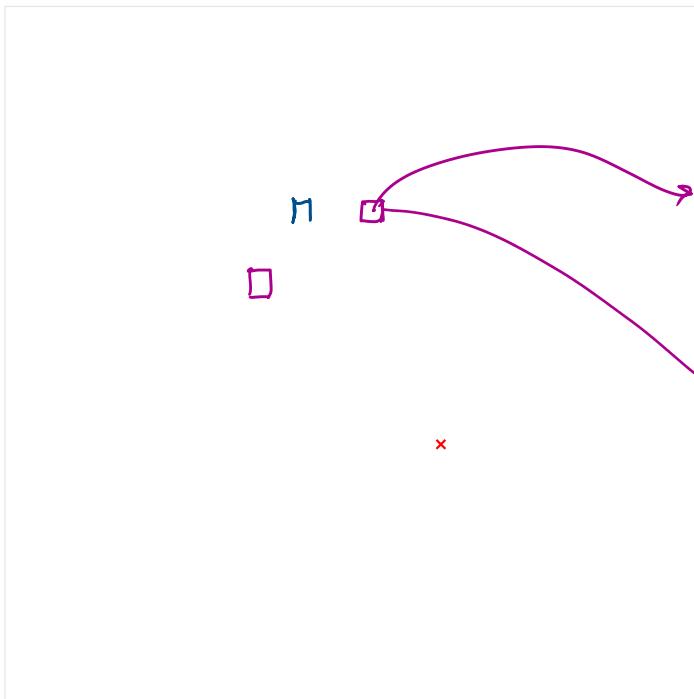
0.3
↓
30.0.

p1 p2 p3

p1g p2g



7
↔ 28px
↔ 28px



0 255

Total combinations : $2^8 = 256$ combinations

We'll be using **MNIST data** from TF datasets repo.

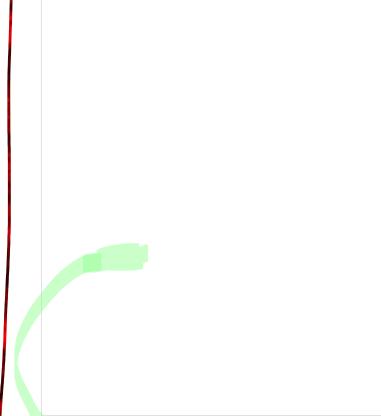
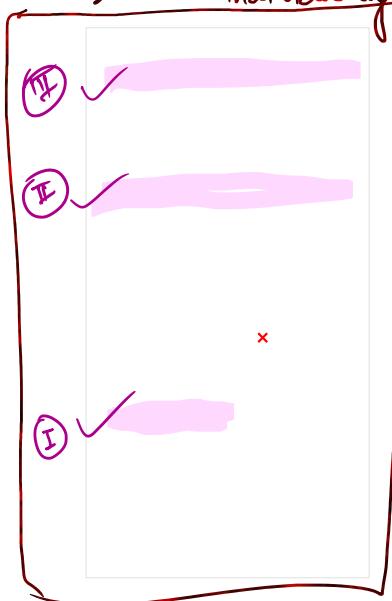
in the
syllabus

✓
✓

✗

→
→

can be done individually



(4x4 array)

1	2	3	4
1x1	1x2	1x3	
			4x4

✗

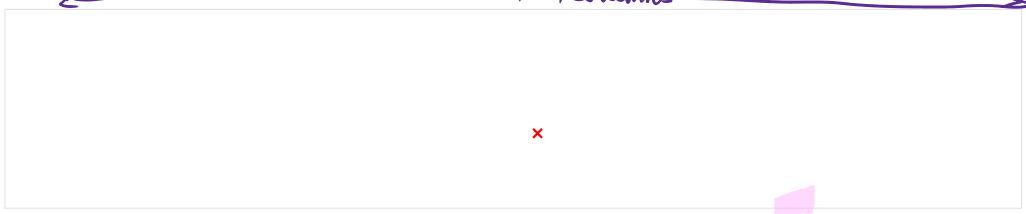


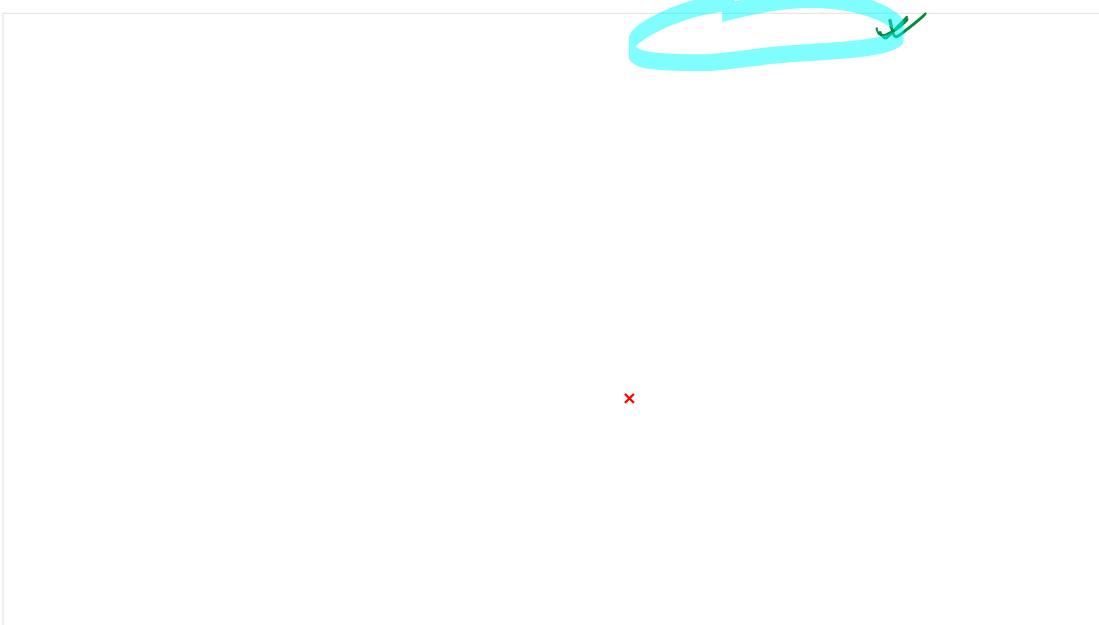
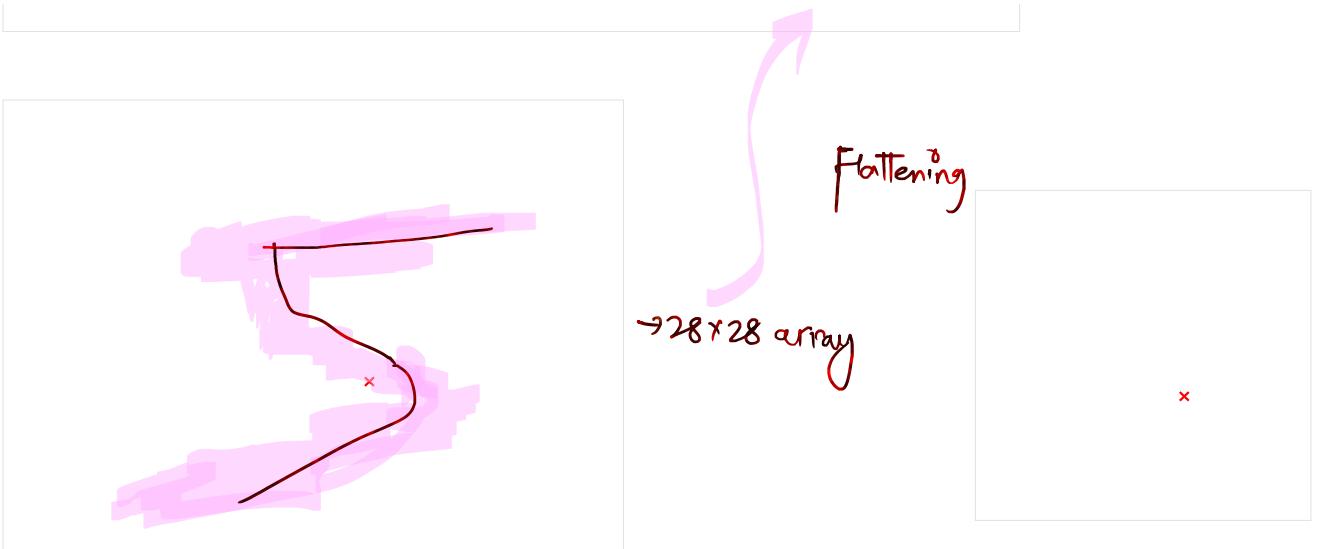
0 1 2 3 4 5 6 7 8 9
← 10 classes →



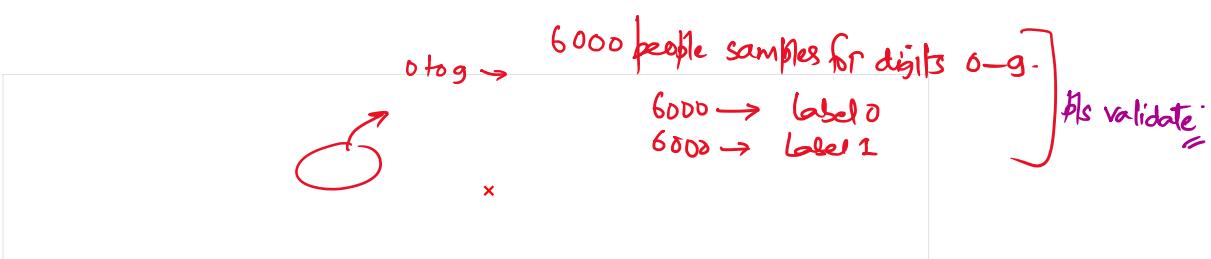
784 columns

any
row is
a digit

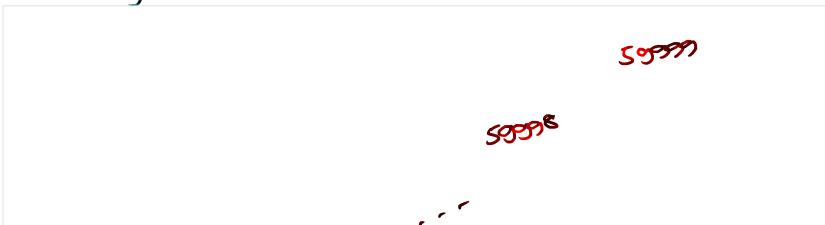


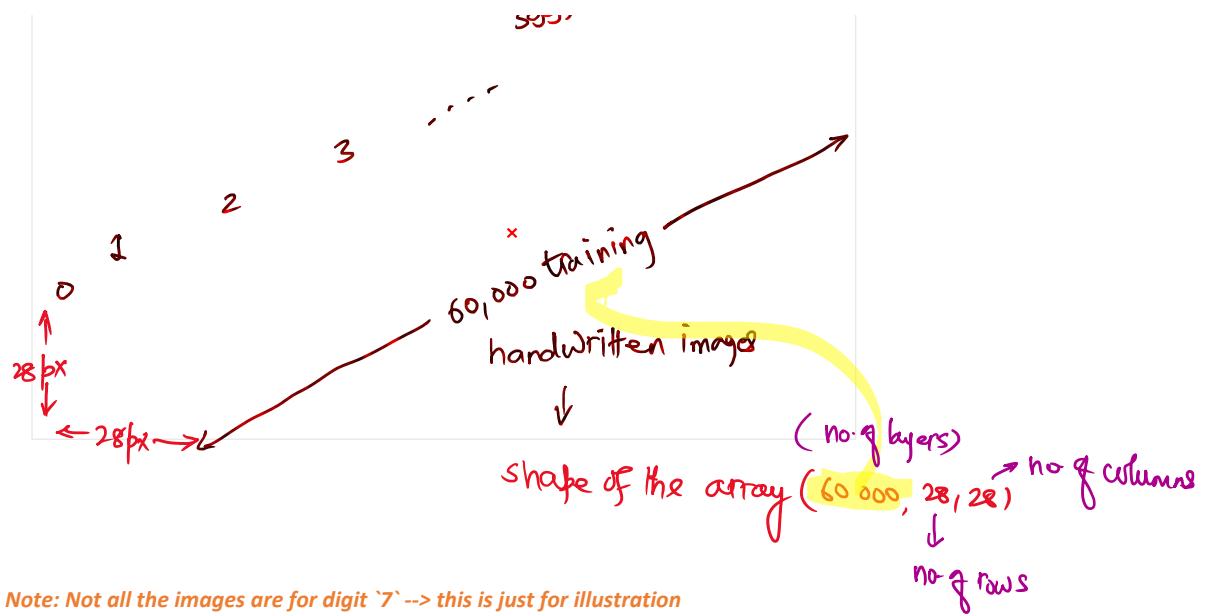


Let us understand now MNIST dataset architecture



Training dataset structure





BUILDING THE MODEL'S ARCHITECTURE

MNIST Handwritten digits NN model's architecture

Input layer → 2 Hidden layer(s) → output layer

Forward Propagation

