

PyImageSearch Gurus

A **course** and **community** designed to take you from computer vision *beginner* to *expert*. **Guaranteed**



The PyImageSearch Gurus course covers **13 modules** broken out into **168 lessons**, with other **2,161 pages** of content — this syllabus lists *each* and *every* lesson inside the course. To check out the syllabus, *just move to the next slide*.

Computer Vision & Image Processing Basics

1.1 Loading, displaying, and saving images

1.2 Image basics

1.3 Drawing

1.4 Basic image processing

1.4.1 Translation

1.4.2 Rotation

1.4.3 Resizing

1.4.4 Flipping

1.4.5 Cropping

1.4.6 Image arithmetic



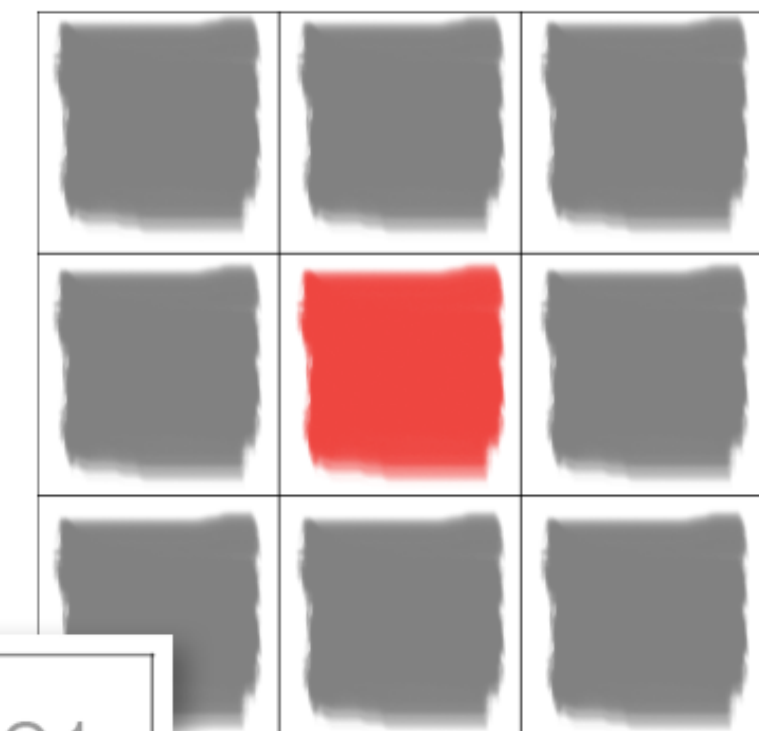
Computer Vision & Image Processing Basics (cont.)

- 1.4.7 Bitwise operations
- 1.4.8 Masking
- 1.4.9 Splitting and merging channels
- 1.5 Kernels
- 1.6 Morphological operations
- 1.7 Smoothing and blurring
- 1.8 Lighting and color spaces
- 1.9 Thresholding
- 1.10 Gradients and edge detection
 - 1.10.1 Gradients

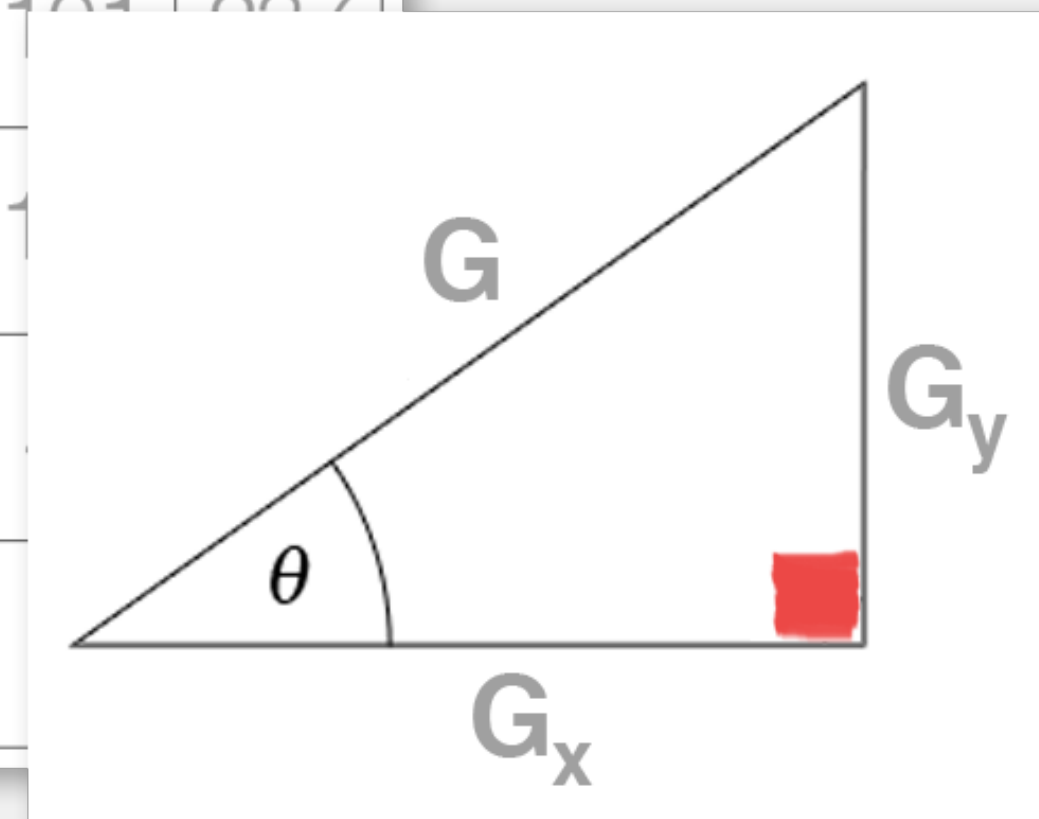
4-neighborhood



8-neighborhood

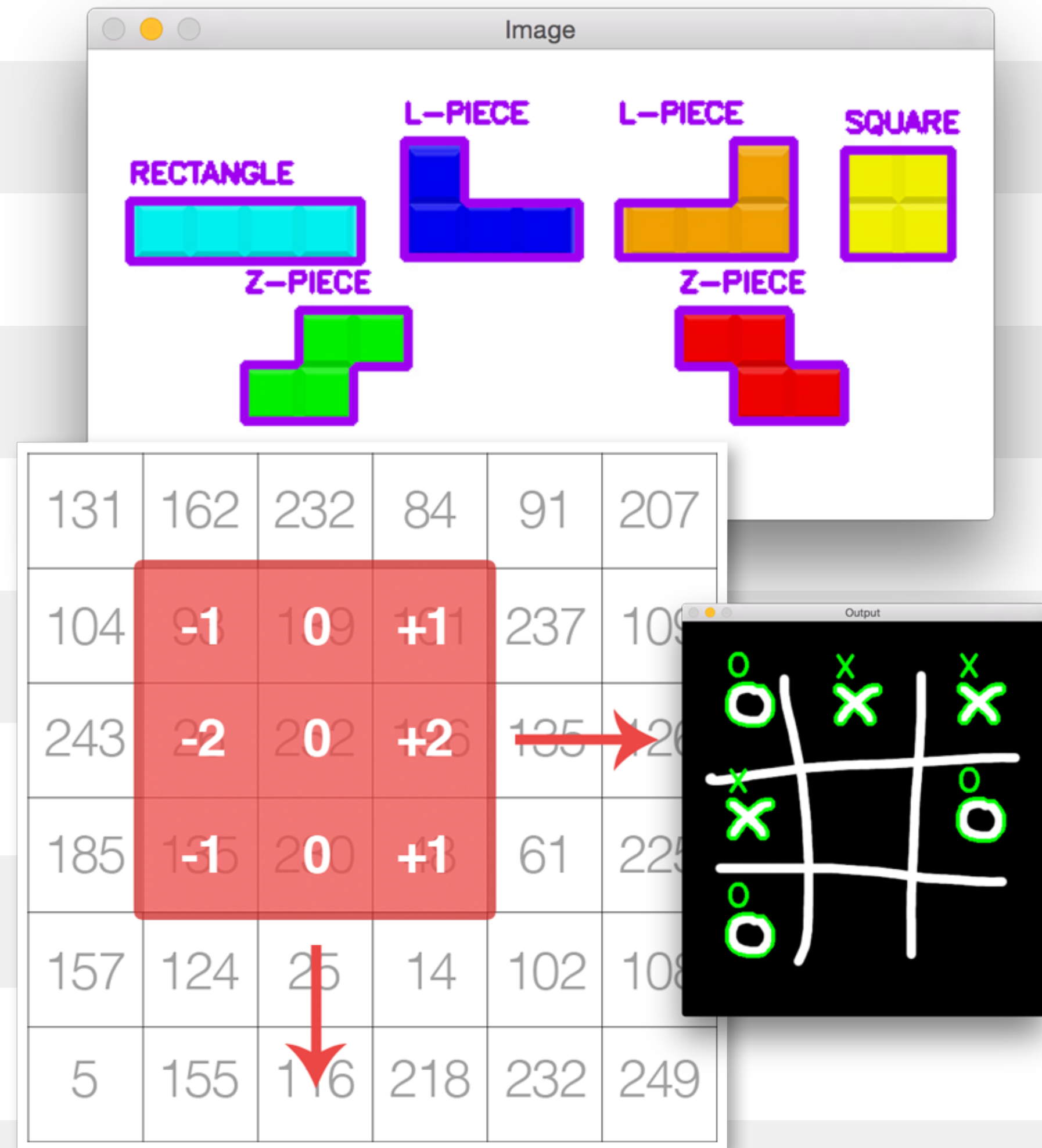


131	162	232	84	91
104	93	139	101	887
243	26	252	1	
185	135	230		
157	124	25		



Computer Vision & Image Processing Basics (cont.)

- 1.10.2 Edge detection
- 1.11 Contours
 - 1.11.1 Finding and drawing contours
 - 1.11.2 Simple contour properties
 - 1.11.3 Advanced contour properties
 - 1.11.4 Contour approximation
 - 1.11.5 Sorting contours
- 1.12 Histograms
- 1.13 Connected-component labeling



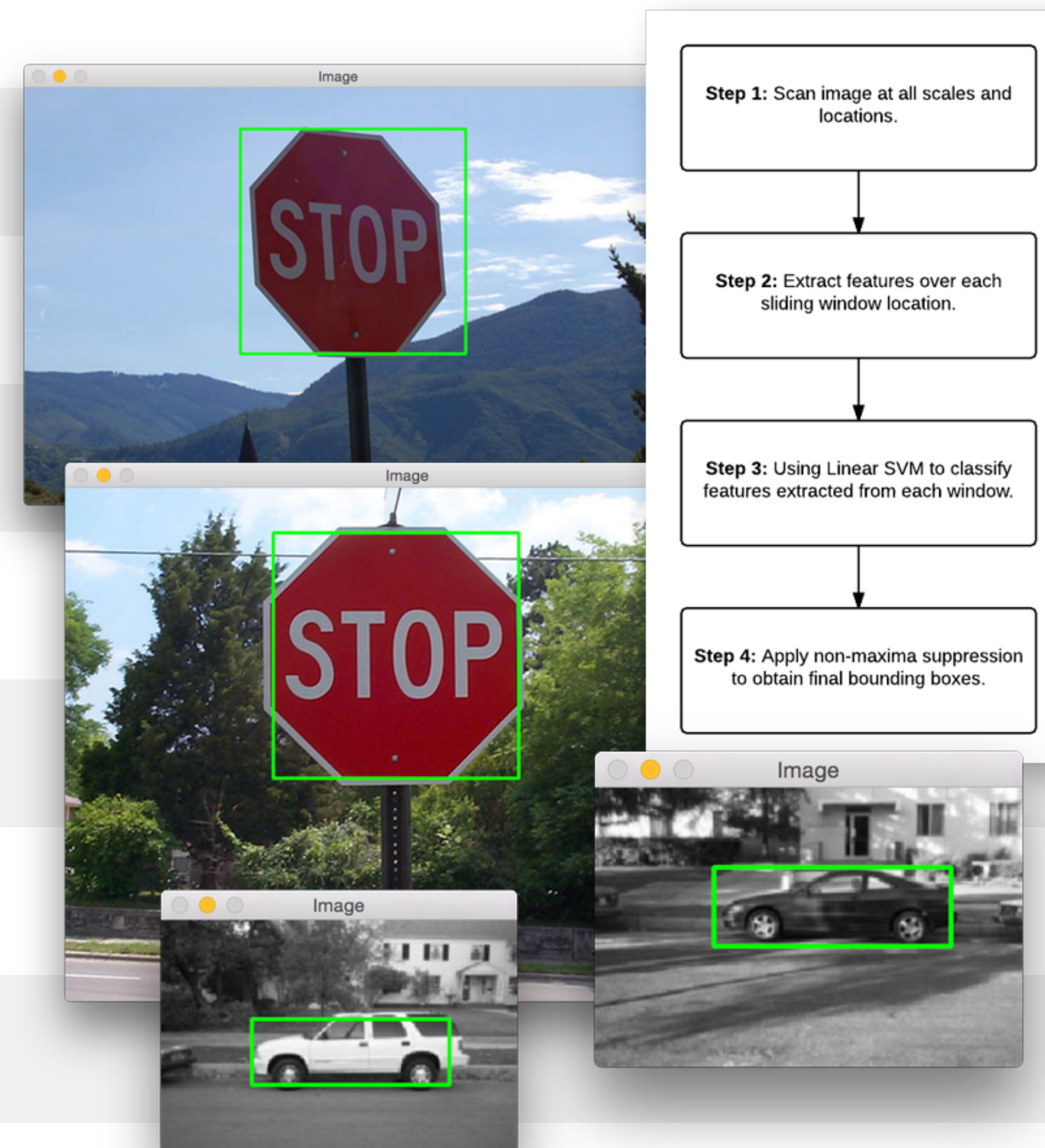
Train Your Own Custom Object Detectors

- 2.1 What are object detectors?
 - 2.1.1 An introduction to object detection
 - 2.1.2 Template matching
- 2.2 Object detection: The easy way
 - 2.2.1 How to install dlib
 - 2.2.2 Object detection made easy
- 2.3 Sliding windows and image pyramids
 - 2.3.1 Image pyramids
 - 2.3.2 Sliding windows



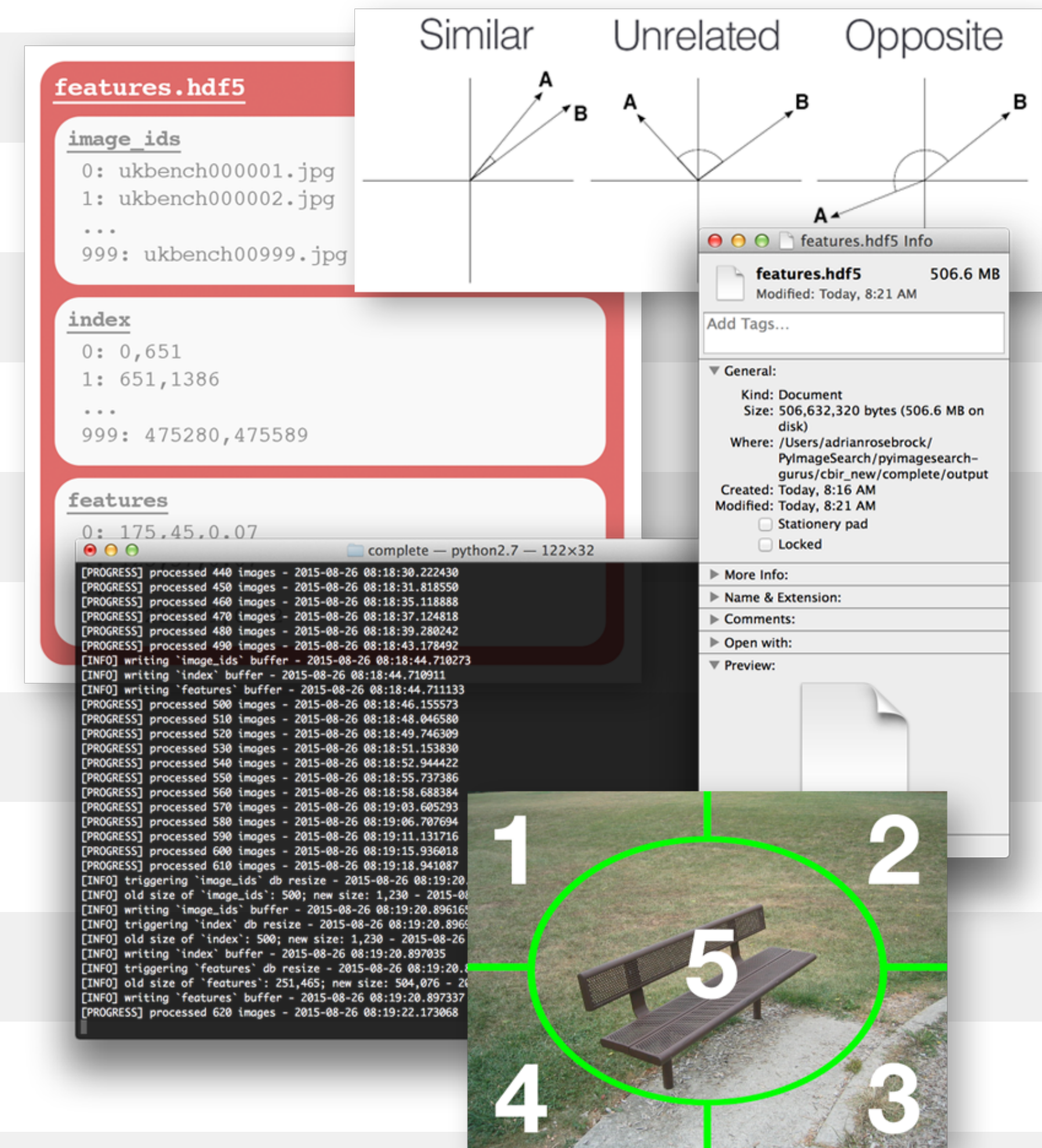
Train Your Own Custom Object Detectors (cont.)

- 2.4 The 6-step framework
- 2.5 Preparing your experiment and training data
- 2.6 Constructing your HOG descriptor
- 2.7 The initial training phase
- 2.8 Non-maxima suppression
- 2.9 Hard-negative mining
- 2.10 Re-training and running your classifier
- 2.11 Training your custom object detector
- 2.12 Tips on training your own object detectors



Content-Based Image Retrieval

- 3.1 What is Content-Based Image Retrieval?
- 3.2 Your first image search engine
- 3.3 The 4 steps of building any image search engine
 - 3.3.1 Defining your image descriptor
 - 3.3.2 Feature extraction and indexing
 - 3.3.3 Defining your similarity metric
 - 3.3.4 Searching
- 3.4 The bag of (visual) words model
- 3.5 Extracting keypoints and local invariant descriptors
- 3.6 Clustering features to form a codebook
- 3.7 Visualizing words in a codebook
- 3.8 Vector quantization



Content-Based Image Retrieval (cont.)

- 3.8.1 From multiple features to a single histogram
- 3.8.2 Forming a BOVW
- 3.9 Inverted indexes and searching
 - 3.9.1 What is Redis?
 - 3.9.2 Building an inverted index
 - 3.9.3 Performing a search
- 3.10 Spatial verification
- 3.11 Tf-idf weighting
- 3.12 Spatial verification
 - 3.12.1 Implementing spatial verification
 - 3.12.2 Searching with spatial verification
- 3.13.3 Evaluating search with spatial verification

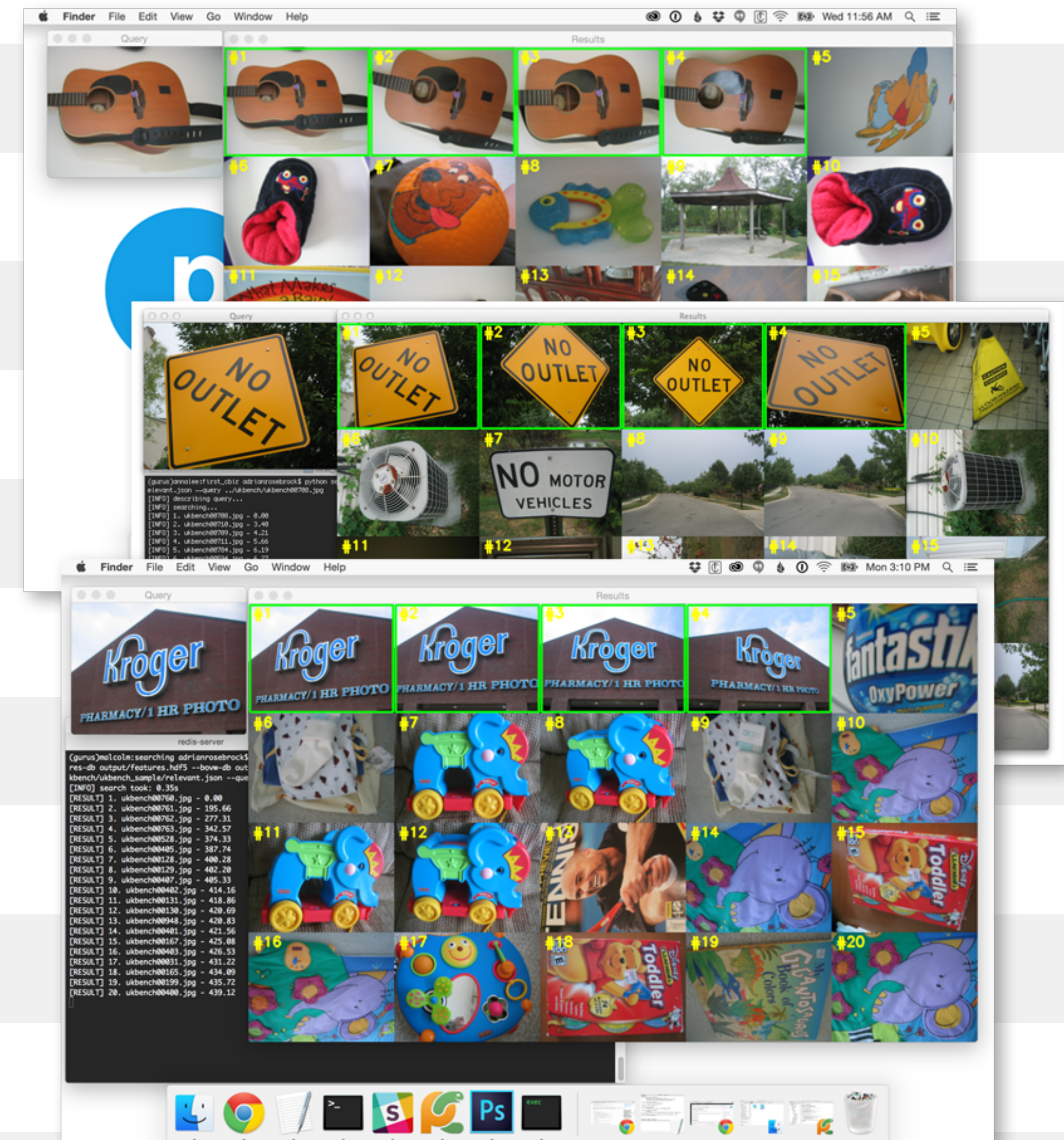


Image Classification + Machine Learning

4.1 A high level overview of image classification

4.1.1 What is image classification?

4.1.2 Types of learning

4.2 The image classification pipeline

4.3 k-Nearest Neighbor classification

4.4 Common machine learning algorithms

4.4.1 Logistic regression

4.4.2 Support Vector Machines

4.4.3 Decision trees

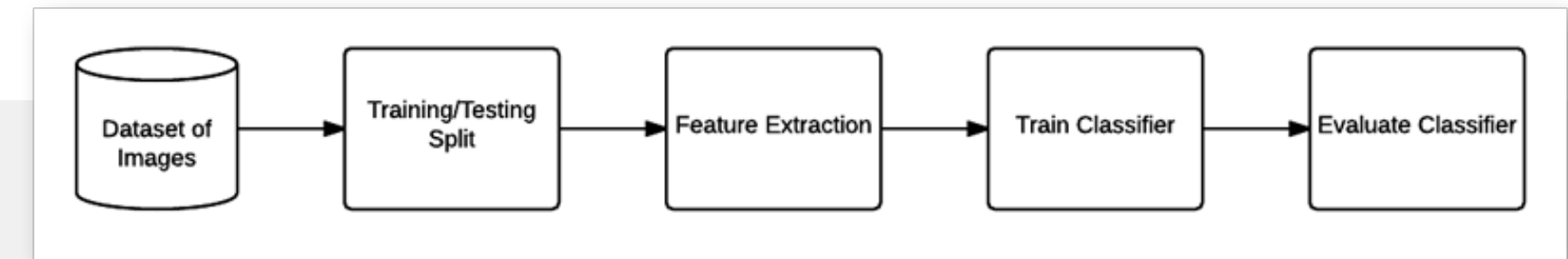


Image Classification + Machine Learning (cont.)

4.4.4

Random forests

4.5

k-means clustering

4.6

Bag of visual words for classification

4.7.1

Image pyramids for classification

4.7.2

PBOW

4.8

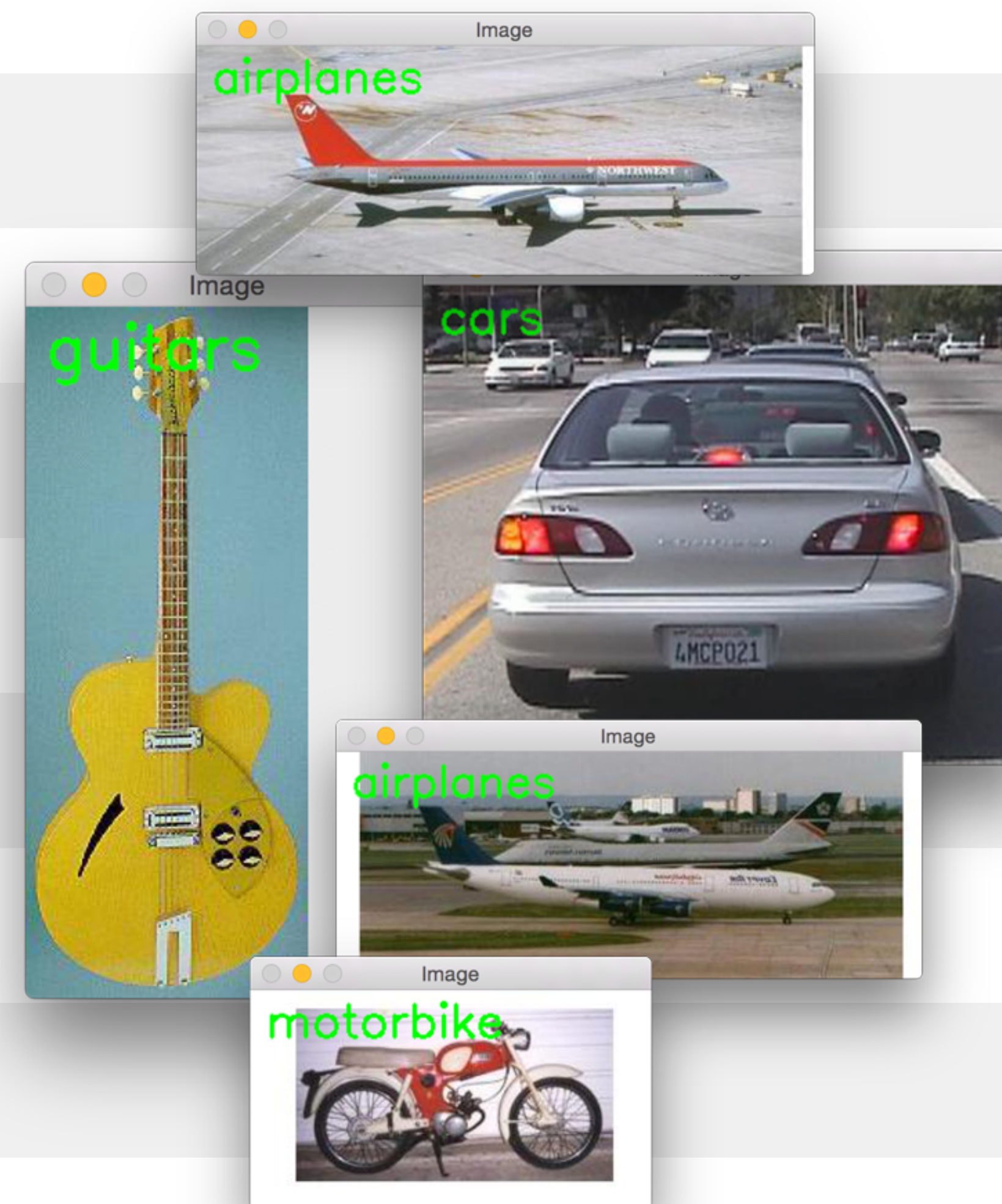
Image classification example: Flowers-17

4.9

Image classification example: CALTECH-101

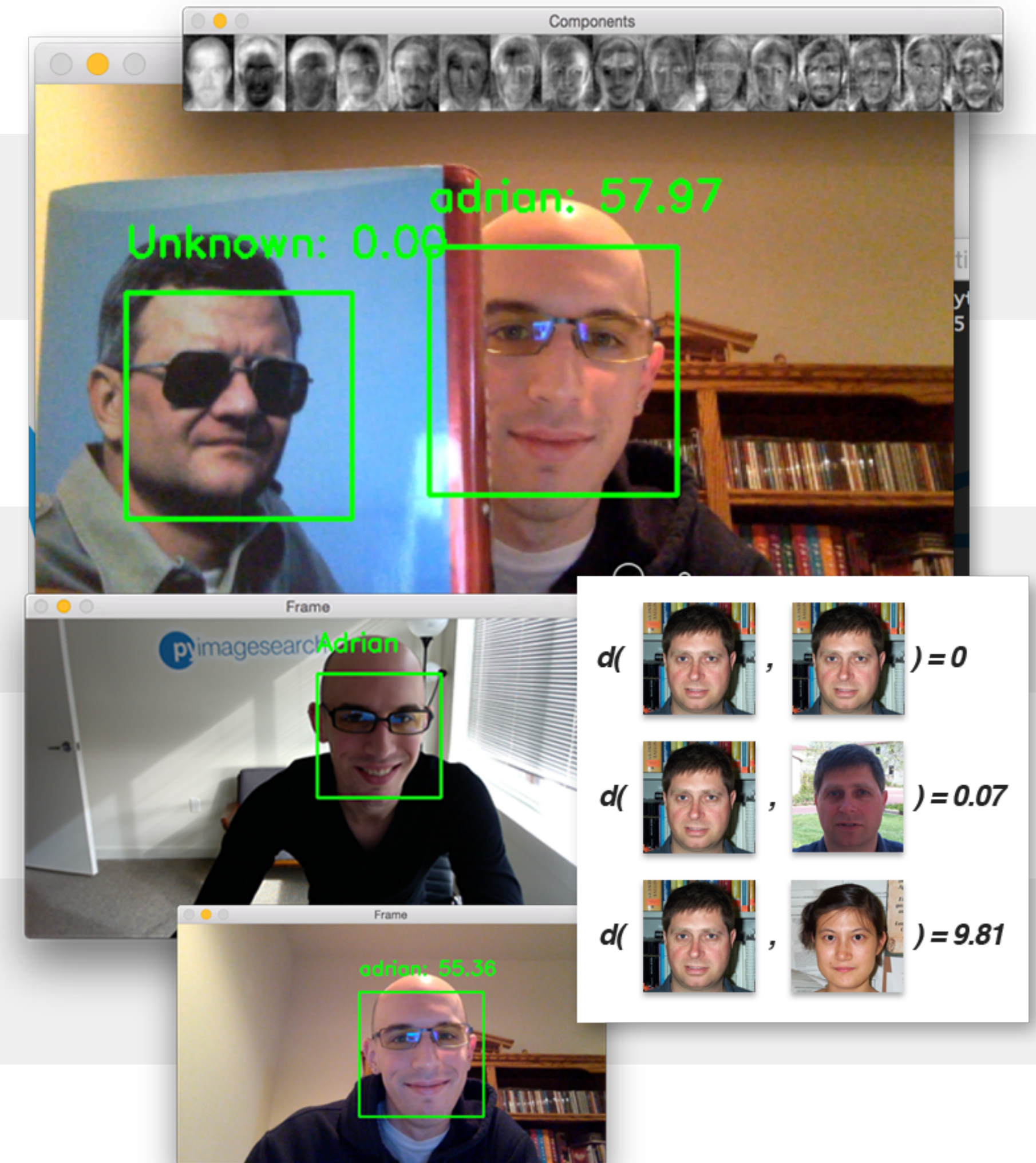
4.10

Tips on training your own image classifiers



Face Recognition

- 5.1 What is face recognition?
- 5.2 LBPs for face recognition
- 5.3 The Eigenfaces algorithm
- 5.4 Preparing and pre-processing your own face data
- 5.5 The complete face recognition pipeline



Automatic License Plate Recognition (ANPR)

- 6.1 What is ANPR?
- 6.2 The problem with ANPR datasets
- 6.3 Localizing license plates in images
- 6.4 Segmenting characters from the license plate
- 6.5 Scissoring the license plate characters
- 6.6 Our first try at recognizing license plate characters
- 6.7 Gathering our own license plate characters
- 6.8 Improving our license plate classifier
- 6.9 Tips on classifying your own license plates

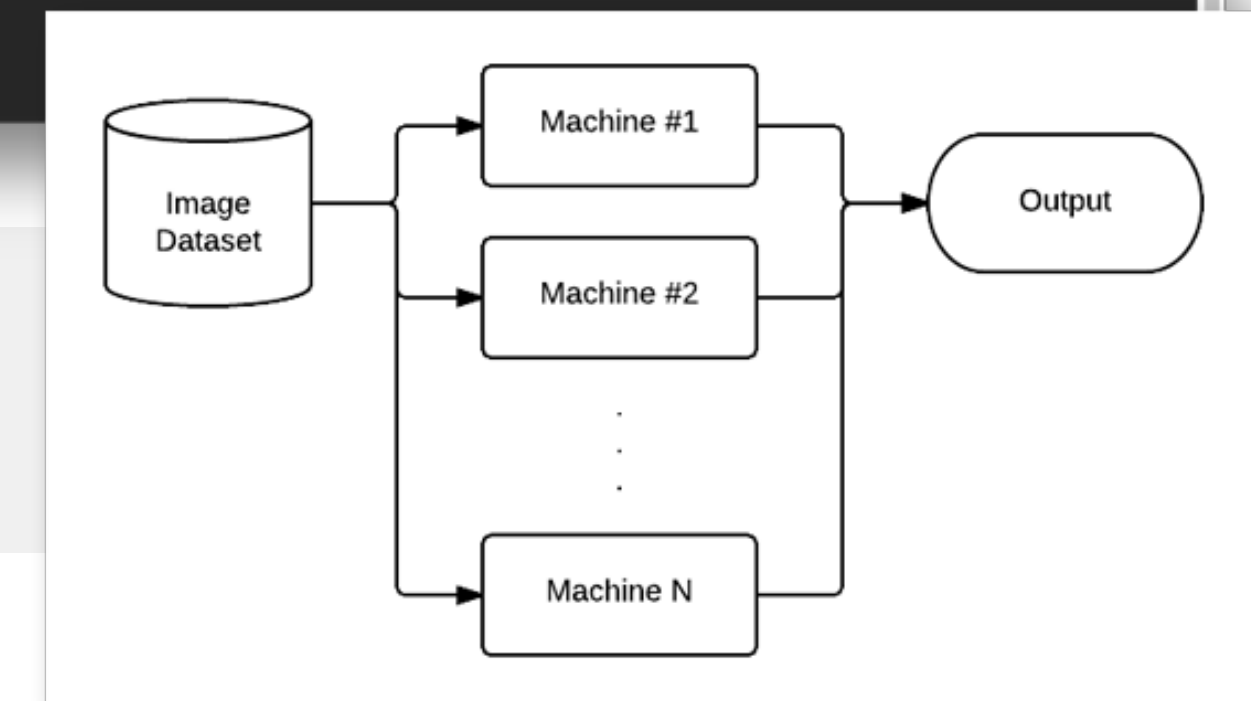


Hadoop + Big Data for Computer Vision

- 7.1 Introduction to Hadoop and MapReduce
- 7.2 Setting up Hadoop on your machine
- 7.3 Preparing your images for use on HDFS
- 7.4 Running computer vision jobs on MapReduce
- 7.5 High-throughput face detection
- 7.6 High-throughput feature extraction

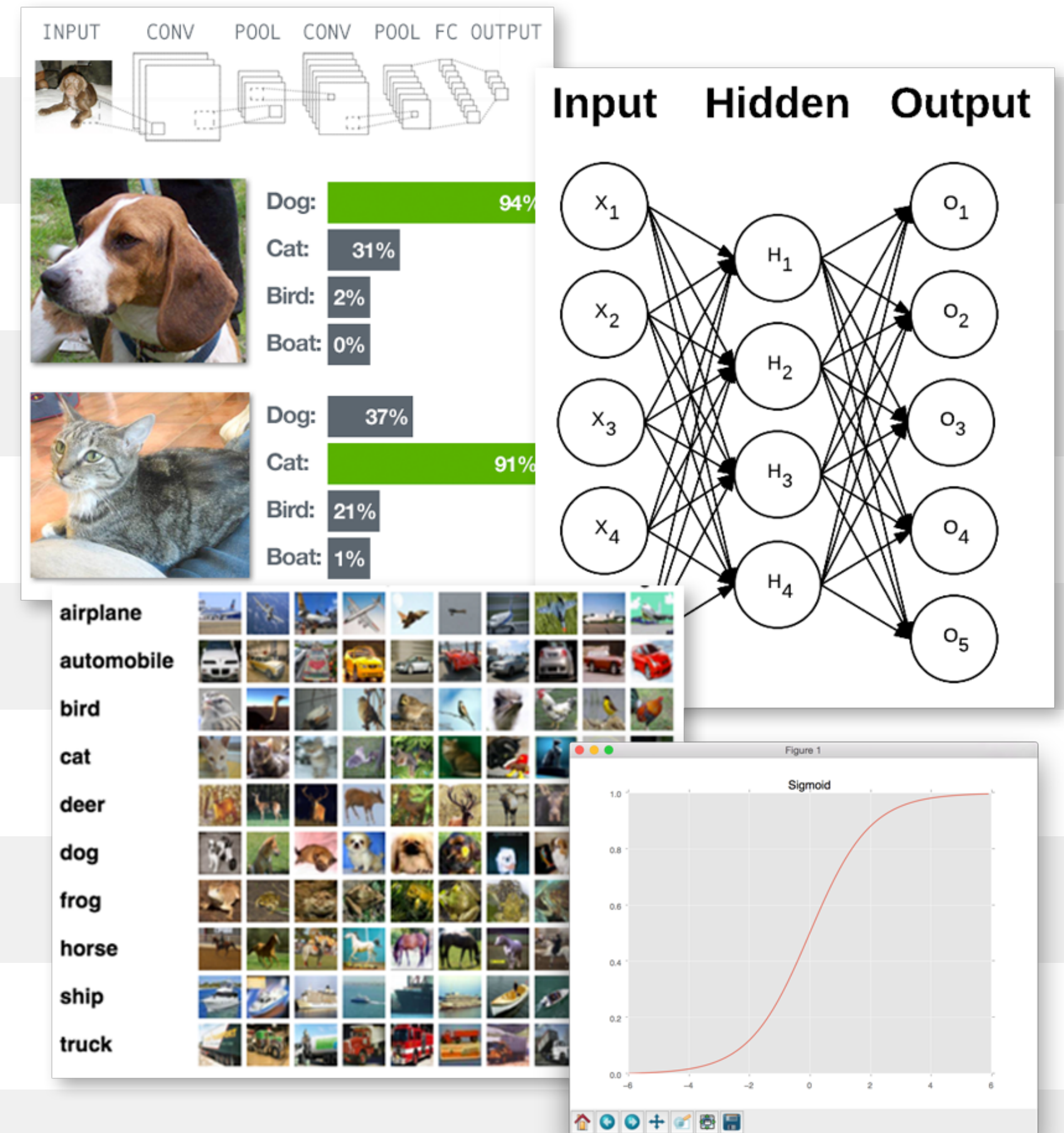


```
malcolm:hadoop adrianrosebrock$ bin/hdfs dfs -ls /user/guru/faces/output
16/02/22 08:24:22 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 30 items
-rw-r--r-- 1 adrianrosebrock supergroup 0 2016-02-22 08:21 /user/guru/faces/output/_SUCCESS
-rw-r--r-- 1 adrianrosebrock supergroup 25100 2016-02-22 08:11 /user/guru/faces/output/part-00000
-rw-r--r-- 1 adrianrosebrock supergroup 24350 2016-02-22 08:12 /user/guru/faces/output/part-00001
-rw-r--r-- 1 adrianrosebrock supergroup 20454 2016-02-22 08:12 /user/guru/faces/output/part-00002
-rw-r--r-- 1 adrianrosebrock supergroup 26994 2016-02-22 08:13 /user/guru/faces/output/part-00003
-rw-r--r-- 1 adrianrosebrock supergroup 28377 2016-02-22 08:13 /user/guru/faces/output/part-00004
-rw-r--r-- 1 adrianrosebrock supergroup 19936 2016-02-22 08:13 /user/guru/faces/output/part-00005
-rw-r--r-- 1 adrianrosebrock supergroup 24908 2016-02-22 08:14 /user/guru/faces/output/part-00006
-rw-r--r-- 1 adrianrosebrock supergroup 33460 2016-02-22 08:14 /user/guru/faces/output/part-00007
-rw-r--r-- 1 adrianrosebrock supergroup 25463 2016-02-22 08:14 /user/guru/faces/output/part-00008
-rw-r--r-- 1 adrianrosebrock supergroup 23264 2016-02-22 08:15 /user/guru/faces/output/part-00009
-rw-r--r-- 1 adrianrosebrock supergroup 23408 2016-02-22 08:15 /user/guru/faces/output/part-00010
-rw-r--r-- 1 adrianrosebrock supergroup 28139 2016-02-22 08:15 /user/guru/faces/output/part-00011
-rw-r--r-- 1 adrianrosebrock supergroup 18172 2016-02-22 08:16 /user/guru/faces/output/part-00012
-rw-r--r-- 1 adrianrosebrock supergroup 28017 2016-02-22 08:16 /user/guru/faces/output/part-00013
-rw-r--r-- 1 adrianrosebrock supergroup 21776 2016-02-22 08:17 /user/guru/faces/output/part-00014
-rw-r--r-- 1 adrianrosebrock supergroup 17619 2016-02-22 08:17 /user/guru/faces/output/part-00015
-rw-r--r-- 1 adrianrosebrock supergroup 22892 2016-02-22 08:17 /user/guru/faces/output/part-00016
-rw-r--r-- 1 adrianrosebrock supergroup 26794 2016-02-22 08:18 /user/guru/faces/output/part-00017
-rw-r--r-- 1 adrianrosebrock supergroup 21114 2016-02-22 08:18 /user/guru/faces/output/part-00018
-rw-r--r-- 1 adrianrosebrock supergroup 25461 2016-02-22 08:18 /user/guru/faces/output/part-00019
-rw-r--r-- 1 adrianrosebrock supergroup 25616 2016-02-22 08:19 /user/guru/faces/output/part-00020
-rw-r--r-- 1 adrianrosebrock supergroup 34896 2016-02-22 08:19 /user/guru/faces/output/part-00021
-rw-r--r-- 1 adrianrosebrock supergroup 25980 2016-02-22 08:19 /user/guru/faces/output/part-00022
-rw-r--r-- 1 adrianrosebrock supergroup 21152 2016-02-22 08:20 /user/guru/faces/output/part-00023
-rw-r--r-- 1 adrianrosebrock supergroup 23953 2016-02-22 08:20 /user/guru/faces/output/part-00024
-rw-r--r-- 1 adrianrosebrock supergroup 27284 2016-02-22 08:20 /user/guru/faces/output/part-00025
-rw-r--r-- 1 adrianrosebrock supergroup 22840 2016-02-22 08:21 /user/guru/faces/output/part-00026
-rw-r--r-- 1 adrianrosebrock supergroup 24616 2016-02-22 08:21 /user/guru/faces/output/part-00027
-rw-r--r-- 1 adrianrosebrock supergroup 14528 2016-02-22 08:21 /user/guru/faces/output/part-00028
malcolm:hadoop adrianrosebrock$
```



Deep Learning and Convolutional Neural Networks

- 8.1 Neural networks in a nutshell
 - 8.1.1 Introduction to neural networks
 - 8.1.2 The Perceptron algorithm
 - 8.1.3 Multi-layer networks
- 8.2 Introduction to deep learning
- 8.3 Setting up your deep learning environment
- 8.4 Deep Belief Networks
 - 8.4.1 Deep Belief Network basics
 - 8.4.2 Training a Deep Belief Network
- 8.5 Convolutional Neural Networks



Deep Learning and Convolutional Neural Networks (cont.)

8.5.1 A CNN primer

8.5.2 Training your first CNN

8.6 Implementing CNN architectures

8.6.1 LeNet

8.6.2 KarpathyNet

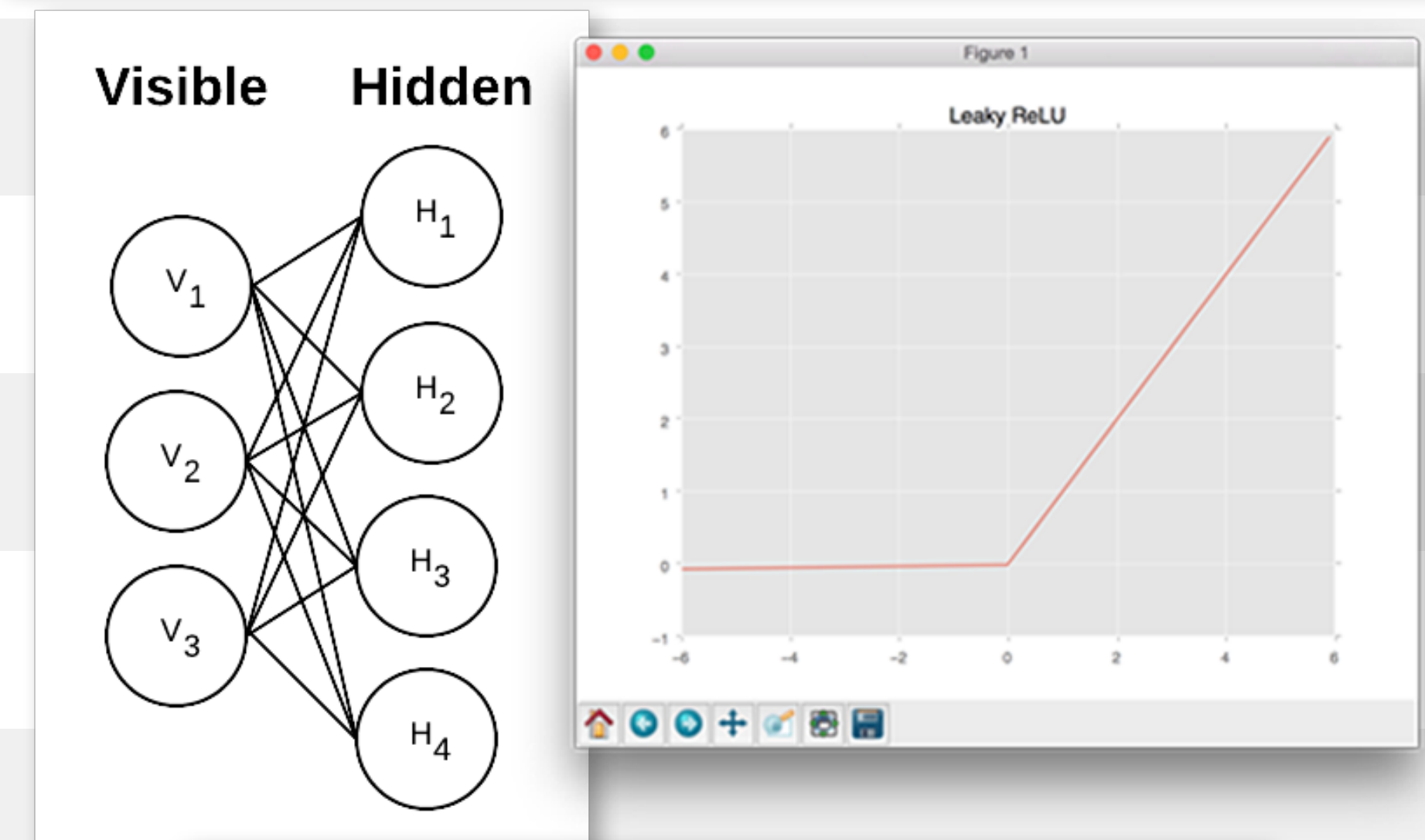
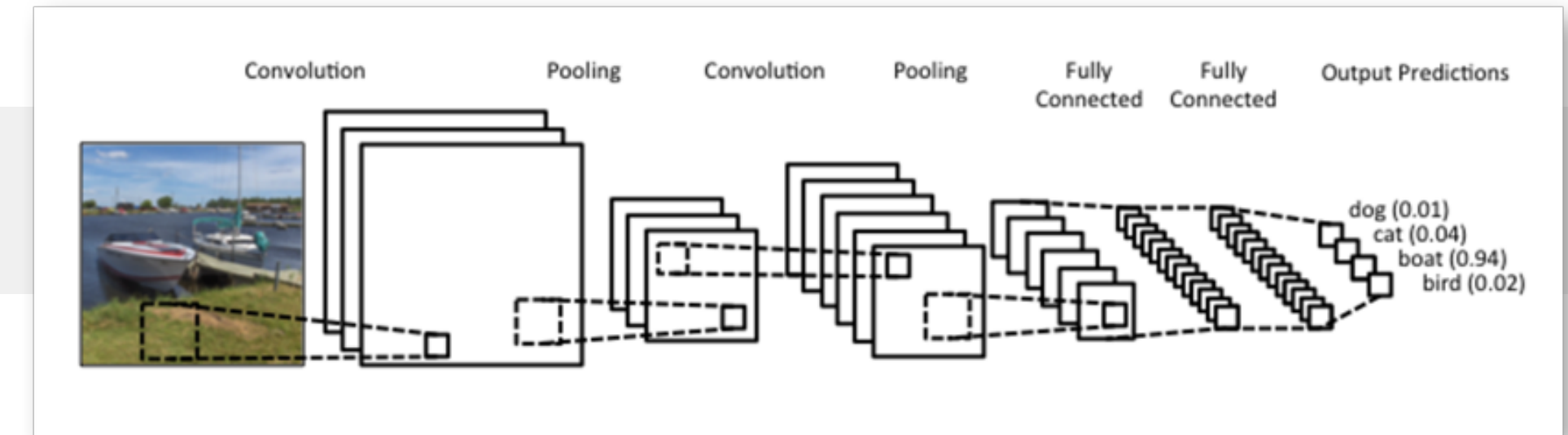
8.6.3 MiniVGGNet

8.6.4 Running a pre-trained network

8.7 The OverFeat framework

8.7.1 What is OverFeat?

8.7.2 OverFeat example: dogs and cats



Layer	1	2	3	4	5	6	7	Output 8
Stage	conv + max	conv + max	conv	conv	conv + max	full	full	full
# channels	96	256	512	1024	1024	3072	4096	1000
Filter size	11x11	5x5	3x3	3x3	3x3	-	-	-
Conv. stride	4x4	1x1	1x1	1x1	1x1	-	-	-
Pooling size	2x2	2x2	-	-	2x2	-	-	-
Pooling stride	2x2	2x2	-	-	2x2	-	-	-
Zero-Padding size	-	-	1x1x1x1	1x1x1x1	1x1x1x1	-	-	-
Spatial input size	231x231	24x24	12x12	12x12	12x12	6x6	1x1	1x1

Deep Learning and Convolutional Neural Networks (cont.)

8.7.3

OverFeat example: flower classification

8.7.4

OverFeat example: CALTECH-101

8.8

Working with Caffe

8.8.1

Making a dataset compatible with Caffe

8.8.2

The anatomy of a Caffe project

8.8.3

Training and evaluating a network with Caffe

8.9

Tips on training your own networks



Raspberry Pi Computer Vision Projects

- 9.1 Installing OpenCV on your Raspberry Pi
- 9.2 Setting up your Raspberry Pi Camera
- 9.3 Accessing the Raspberry Pi camera and video stream
- 9.4 Home surveillance and motion detection
- 9.5 Face recognition for security

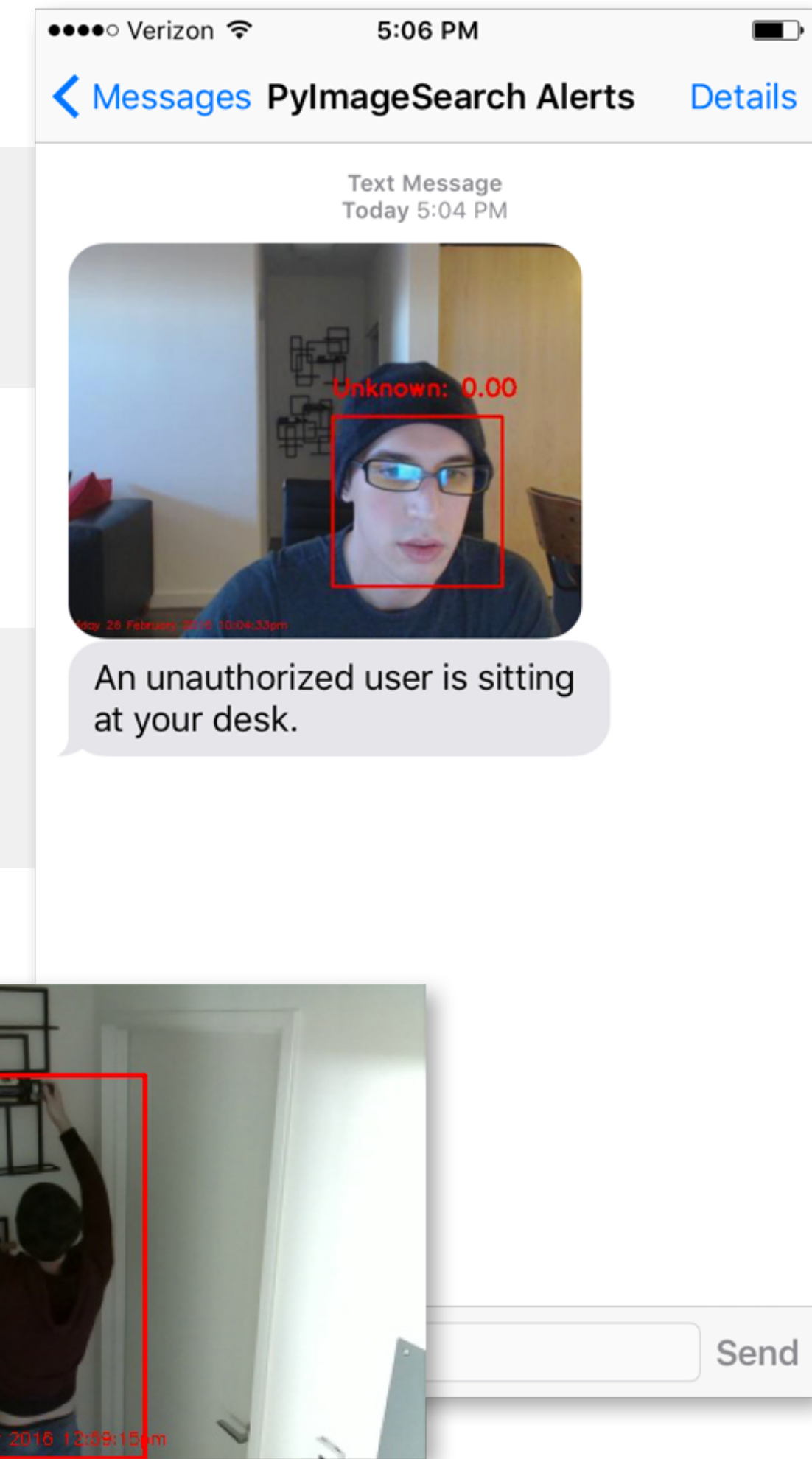


Image Descriptors

10.1 What are image descriptors and feature vectors?

10.2 Color channel statistics

10.3 Color histograms

10.4 Hu Moments

10.5 Zernike Moments

10.6 Haralick texture

10.7 Local Binary Patterns

10.8 Histogram of Oriented Gradients

10.9 Understanding local features

10.10 Keypoint detectors

10.10.1 FAST

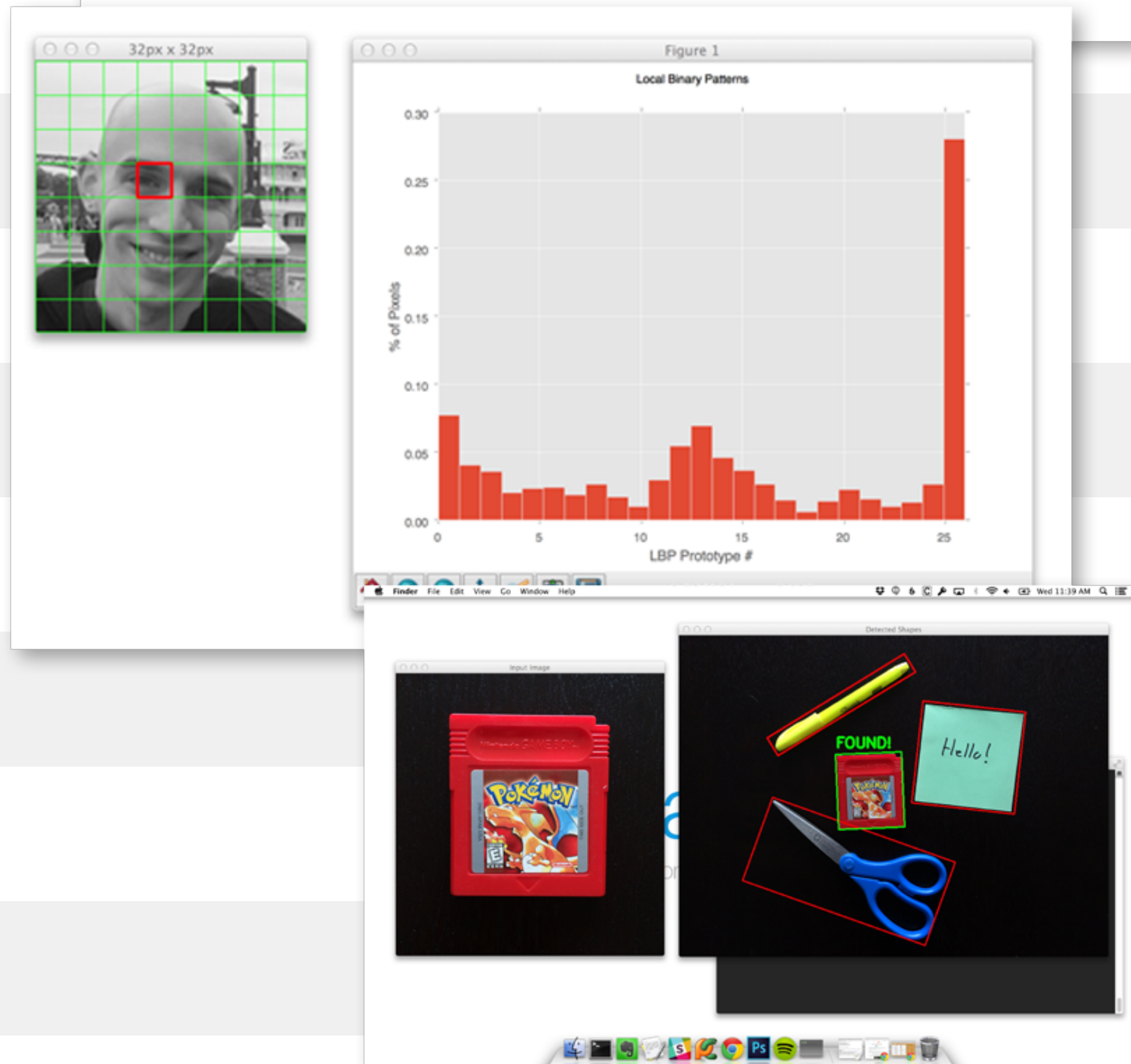
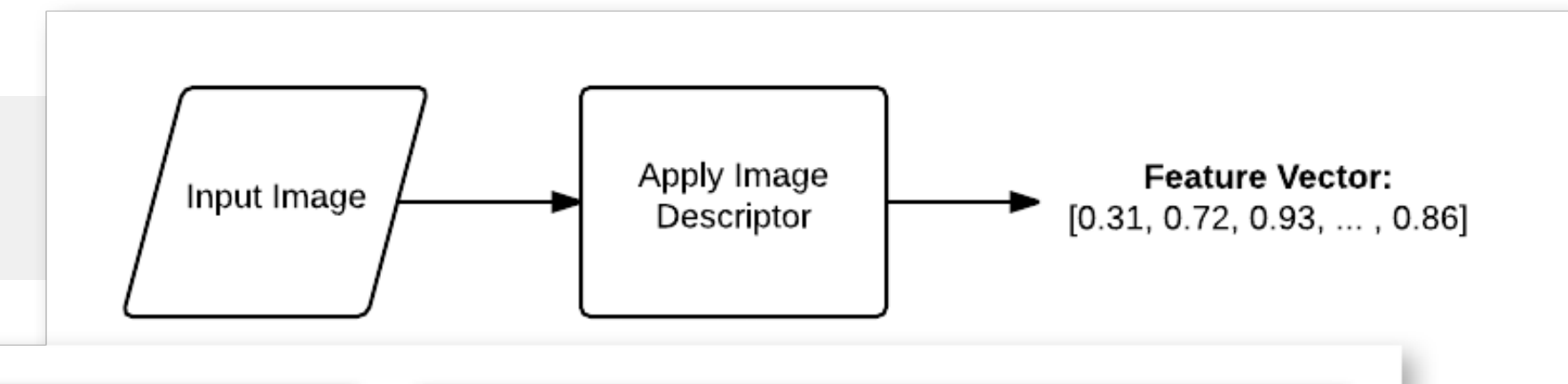


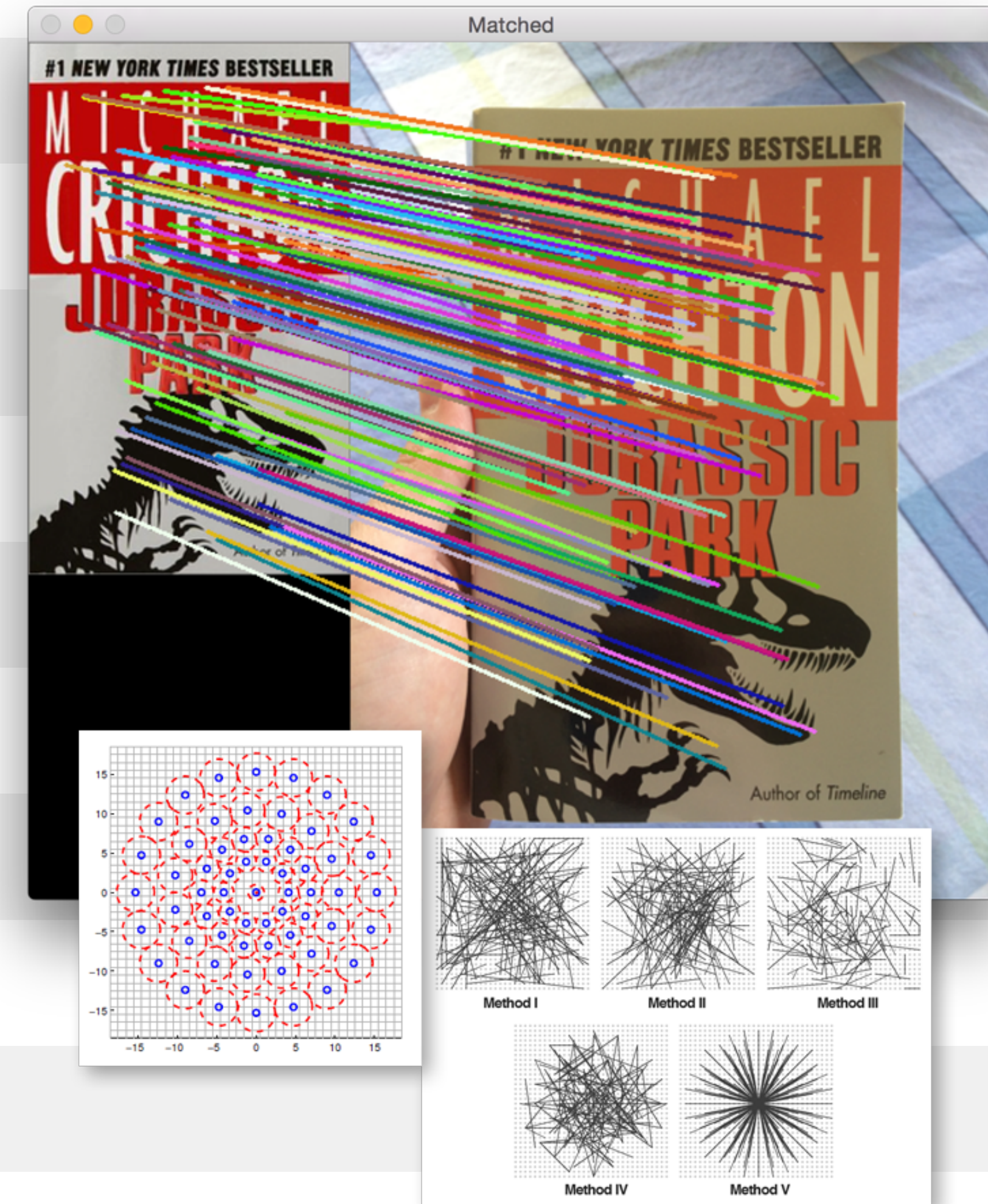
Image Descriptors (cont.)

10.10.2	Harris
10.10.3	GFTT
10.10.4	DoG
10.10.5	Fast Hessian
10.10.6	STAR
10.10.7	MSER
10.10.8	Dense
10.10.9	BRISK
10.10.10	ORB
10.11	Local invariant descriptors
10.10.2	Harris



Image Descriptors (cont.)

- 10.11.1 SIFT
- 10.11.2 RootSIFT
- 10.11.3 SURF
- 10.11.4 Real-valued feature extraction and matching
- 10.12 Binary Descriptors
 - 10.12.1 What are binary descriptors?
 - 10.12.2 BRIEF
 - 10.12.3 ORB (descriptor)
 - 10.12.4 BRISK (descriptor)
 - 10.12.5 FREAK
 - 10.12.6 Binary feature extraction and matching



Computer Vision Case Studies

11.1 Measuring distance from camera to object in image

11.2 Face detection in images

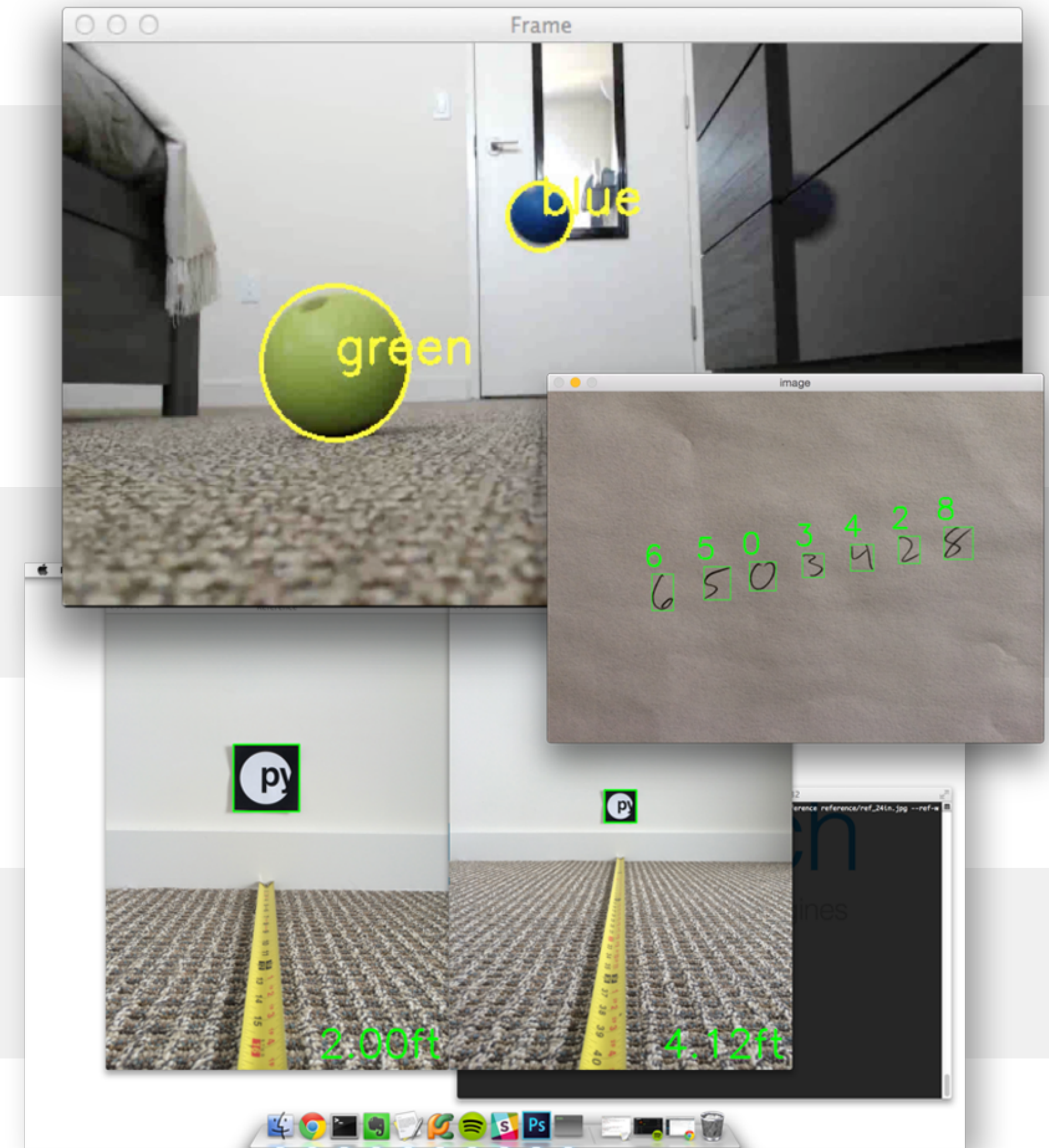
11.3 Face detection in video

11.4 Object tracking in video

11.5 Identifying the covers of books

11.6 Plant classification

11.7 Handwriting recognition



Building Computer Vision Apps for your Mobile Device

12.1

Introduction to PhoneGap

12.2

Overview of PhoneGap

12.3

PhoneGap environment setup

12.4

PhoneGap “Hello, World”

12.5

PhoneGap UI Setup

12.6

Capturing and uploading a photo with PhoneGap

12.7

Displaying face detection results

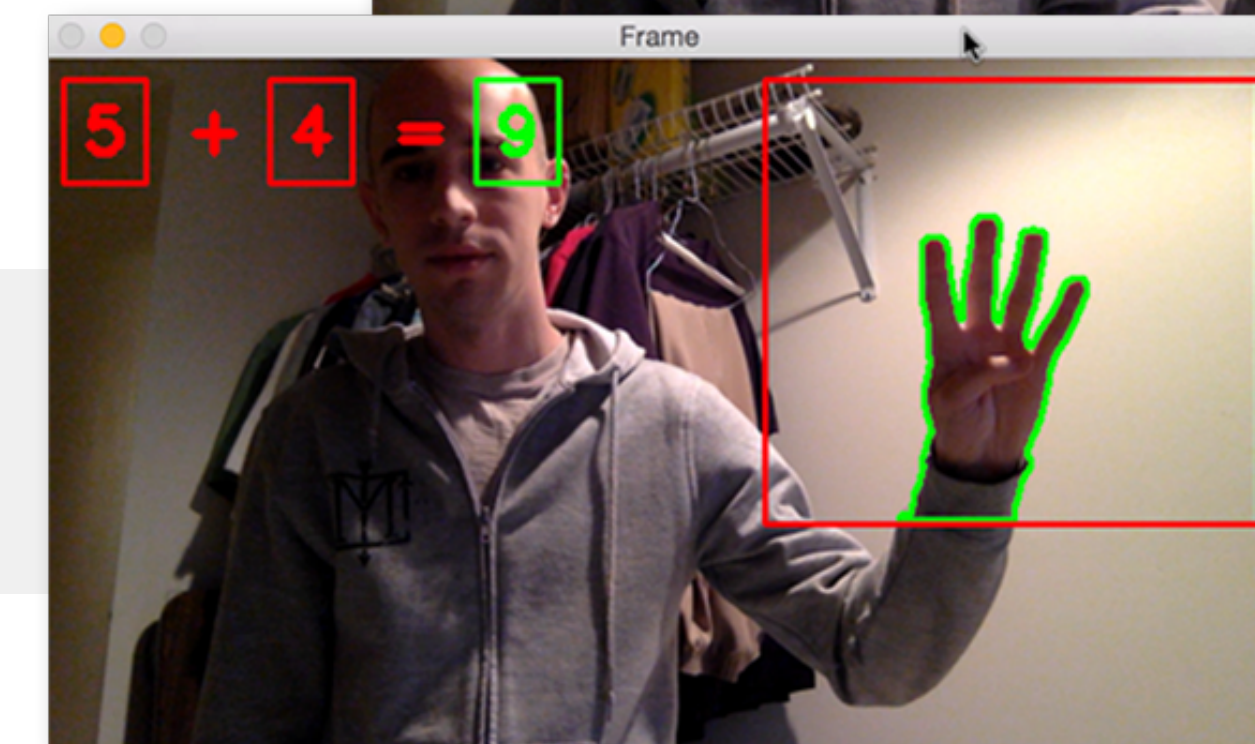
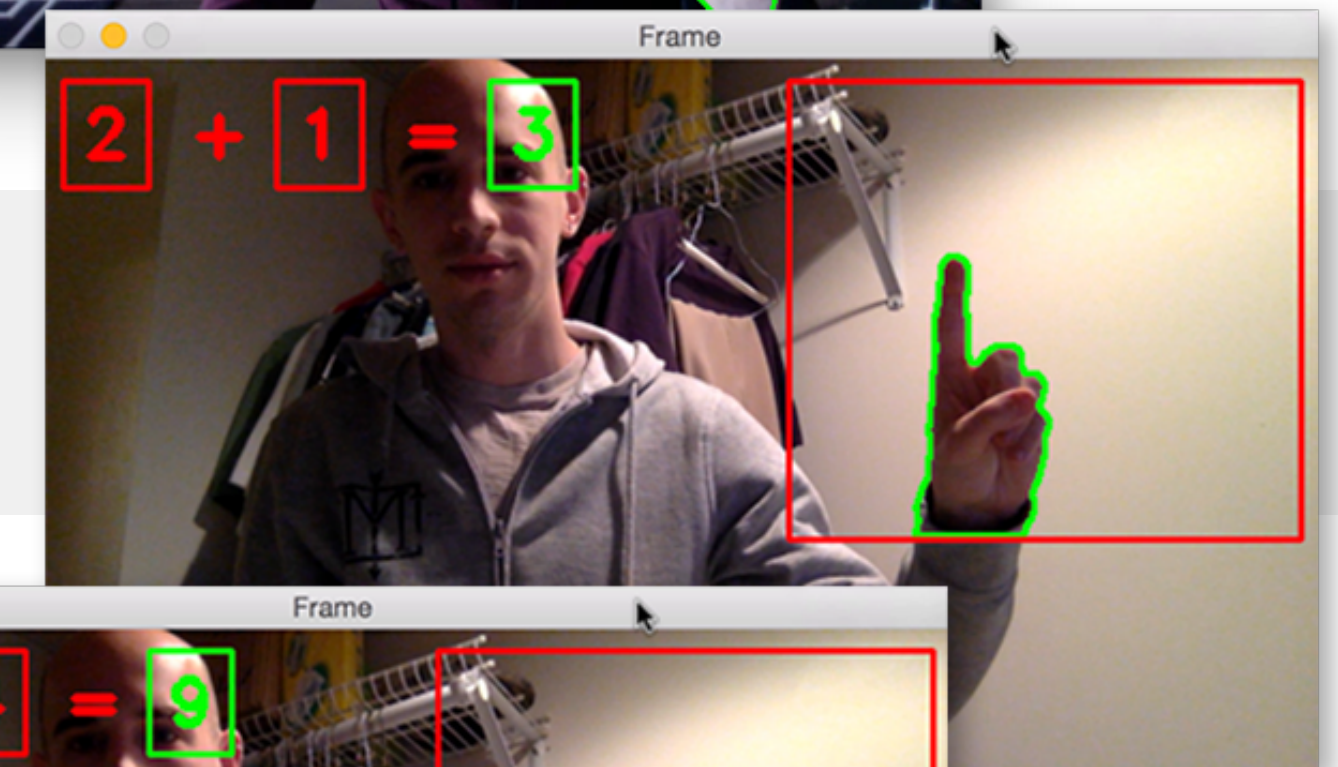


Hand Gesture Recognition

13.1 Introduction to hand gesture recognition

13.2 Hand, finger, and motion segmentation

13.3 Recognizing gestures



As you can see, PyImageSearch Gurus is the most *comprehensive, in-depth, and easy-to-follow* computer vision course online.

But don't take my word for it.

*“I cannot say it enough: **PyImageSearch Gurus** is **amazing**. I really enjoy myself and the way you set everything up (blog, course, community, incredible response time). It tells a lot about how much you enjoy it too.” — Claude Cavelius*

*"During the last few weeks, I had the opportunity to collaborate in a truly challenging, interesting project involving Computer Vision (CV) and Natural Language Processing (NLP). The CV part is now deployed and fully functional. I just wanted to tell you: **THANK YOU. Without your course, I would have never been able to complete the project.** Your course is the best content vs. value combination I have come across in ages." — Javier Rodriguez Zaurin*

"Your course is awesome. I've been working through the lessons and trying the sample code with additional images. Your explanations are very concise and well-thought out. **I'm most impressed by the way the lessons have seamlessly flowed together.** Adrian, I think you've really got a skill for keeping students focused on what's essential." — John Stocking

The PyImageSearch Gurus course has clearly helped these developers, researchers, and students master computer vision — *so what about you?*

Are you ready to take the *next step* in your journey to become a computer vision master and join the PyImageSearch Gurus course?

I believe you are.

I know you can do it.

And I'll be there to help you out every step of the way.

[Click here to join PyImageSearch Gurus.](#)

I'll see you inside the course.