The Boundary Forest Algorithm for Fast Online Supervised Learning

Charles Mathy, Nate Derbinsky, José Bento, Jonathan Rosenthal, Jonathan Yedidia

Disney Research

The problem

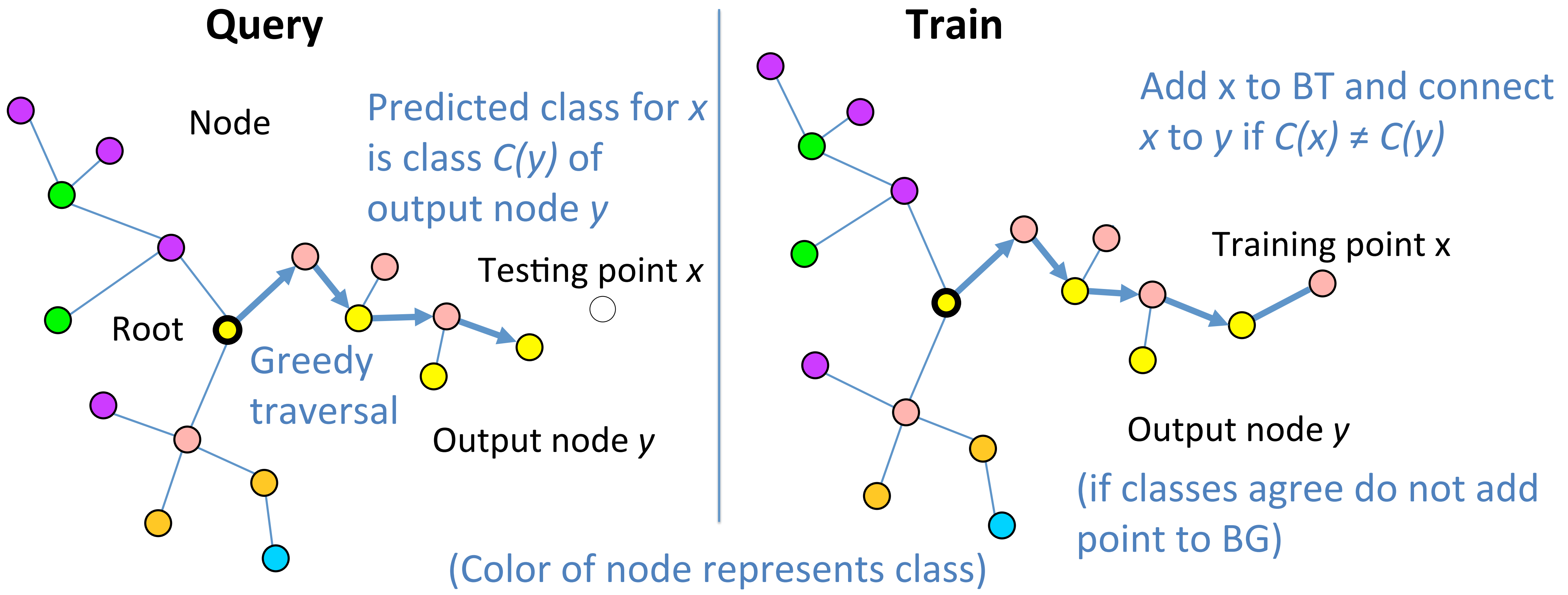
We want a supervised learning algorithm to have good generalization, but also the following properties:

- Fast online processing of both training examples and test queries
- One-shot learning, and can achieve zero error on training set
- Can learn and represent complex functions
- Able to absorb and learn from unlimited training examples – query time and memory used grow only very slowly with number of training examples
- Easy to understand how and why it works

Boundary Tree

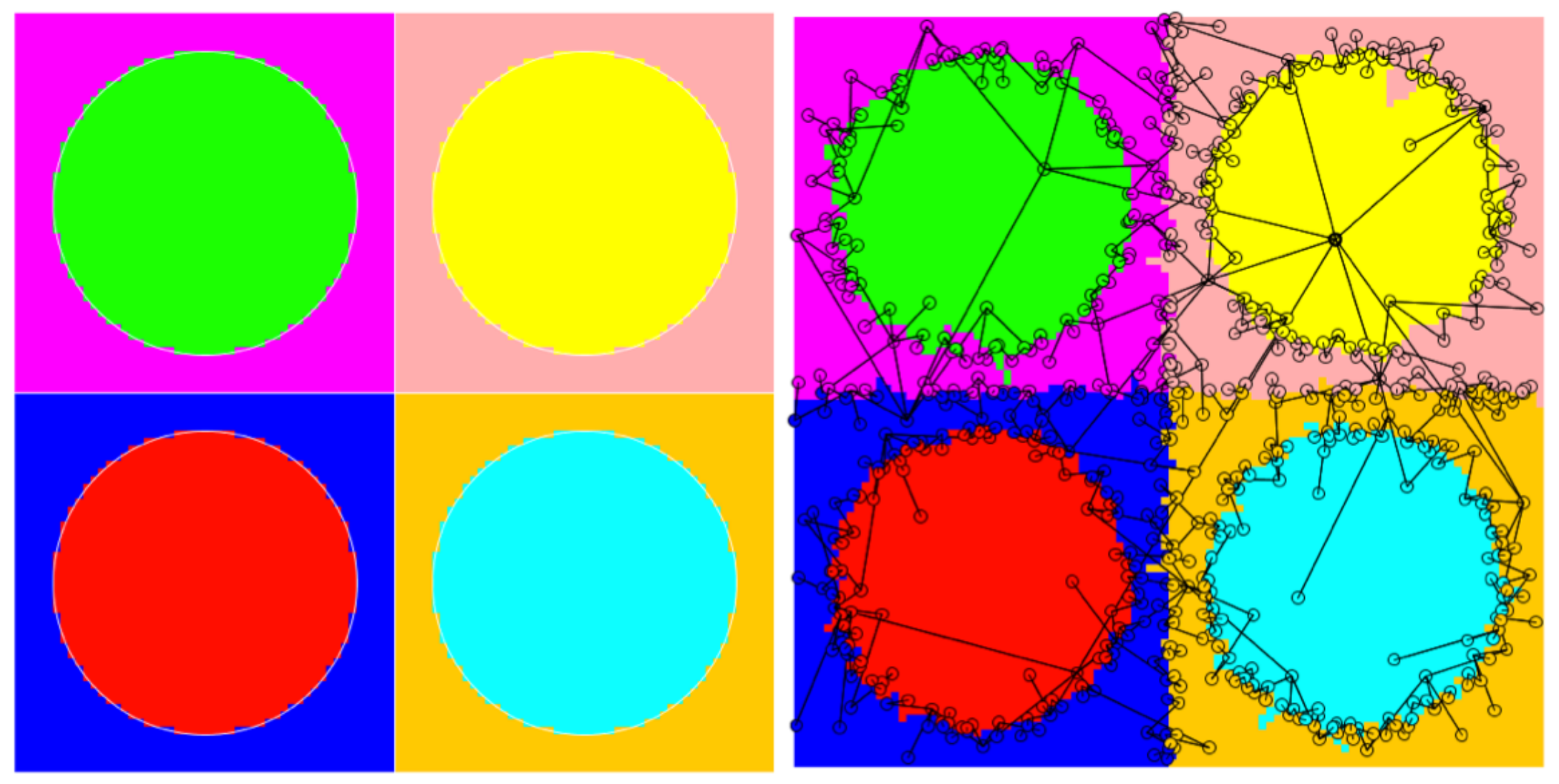
The nodes of the Boundary Tree (BT) are previously seen data points x and their associated class *C(x)*. When queried with a new point *x*, it finds a point *y* close to *x* (caveat: the algorithm requires a metric). The class *C(y)* of *y* is the predicted class of *x*.

The output point is found by starting at the root of the tree and recursively looking through children for the node closest to *x*. It stops when it finds a locally closest node.


Query
Node
Predicted class for x is class C(y) of output node y
Testing point x
Root
Greedy traversal
Output node y
Train
Add x to BT and connect x to y if C(x) ≠ C(y)
Training point x
Output node y
(if classes agree do not add point to BG)
(Color of node represents class)

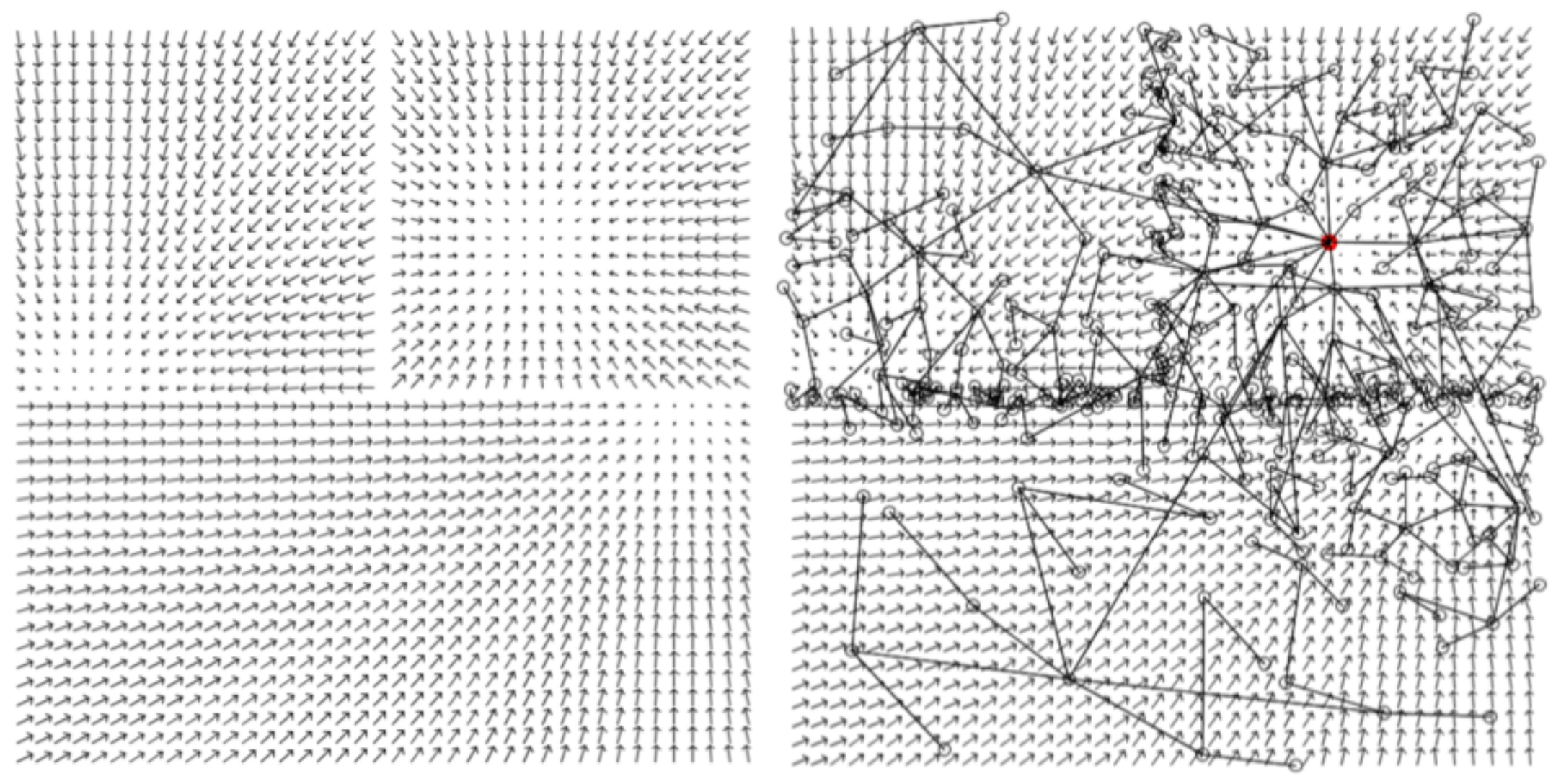
Toy examples

Toy classification problem



Ground truth BG: 724 nodes after seeing 10,000 samples

Toy regression problem



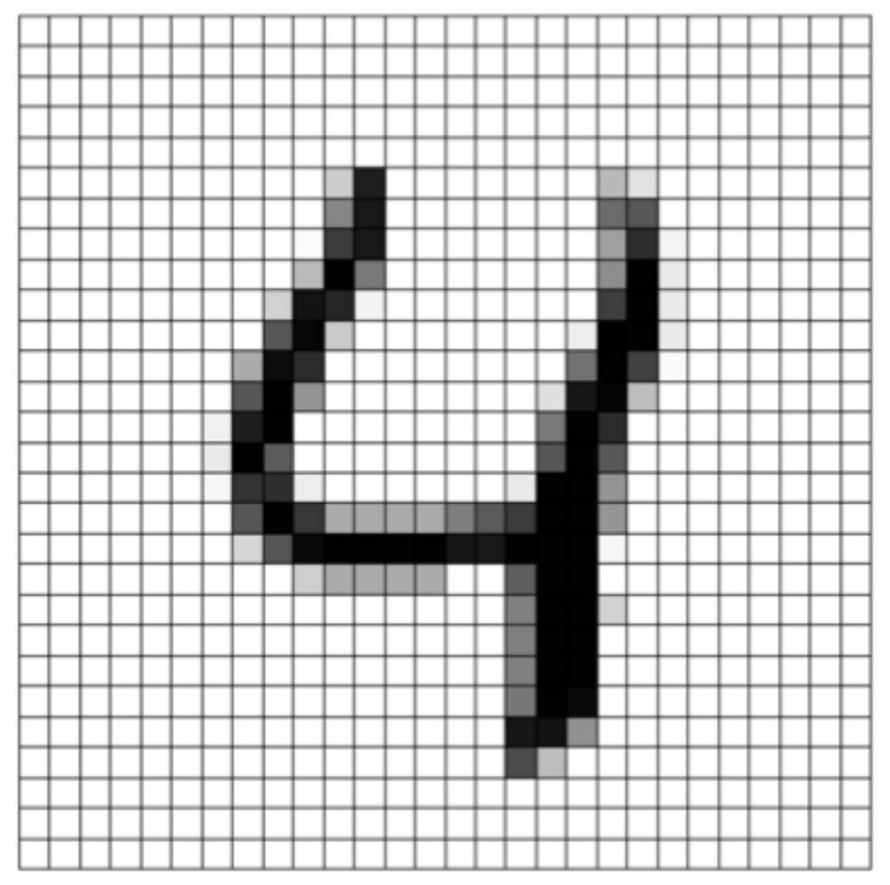
Ground truth BG: 395 nodes after seeing 10,000 samples

Comparison with naïve Nearest Neighbor:

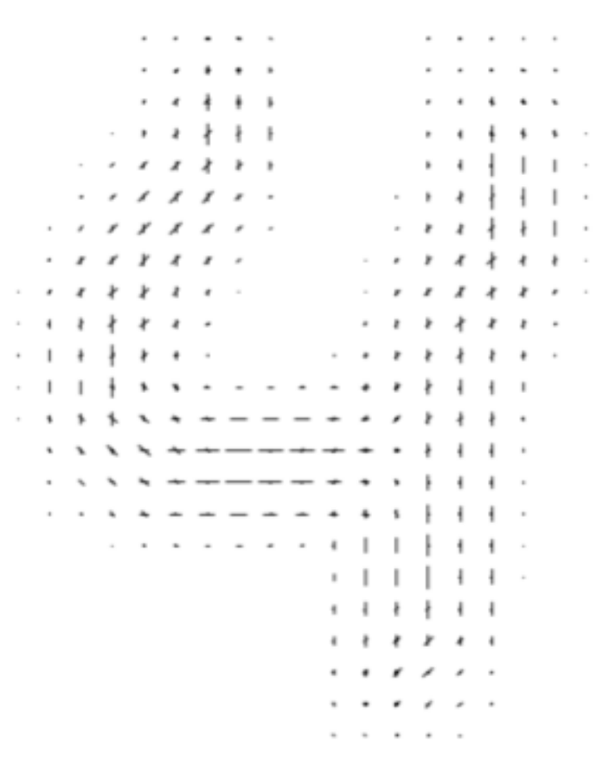
- Compresses the data by only storing points near boundaries
- Provides tree data structure for fast training and testing
- Empirically gives similar accuracy to kNN while retaining space and time advantage, when building forest of several Boundary Trees

Benchmark

- MNIST: 60,000 labeled handwritten digits for training, 10,000 for testing (784 pixels).
- K Nearest Neighbor with Euclidean L2 distance: 97.1% accuracy (3-NN).
- Committee of 30 boundary graphs: 97.6%.
- Training on full MNIST with 4 cores in 17 seconds in Java.
- Testing on 10,000 samples in 4 seconds. 3-NN takes about 2 hours.
- Better metric: HOG (Histogram of Gradients). Gets 98.9% accuracy (3-NN: 98.6%).



Original image



Histogram of gradients representation 27x27x4 = 2916 dimensional.