
Vehicle Identification Image Recognition



EE380L Data Mining

Ghosh

Andrew Cary, Olamide Kolawole, Evan Platt, Chance Pianta

Overview & Datasets

Exploring the use of classification algorithms to find and identify cars as class objects in image data sets.

Dataset:

- Stanford Cars dataset which contains 8,144 training images and 8,041 testing images (16,185 total car images)
 - ImageNet large-scale database.
-

Preprocessing

Preprocessing

Greyscaling and Otsu thresholding applied



Preprocessing



Preprocessing

Clearing the image along the border

```
>>> import numpy as np
>>> from skimage.segmentation import clear_border
>>> image = np.array([[0, 0, 0, 0, 0, 0, 0, 1, 0],
...                   [0, 0, 0, 0, 1, 0, 0, 0, 0],
...                   [1, 0, 0, 1, 0, 1, 0, 0, 0],
...                   [0, 0, 1, 1, 1, 1, 1, 0, 0],
...                   [0, 1, 1, 1, 1, 1, 1, 1, 0],
...                   [0, 0, 0, 0, 0, 0, 0, 0, 0]])
>>> clear_border(image)
array([[0, 0, 0, 0, 0, 0, 0, 0, 0],
       [0, 0, 0, 0, 1, 0, 0, 0, 0],
       [0, 0, 0, 1, 0, 1, 0, 0, 0],
       [0, 0, 1, 1, 1, 1, 1, 0, 0],
       [0, 1, 1, 1, 1, 1, 1, 1, 0],
       [0, 0, 0, 0, 0, 0, 0, 0, 0]])
```

Preprocessing

Some detail is lost



Preprocessing

Canny Edge Detection



Canny edge with varying smoothing



Preprocessing

coordinates of edges zeroed out, then border is cleared



Preprocessing

Small objects removed. Parameter is passed on how small objects can be.



Classification Algorithms

Haar-like features / Haar-cascade

Shaped overlays are used to detect similar areas and patterns of contrasting shades between photos

SVM

Attempt to create a binary classifier by choosing a hyperplane separating the data by class

Convolutional Neural Network

The input (image) is transformed through a series of neuron layers to a final output holding class scores.

Haar-Cascade: Model

Haar-like features / Haar-cascade

Shaped overlays are used to detect similar areas and patterns of contrasting shades between photos

Haar-Cascade: Experiment

Haar-like features / Haar-cascade

Shaped overlays are used to detect similar areas and patterns of contrasting shades between photos

SVM: Model

SVM

Attempt to create a binary classifier by choosing a hyperplane separating the data by class

SVM: Experiment

SVM

Attempt to create a binary classifier by choosing a hyperplane separating the data by class

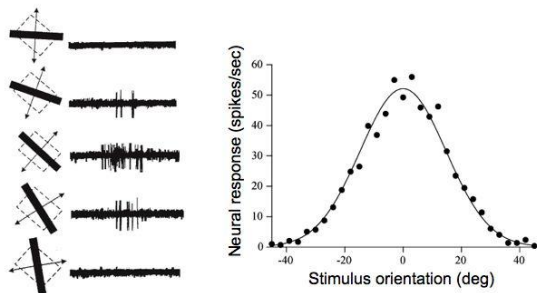
Convolutional Neural Networks

ConvNets / CNNs

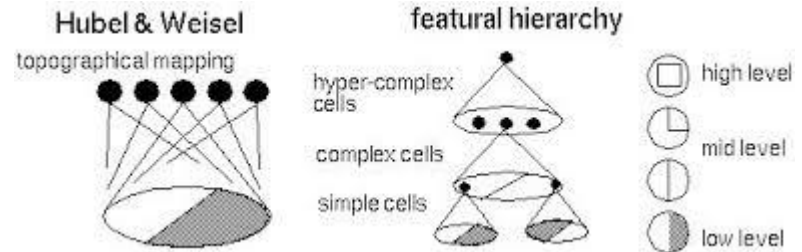
ConvNet: Model

- Feed-forward artificial neural networks with learnable weights and biases often used for image recognition in machine learning
- Trace to Hubel and Wiesel's breakthrough understanding of the locally-sensitive, orientation-specific neurons in a cat's visual cortex in field of biology in 1960's

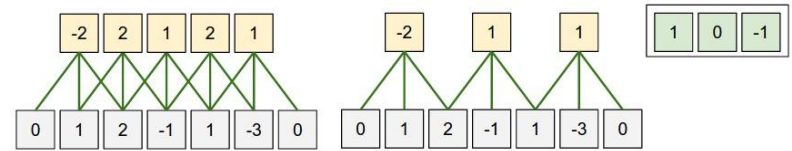
V1 physiology: orientation selectivity



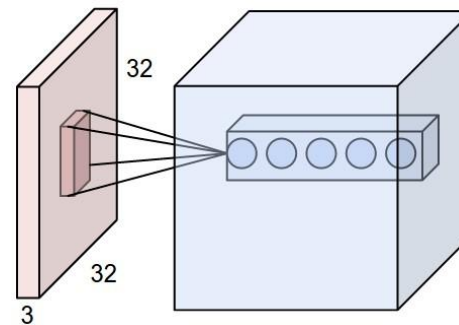
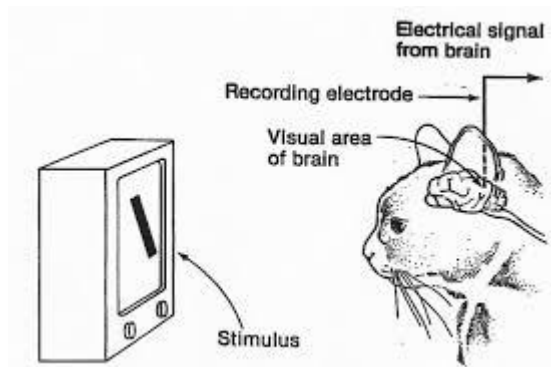
Hubel & Wiesel, 1968



ConvNet: Model



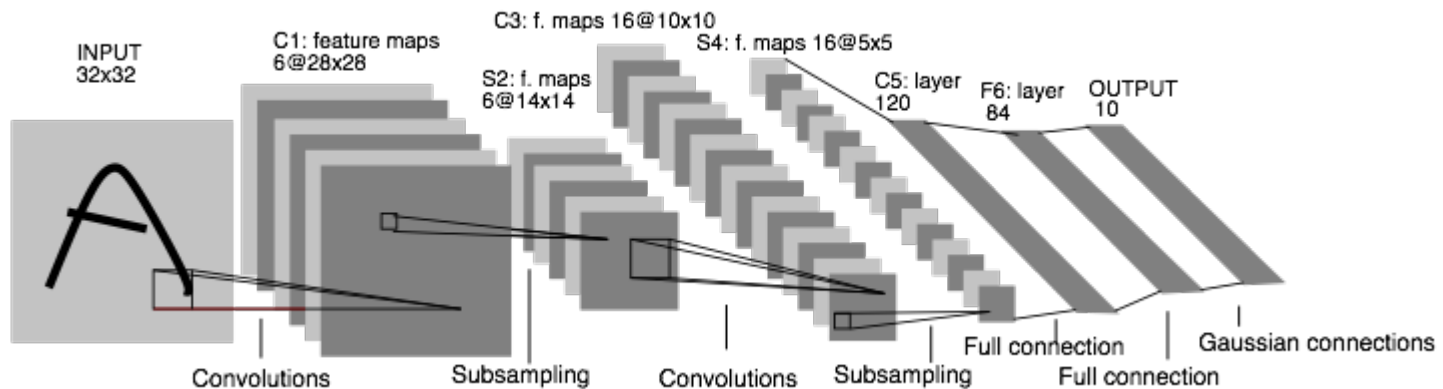
- Two key concepts:
 - Local receptive fields
 - Each unit in a neuron receives input from a small, localized region of the input image or the previous layer
 - Neurons then extract basic features such as corners, edges, and end points and combine them with subsequent layers to extract higher level features
 - Weight-sharing
 - Elementary features across different image locations share the same weights
 - Reduces the number of learnable parameters and makes it possible to train them with fewer examples.
 - The forward pass of the shared weights along with the input image form the **convolution** that gives the model its name.



ConvNet: Model

CNNs receive a raw pixel image as an input and perform a series of transformations through hidden layers.

- The **Convolution** layer performs a directed grid search across the image to detect features and construct a feature map that is output to the next layer.
- The **Pooling** layer performs downsampling to reduce the resolution of the feature map and reduce the output sensitivity to shifts and distortions.
- Finally, the **Fully-Connected** or output layer reduces the image to a single vector of class scores.



ConvNet: Experiment

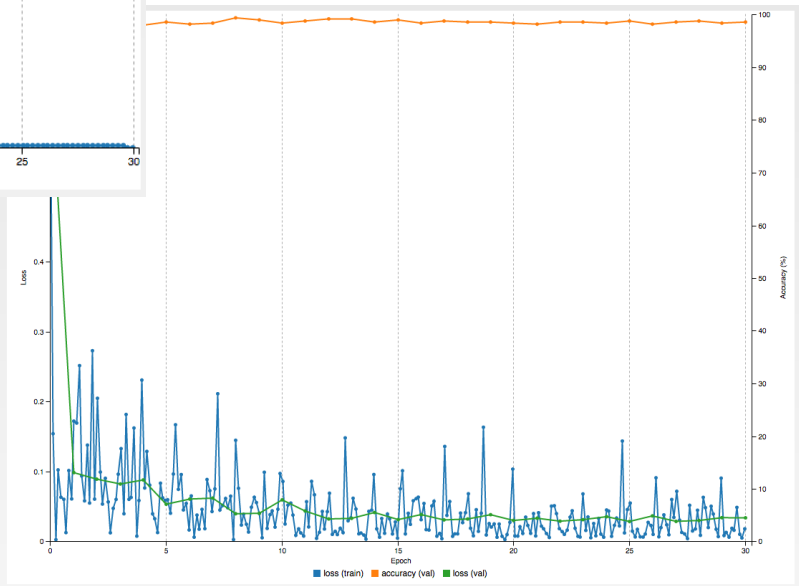
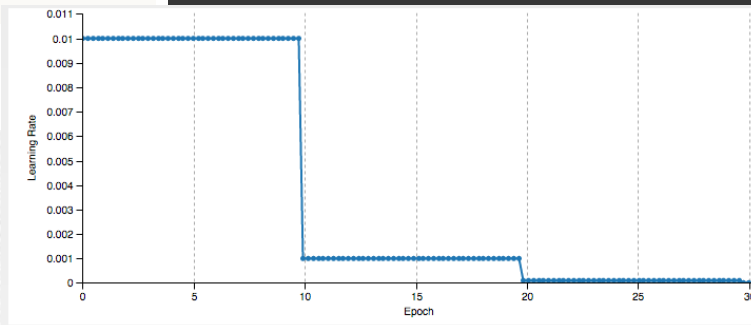
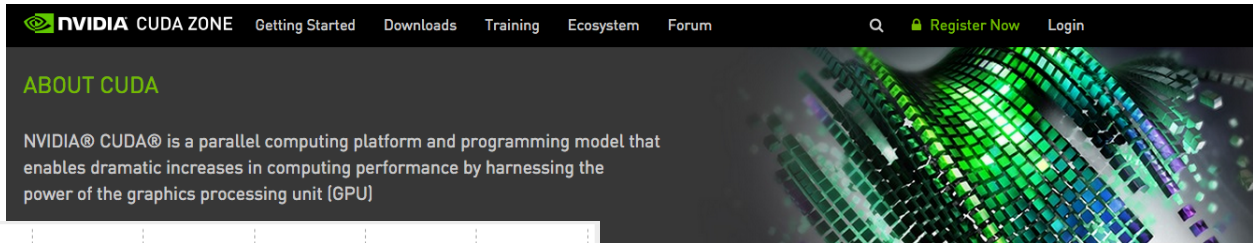
- Classification experiment of “Car” versus “Non-Car” images

Caffe

Deep learning framework
by the **BVLC**



OPENCV
(OPEN SOURCE
COMPUTER VISION)



/home/coyote/test-cars-final/building1.jpg			test	60.59%		cars train	39.41%
/home/coyote/test-cars-final/building2.jpg			test	93.8%		cars train	6.2%
/home/coyote/test-cars-final/building3.jpg			test	97.39%		cars train	2.61%
/home/coyote/test-cars-final/building4.jpg			test	79.26%		cars train	20.74%
/home/coyote/test-cars-final/cablecar.jpg			cars train	99.19%		test	0.81%
/home/coyote/test-cars-final/carobfuscated.jpg			cars train	99.97%		test	0.03%
/home/coyote/test-cars-final/carsrear.jpg			test	95.8%		cars train	4.2%
/home/coyote/test-cars-final/clouds.jpg			test	86.83%		cars train	13.17%
/home/coyote/test-cars-final/clouds1.jpg			cars train	99.32%		test	0.68%
/home/coyote/test-cars-final/crashedracer.jpg			cars train	99.54%		test	0.46%
/home/coyote/test-cars-final/doubledeckerbus.jpg			cars train	99.86%		test	0.14%
/home/coyote/test-cars-final/horsecarriage.jpg			cars train	92.73%		test	7.27%
/home/coyote/test-cars-final/militaryvehicle.jpg			cars train	100.0%		test	0.0%
/home/coyote/test-cars-final/parkinglotcars.jpg			test	94.92%		cars train	5.08%
/home/coyote/test-cars-final/parkinglotcarsfront.jpg			cars train	88.88%		test	11.12%
/home/coyote/test-cars-final/parkinglotcarsrear.jpg			test	85.54%		cars train	34.46%
/home/coyote/test-cars-final/sidewalk.jpg			test	60.59%		cars train	39.41%
/home/coyote/test-cars-final/smallcar1.jpg			test	93.8%		cars train	6.2%
/home/coyote/test-cars-final/smallcar2.jpg			test	97.39%		cars train	2.61%
/home/coyote/test-cars-final/streetcar.jpg			test	79.26%		cars train	20.74%
/home/coyote/test-cars-final/tank1.jpg			cars train	99.19%		test	0.81%
/home/coyote/test-cars-final/traincar.jpg			cars train	99.97%		test	0.03%
/home/coyote/test-cars-final/tree.jpg			test	95.8%		cars train	4.2%
/home/coyote/test-cars-final/twosmallcars.jpg			test	86.83%		cars train	13.17%
/home/coyote/test-cars-final/wheel1.jpg			cars train	99.32%		test	0.68%

ConvNet: Experiment

	test 100.0%	train 0.0%
	test 100.0%	train 0.0%
	test 100.0%	train 0.0%
	test 100.0%	train 0.0%
	test 99.99%	train 0.01%
	test 100.0%	train 0.0%
	test 99.7%	train 0.3%
	test 100.0%	train 0.0%
	test 100.0%	train 0.0%
	test 99.77%	train 0.23%



	test 100.0%	train 0.0%
	test 100.0%	train 0.0%
	test 100.0%	train 0.0%
	test 100.0%	train 0.0%
	test 99.99%	train 0.01%
	test 100.0%	train 0.0%
	test 99.97%	train 0.03%
	test 100.0%	train 0.0%
	test 100.0%	train 0.0%
	test 93.08%	train 6.92%

	test 99.93%	train 0.07%
	test 99.97%	train 0.03%
	test 100.0%	train 0.0%
	test 99.99%	train 0.01%
	test 99.96%	train 0.04%
	test 99.96%	train 0.04%
	test 99.91%	train 0.09%
	test 100.0%	train 0.0%
	test 99.98%	train 0.02%
	test 99.61%	train 0.39%



ConvNet: Experiment

TP



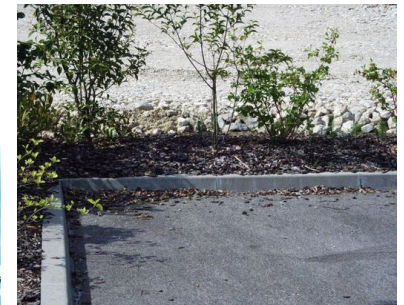
FN



FP



TN



Discussion

- **Performance/Predictive capacity**
 - **Training time**
 - **Limitations or challenges**
 - **Applications**
-