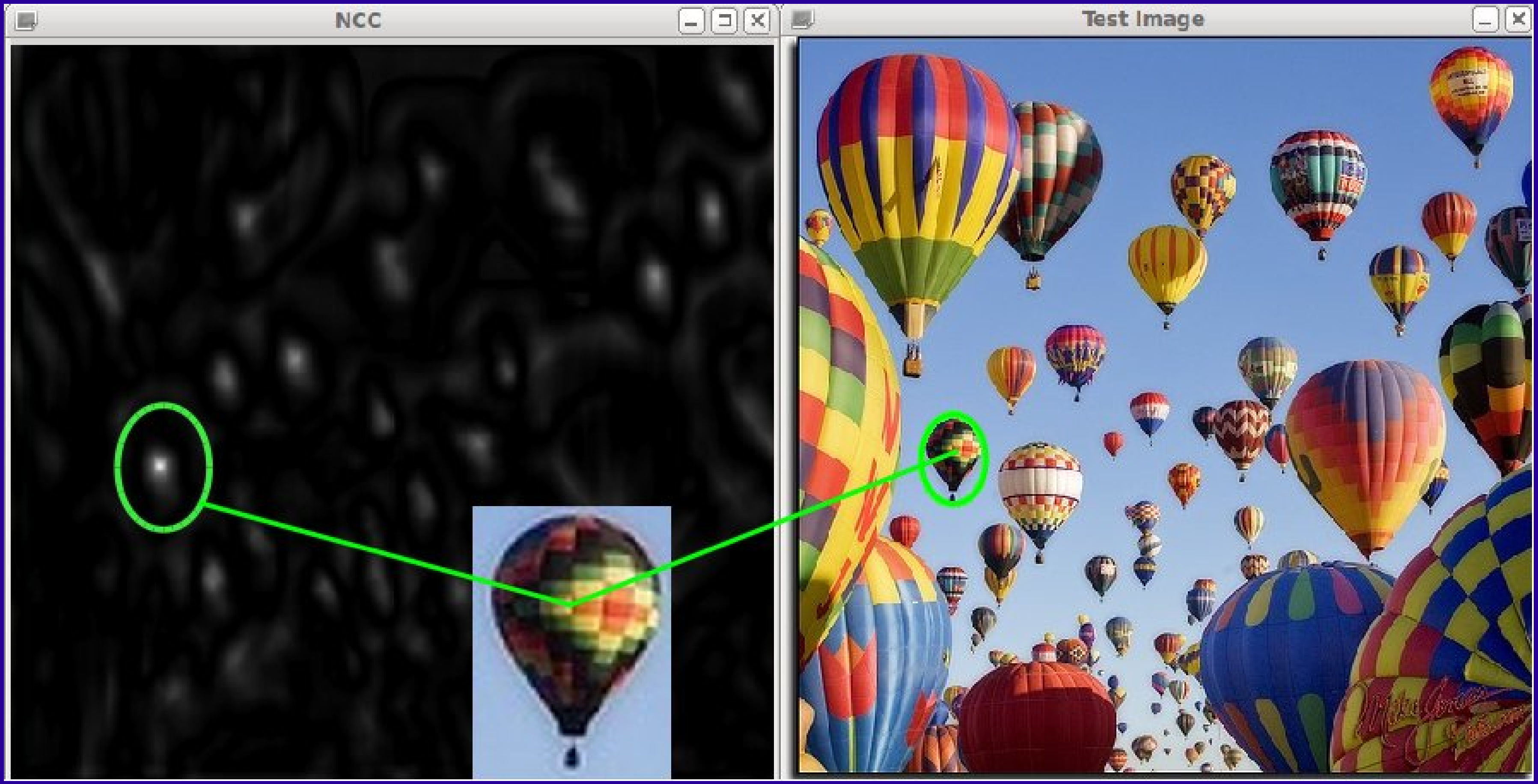
Computer Vision & Machine Learning for Robots



Where's Waldo?

OpenCV Template Match

30 milliseconds

Visual Pattern Matching

- Household objects; e.g. mug on a table, beer in the fridge
- People, faces, pets
- Landmarks for navigation; e.g. AR markers (fiducials)
- Keypoints for Visual-SLAM

Strategies for Object Detection & Recognition

- Template Matching
- Feature Matching
- Machine Learning (learning from examples)

Detection *vs* Recognition

- Detection: is there a face, any face, in this picture? E.g., OpenCV Haar face detector

- Recognition: ***who*** is that face in the picture? E.g. Eigenfaces

Vision Software

- OpenCV – Linux, Windows, MacOS X, Android
  - C, C++, Python, no GUI)-:
  - Advanced vision algorithms including SIFT, SURF, face detection, machine learning
- RoboRealm – Windows Only
  - C, C++, Python, VBScript, very nice GUI
  - Many, many vision filters
  - Control for many popular robots and cameras
  - None of the patent-protected algorithms

Software Continued...

- PCL (Point Cloud Library)
  - Linux, Windows, MacOS X, Android
  - C++, no Python, no GUI)-:
  - 3D planar segmentation, clustering
  - www.pointclouds.org
- Ecto – Linux only?
  - Combine OpenCV & PCL using Python image processing pipelines
  - Easily extendible
  - ecto.willowgarage.com

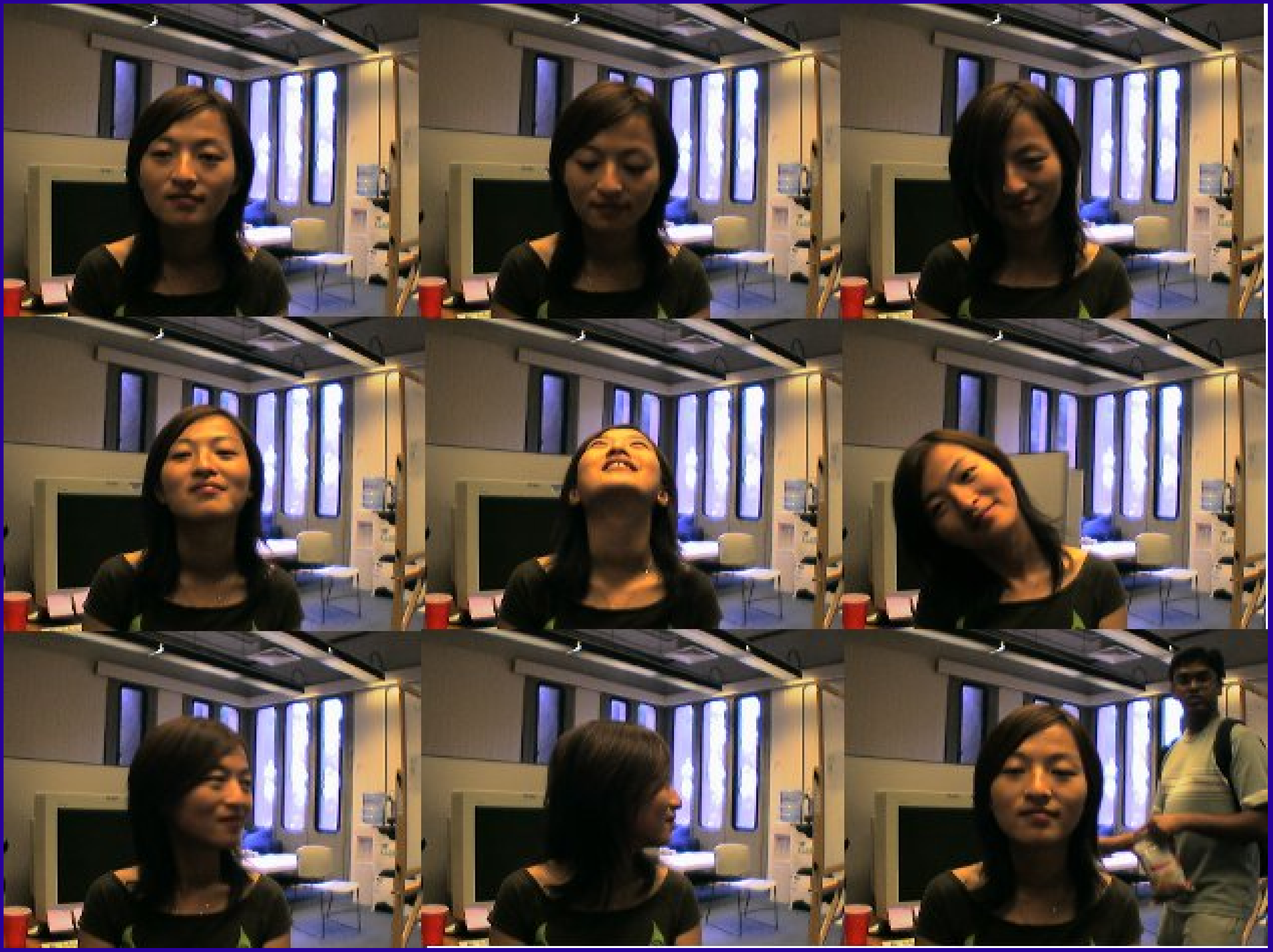
Appearances are Everything

Special Challenges for Robotics

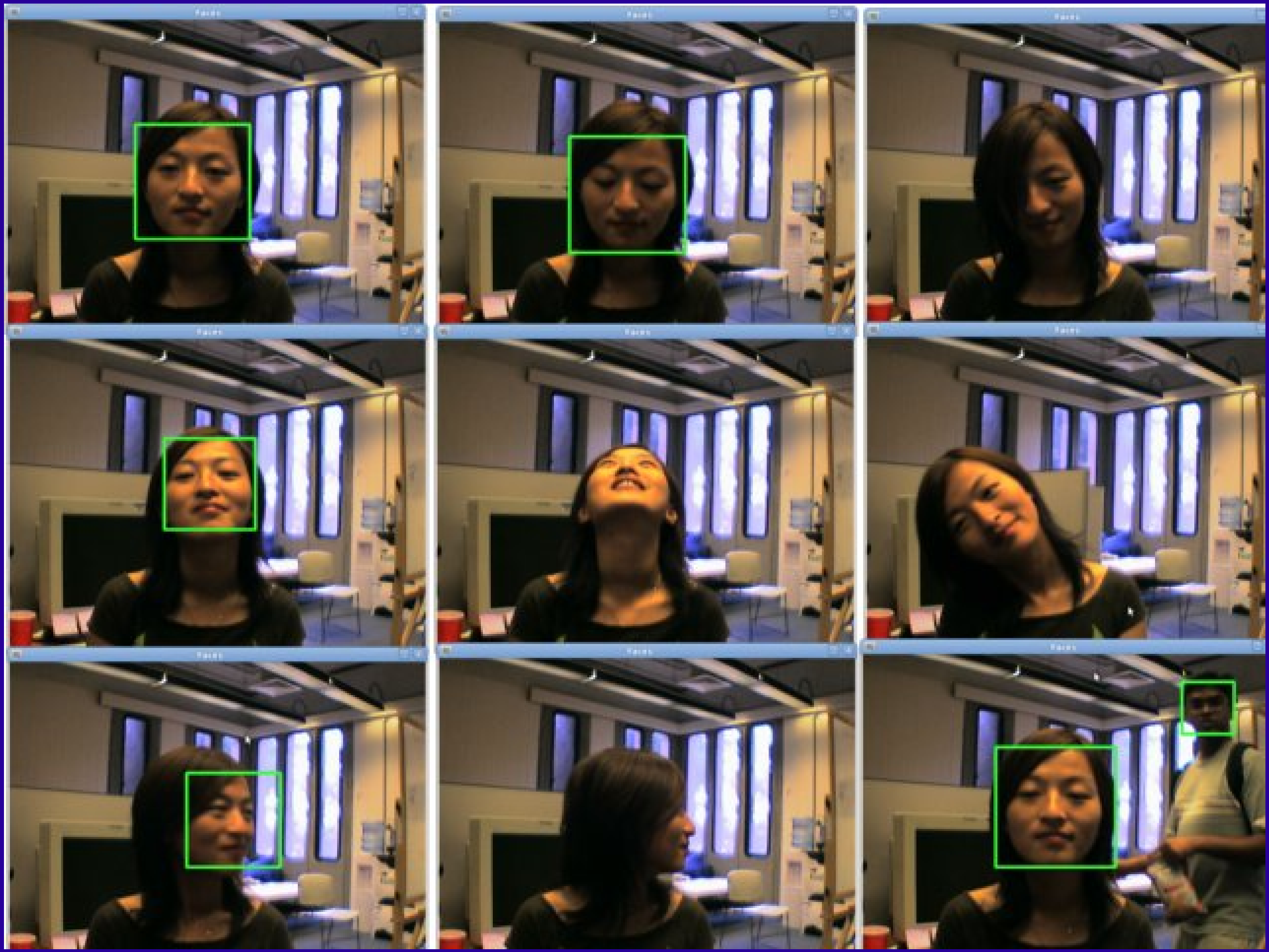
- Live video rather than static pictures
- Changes in orientation and position (occlusions, rotations, scale, affine transformations)
- Changes in lighting and background
- Motion of object and/or camera
- Changing appearance (today Joe has on a hat and glasses)
- The world is 3D while pictures are 2D (Kinect/Xtion)
- Pose estimation (e.g. for grasping)
- Speed: at least 10-30 fps

Facing the Challenge

Courtesy of Honda/UCSD Video Database

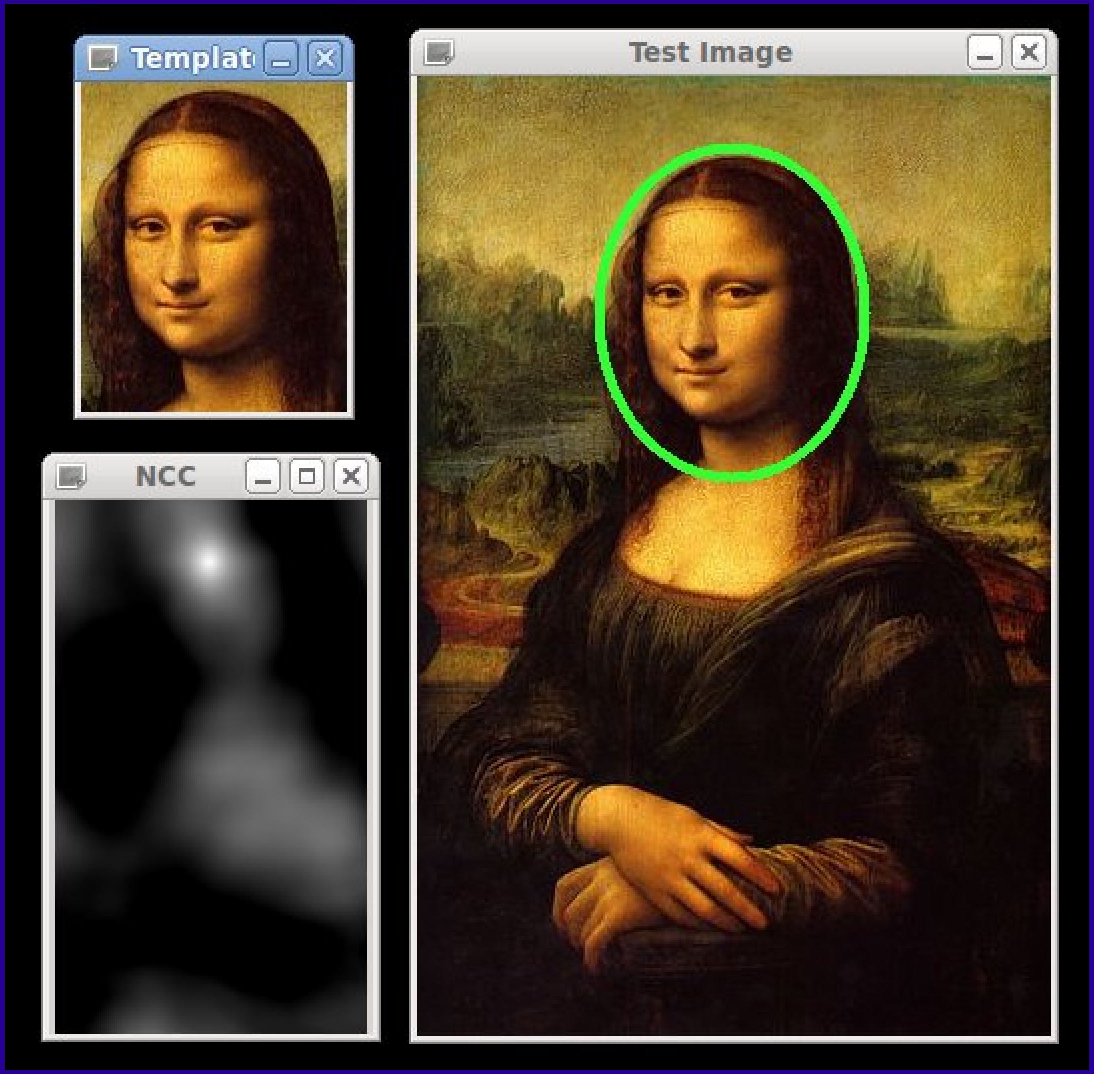


Haar Face Detector on Video



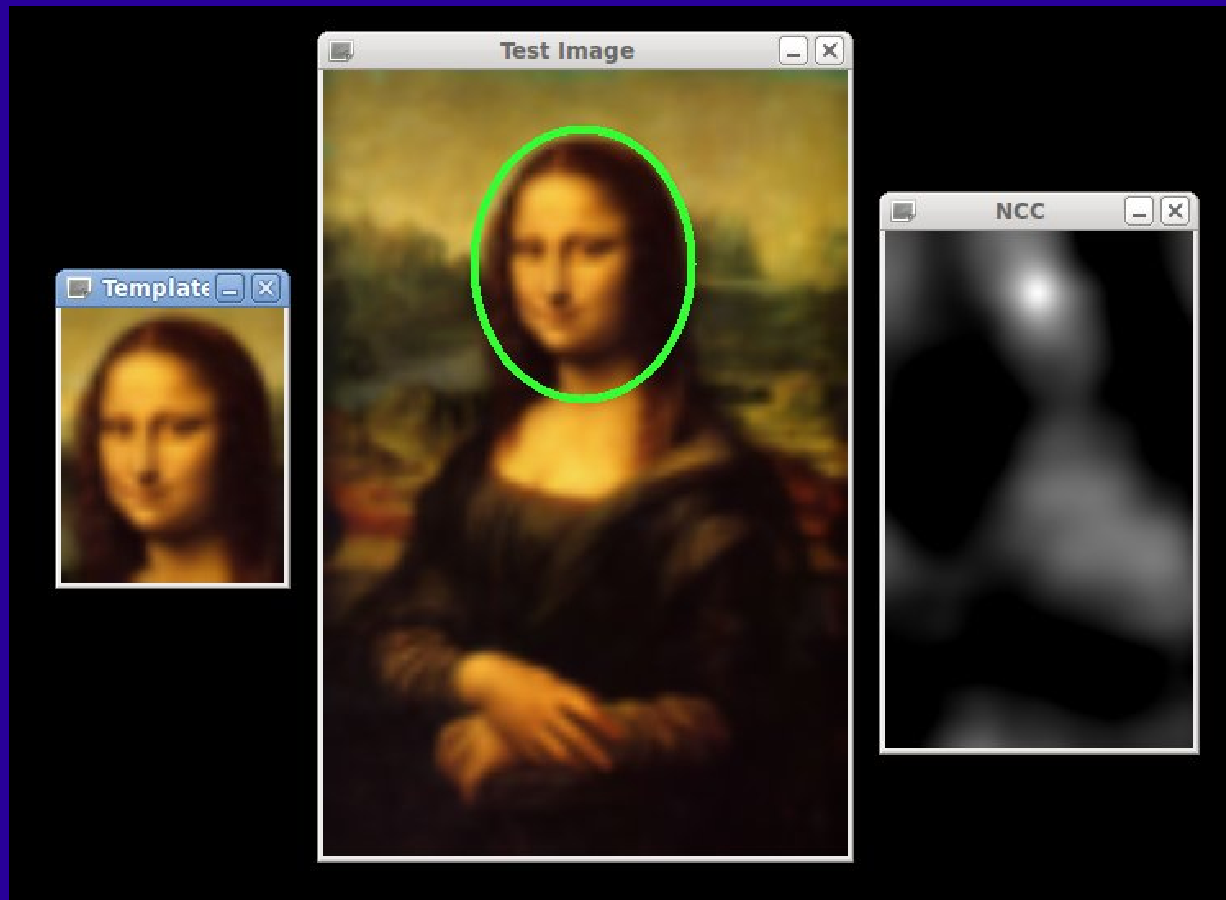
Template Matching

317x480 pixels, *40 milliseconds*



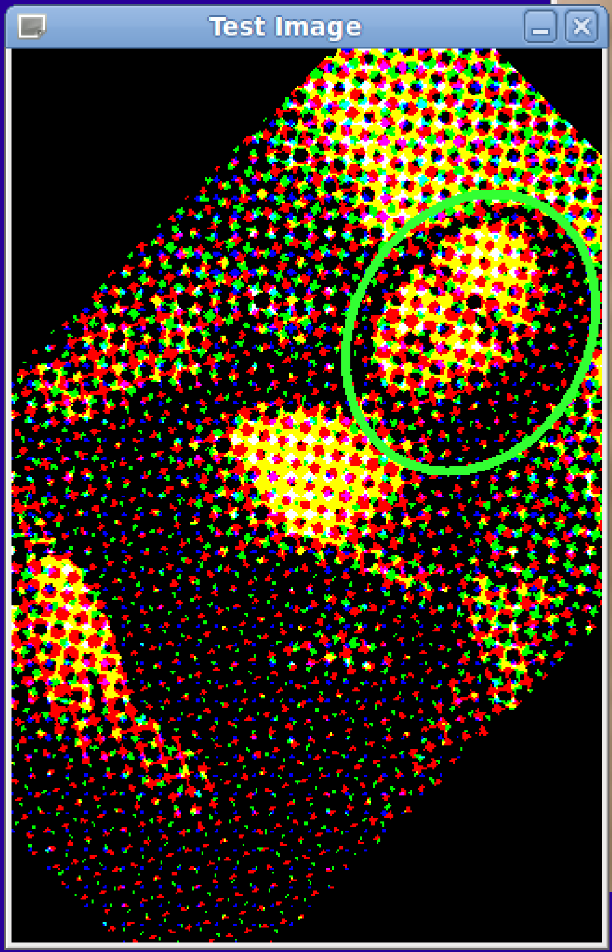
Using pyrDown() Twice

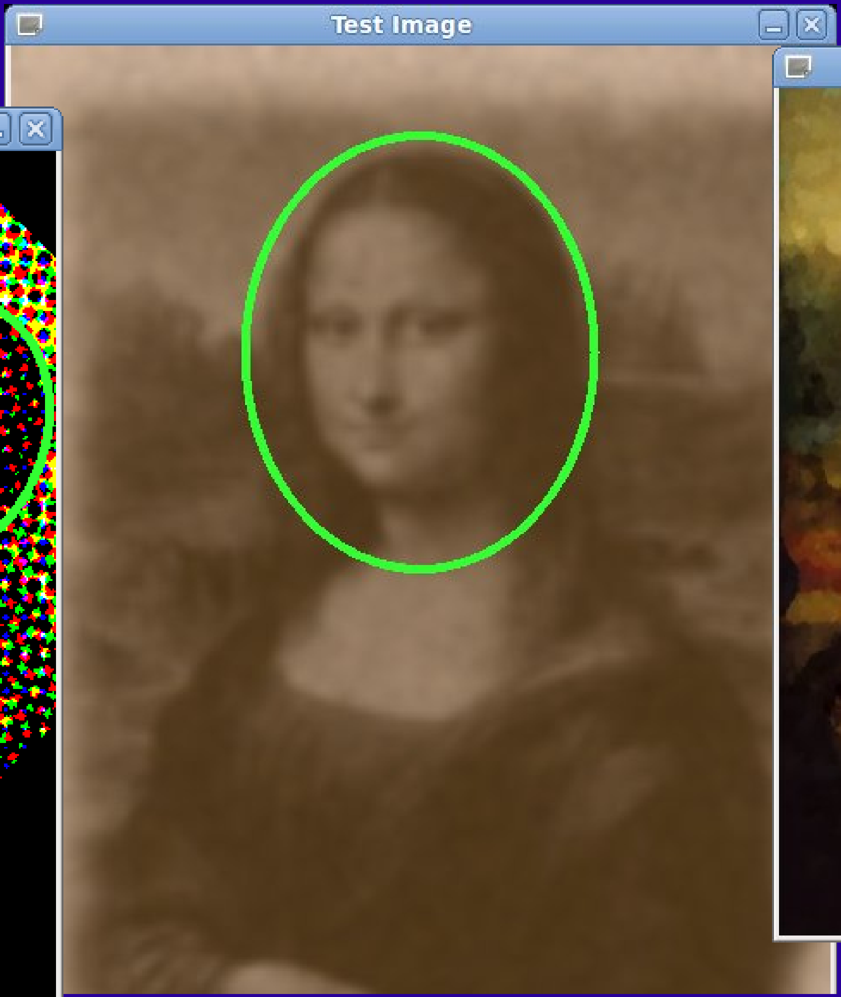
under 10 milliseconds

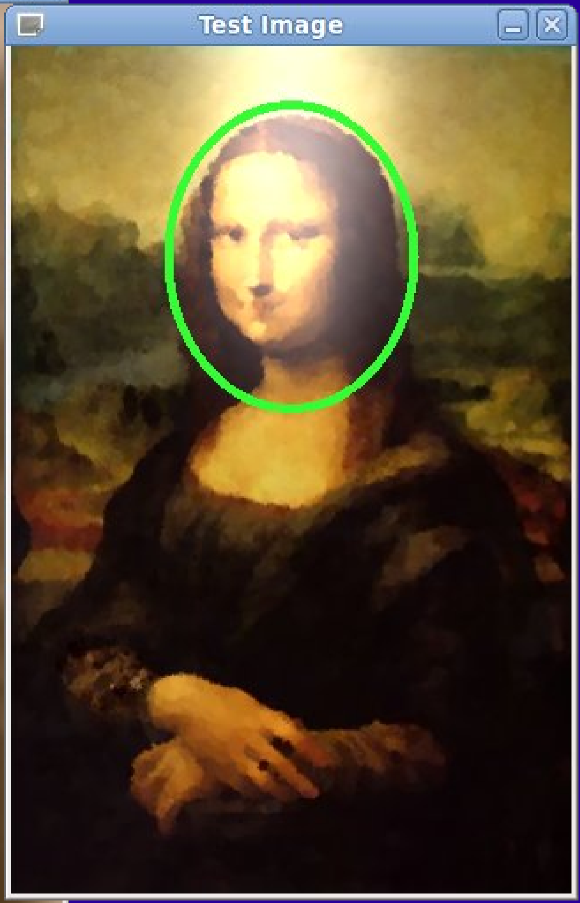


Template Matching Including Scale & Rotation

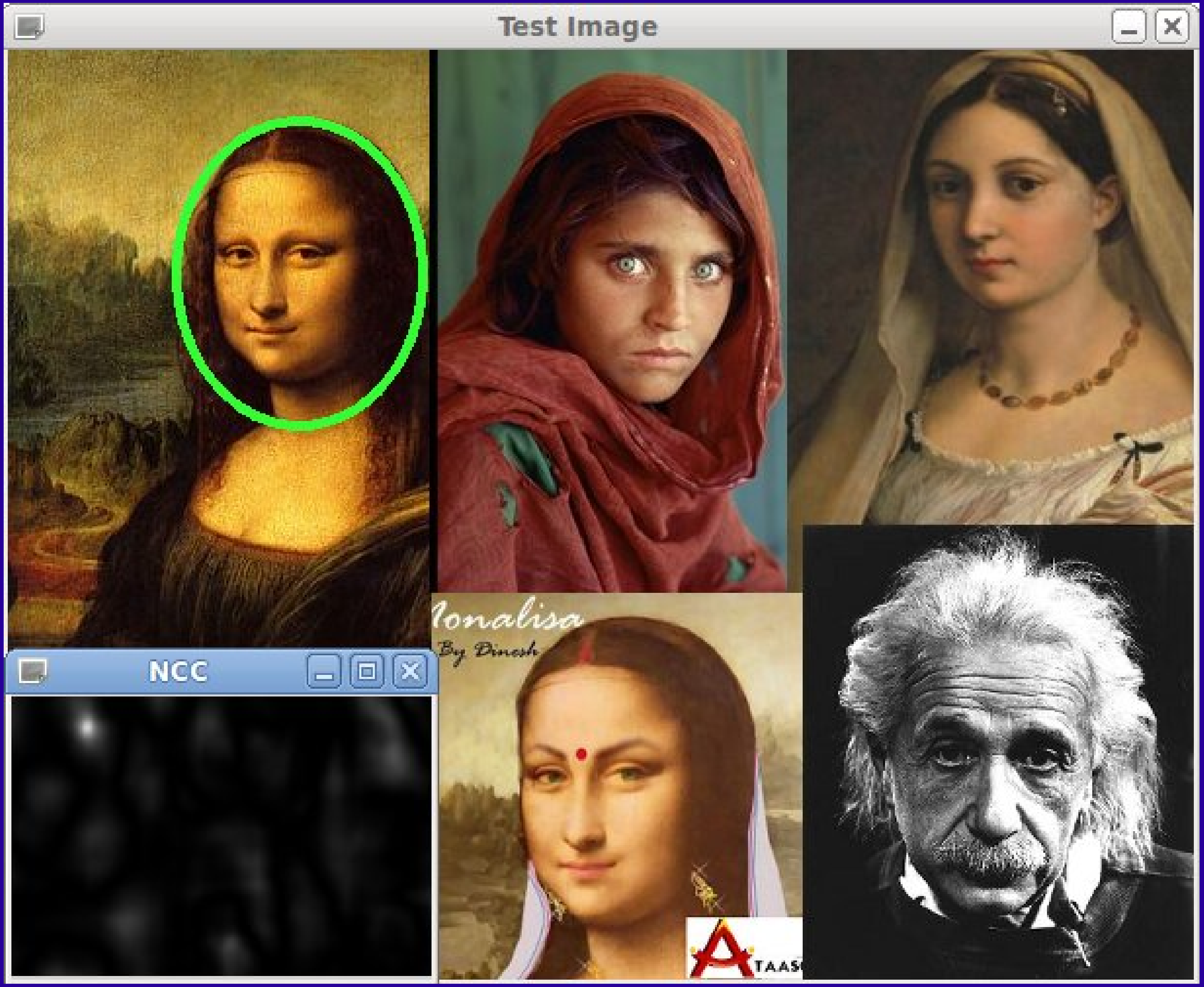
230 millseconds





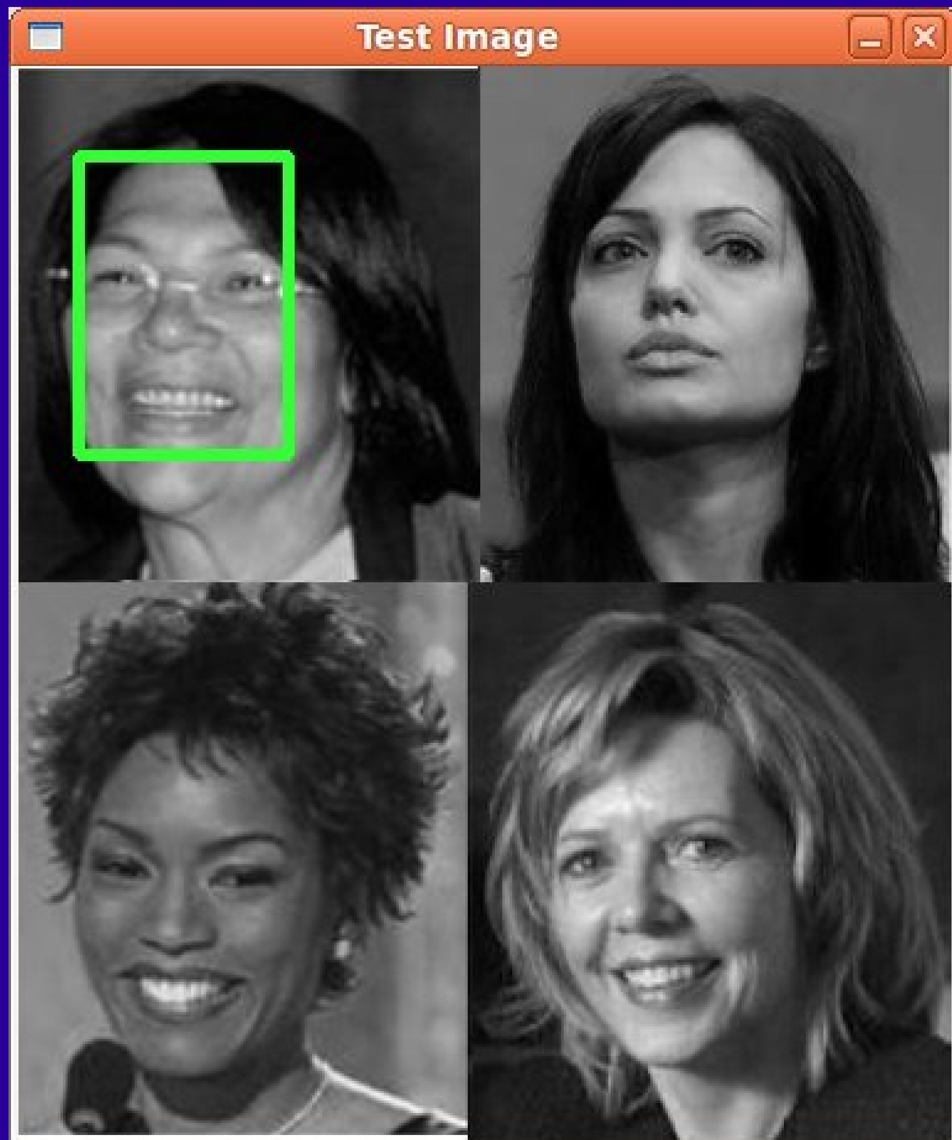


Template is Object Specific



Template Matching on Video

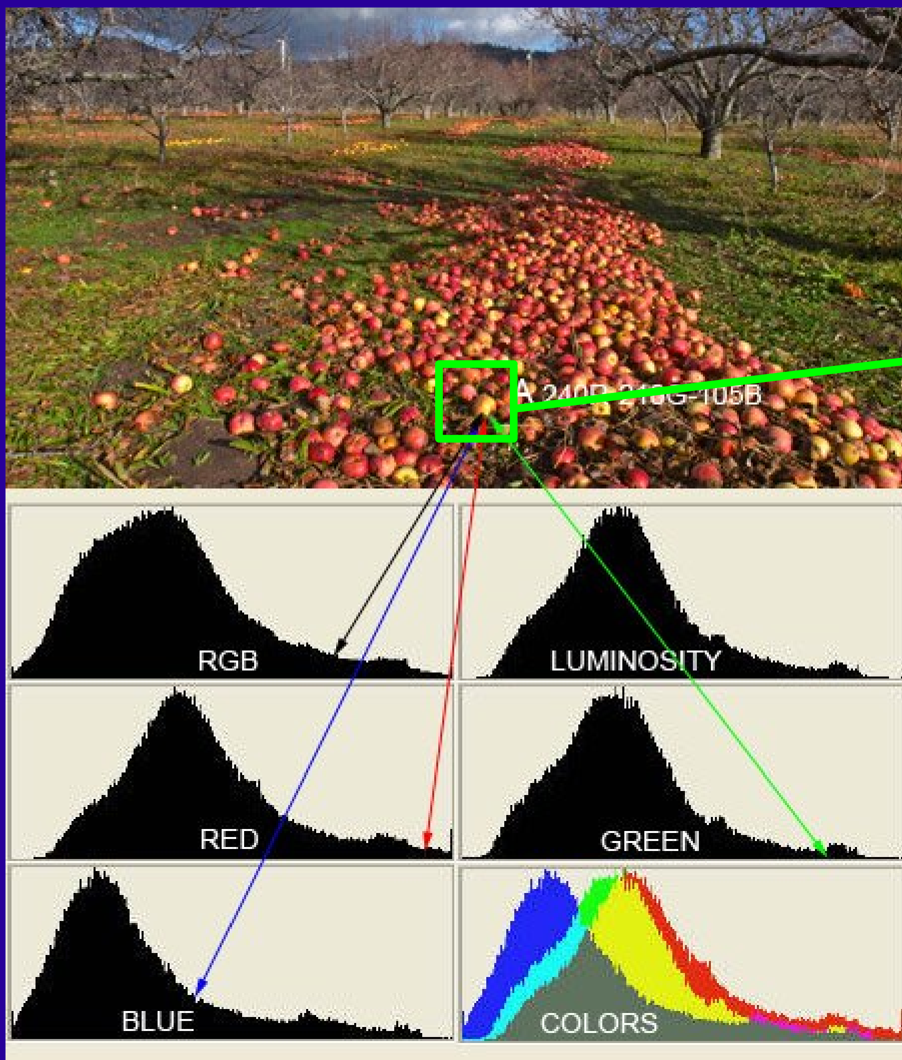
Template Limitations

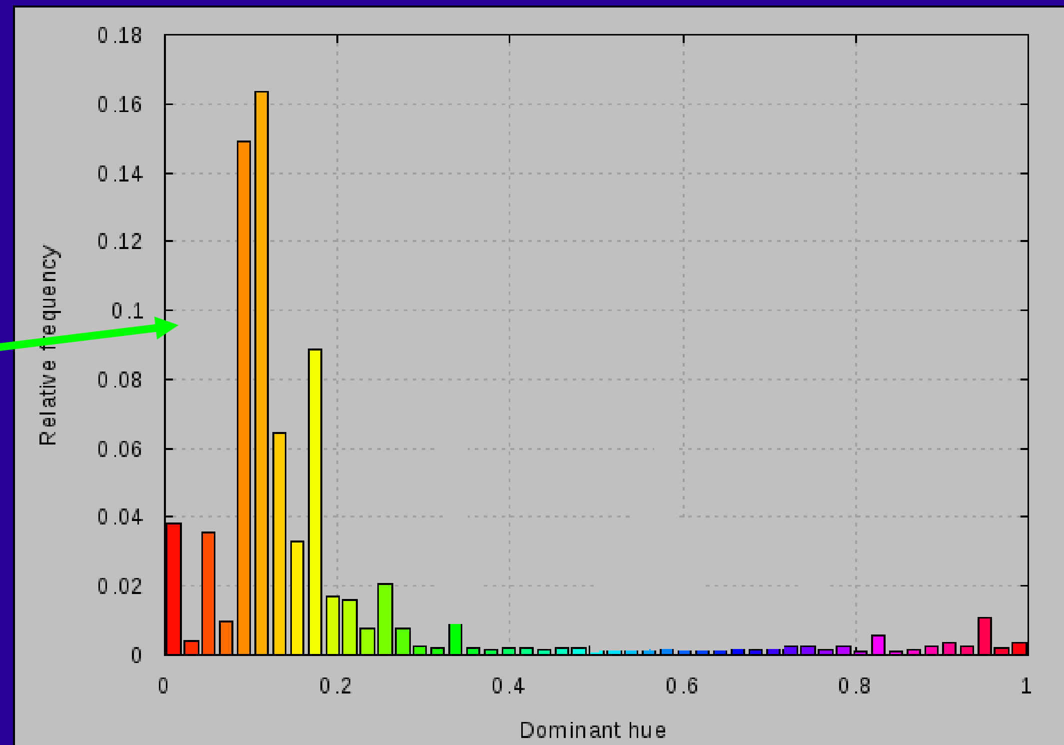


From Pixels to Features

- What is a feature? A summary statistic or descriptor computed from pixel values in an image patch, region of interest (ROI), or even the whole image
- Colors → histograms (e.g. RGB, hue)
- Pixel intensities → gradients, binary comparisons, histograms, interest points (well defined locations)
- Connected regions → contours, segmentation
- Ideally, invariant to scale and rotations

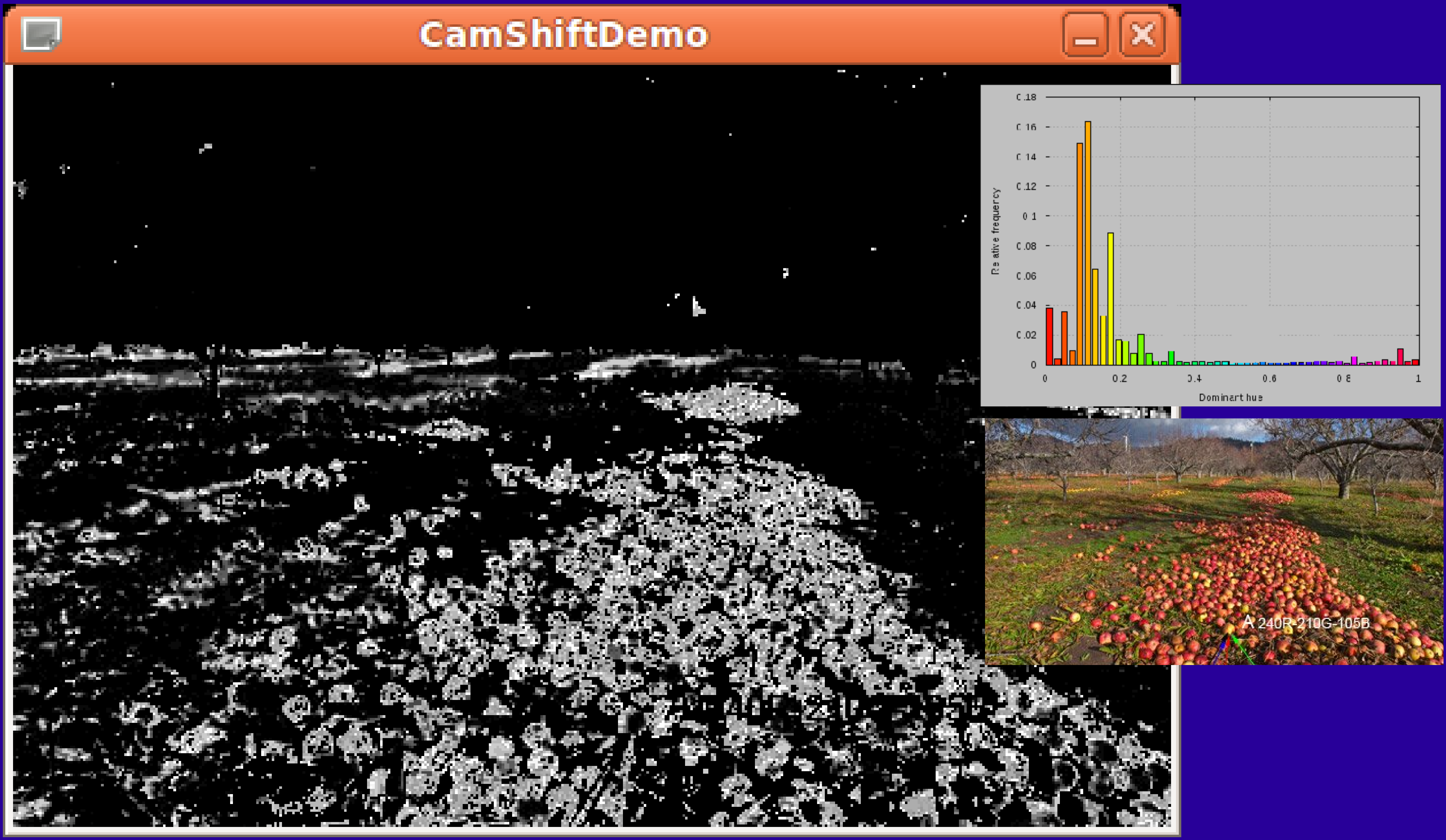
Color Histograms





0.038	0.01	0.036	0.011	0.15	. . .	0.05	0.062	0.041	0.002	0.005

OpenCV CamShift Tracking



Interest Point Detection

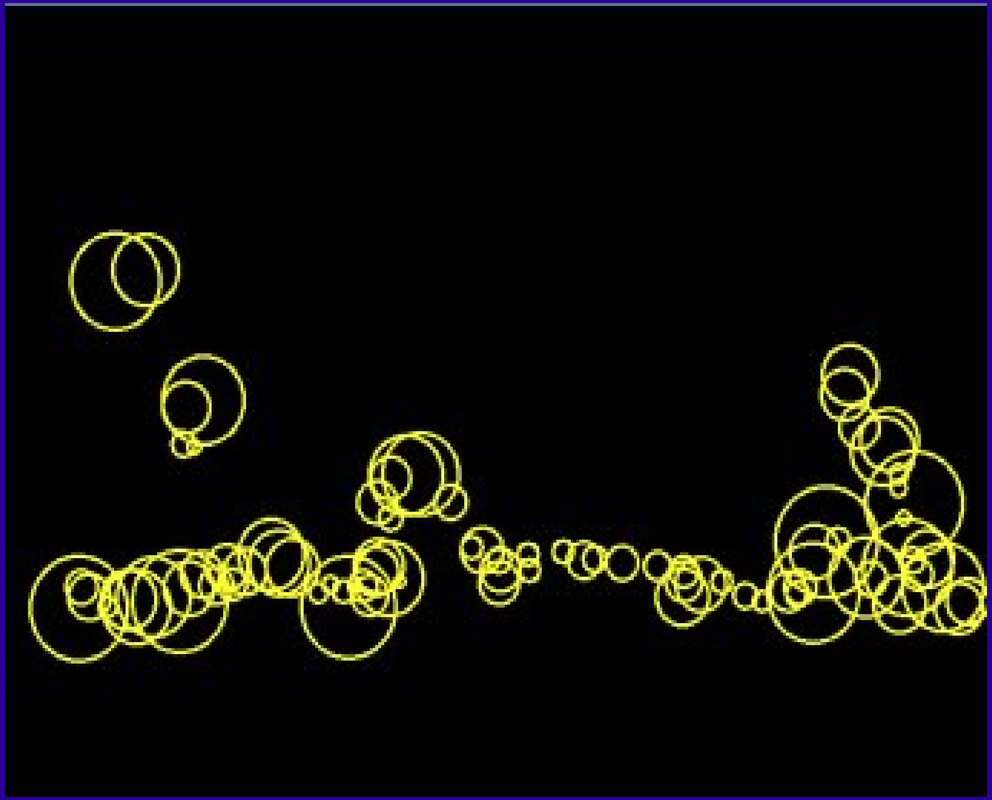
Harris Corners, SUSAN, FAST

$$\mathbf{M} = \begin{bmatrix} \left(\frac{\partial f}{\partial x}\right)^2 & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \left(\frac{\partial f}{\partial y}\right)^2 \end{bmatrix}$$

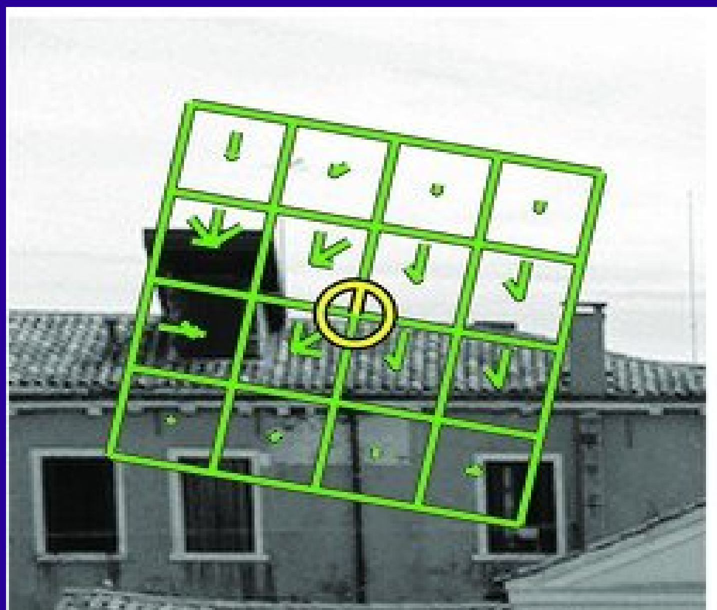
$$R = det\,\mathbf{M} - k\,Trace^2\,\mathbf{M}$$

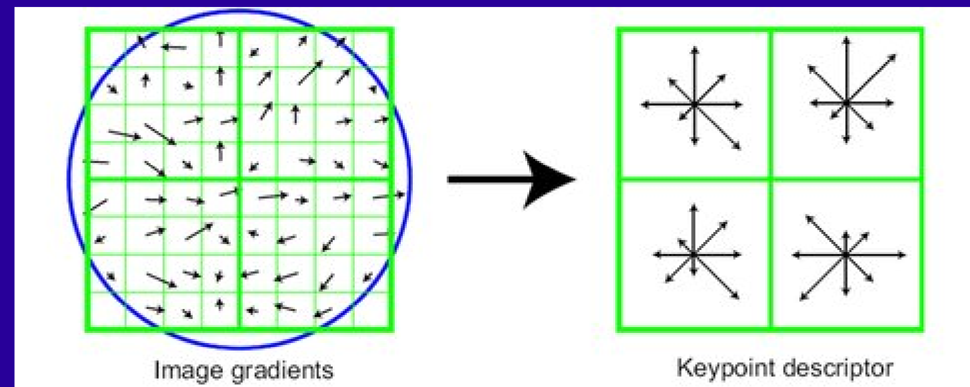
Interest Points = Keypoints

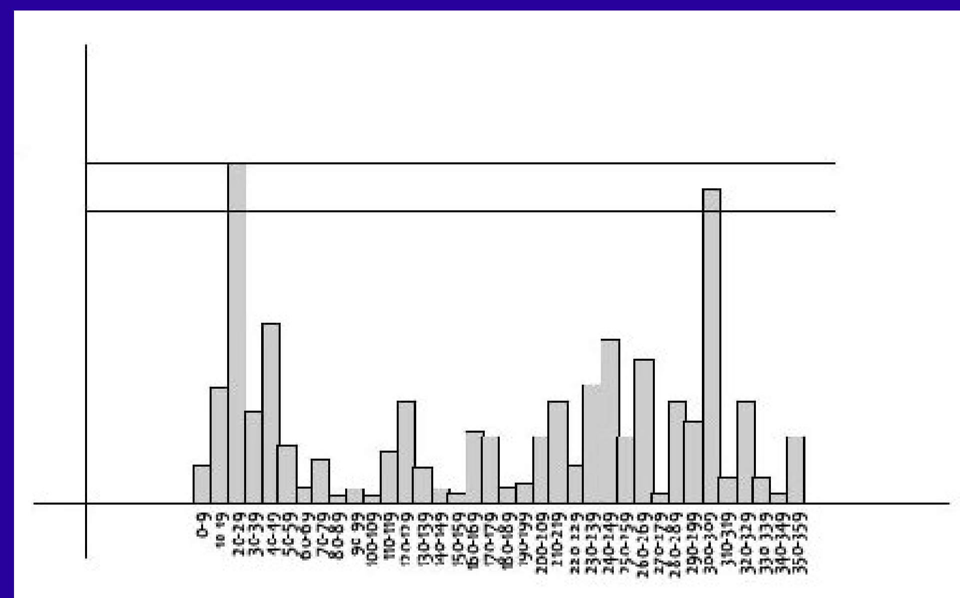
Another Keypoint Example



Keypoint Descriptors



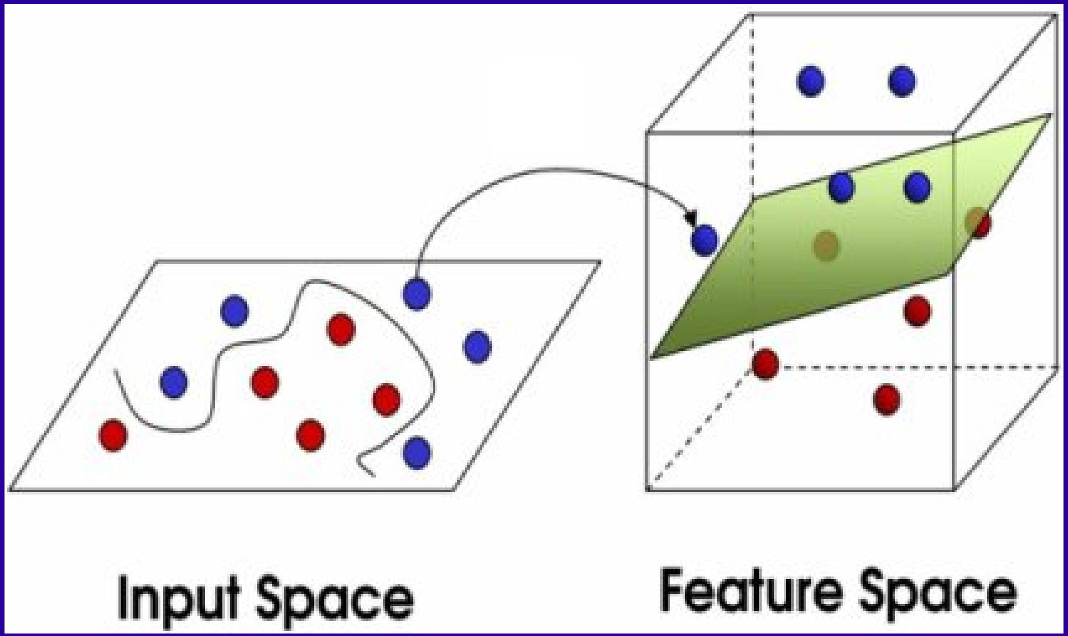


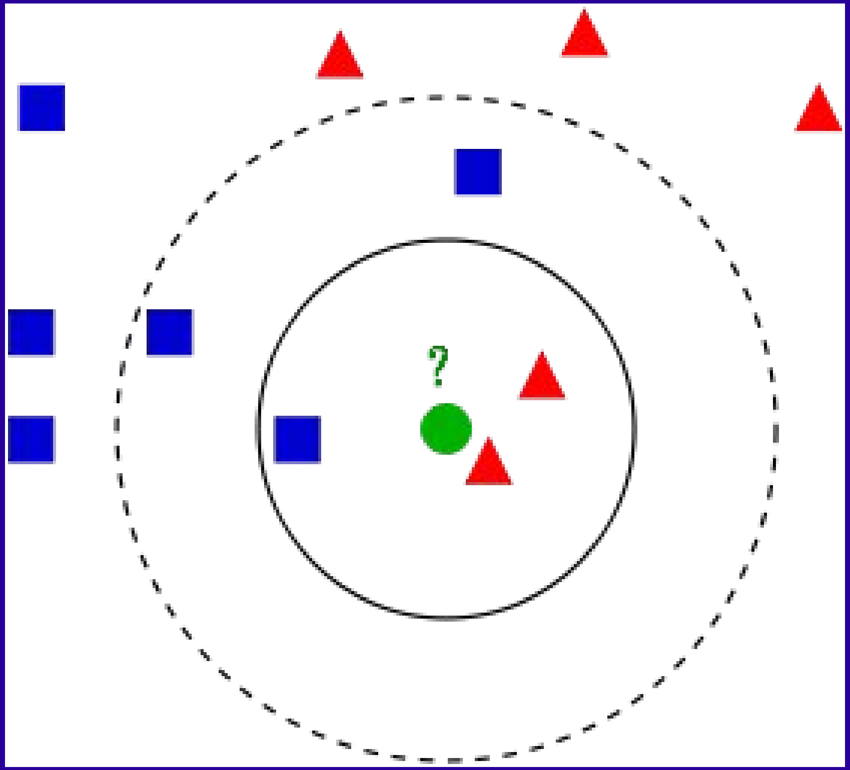


0.038	0.01	0.036	0.011	0.15	. . .	0.05	0.062	0.041	0.002	0.005

Descriptors Live in Feature Space

Distance Measures: Euclidean, Manhattan, Hamming



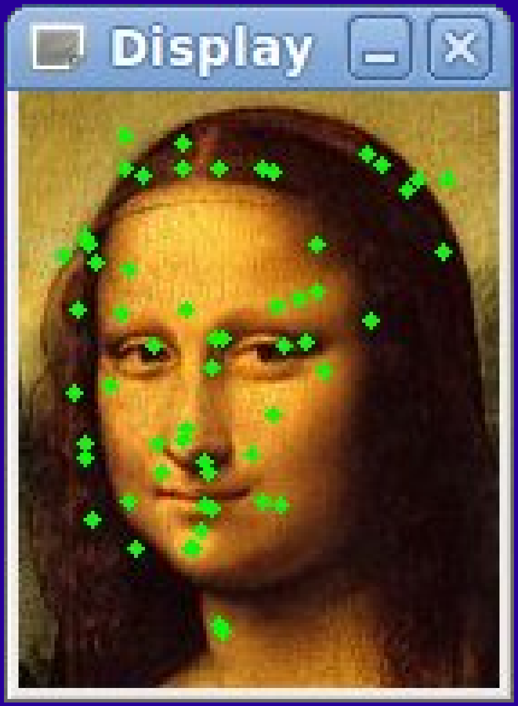


Descriptors Available in OpenCV

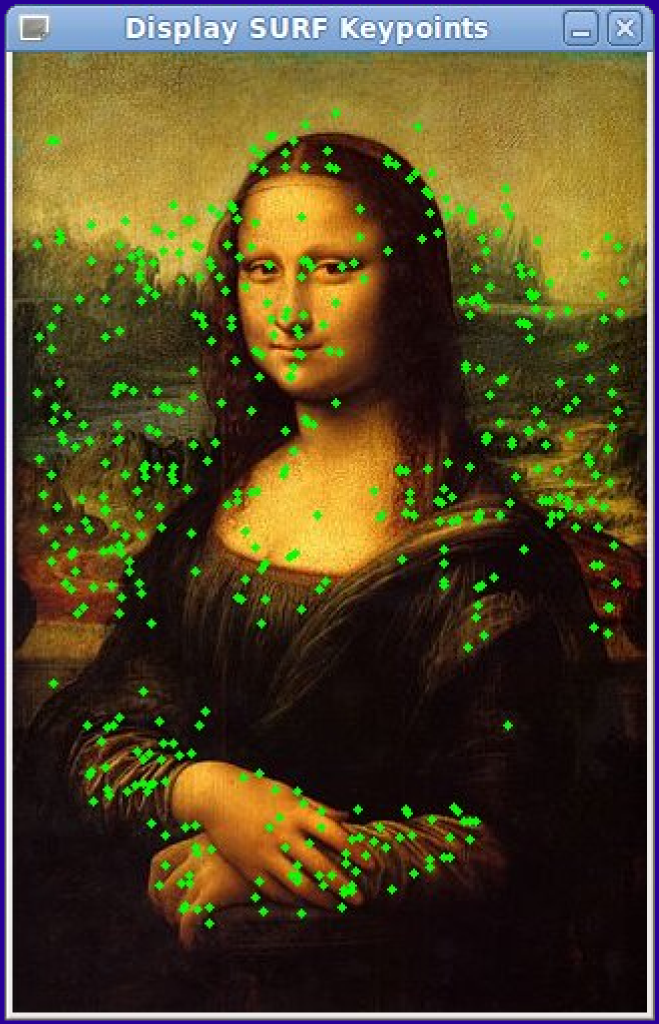
- SIFT – Scale Invariant Feature Transform
- SURF – Speed Up Robust Features
- BRIEF – Binary Robust Independent Features
- ORB – Oriented FAST + Rotated BRIEF

OpenCV SURF Keypoints

58 keypoints
10 ms

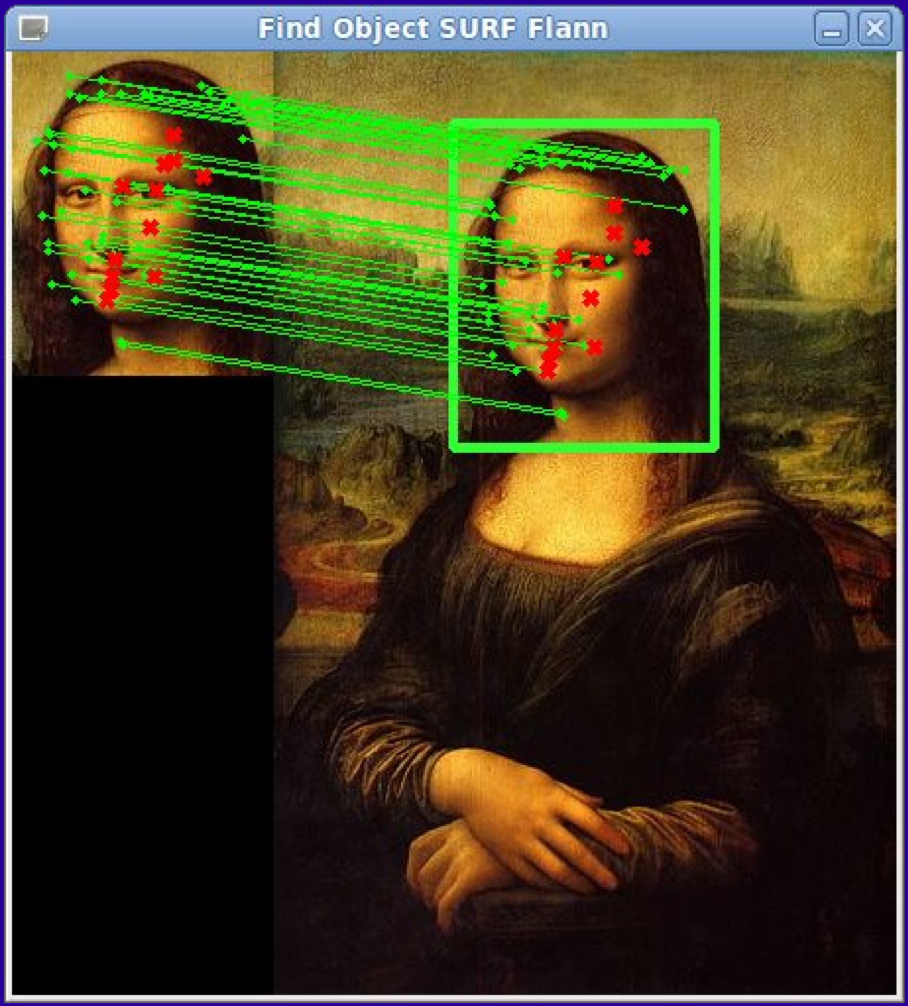


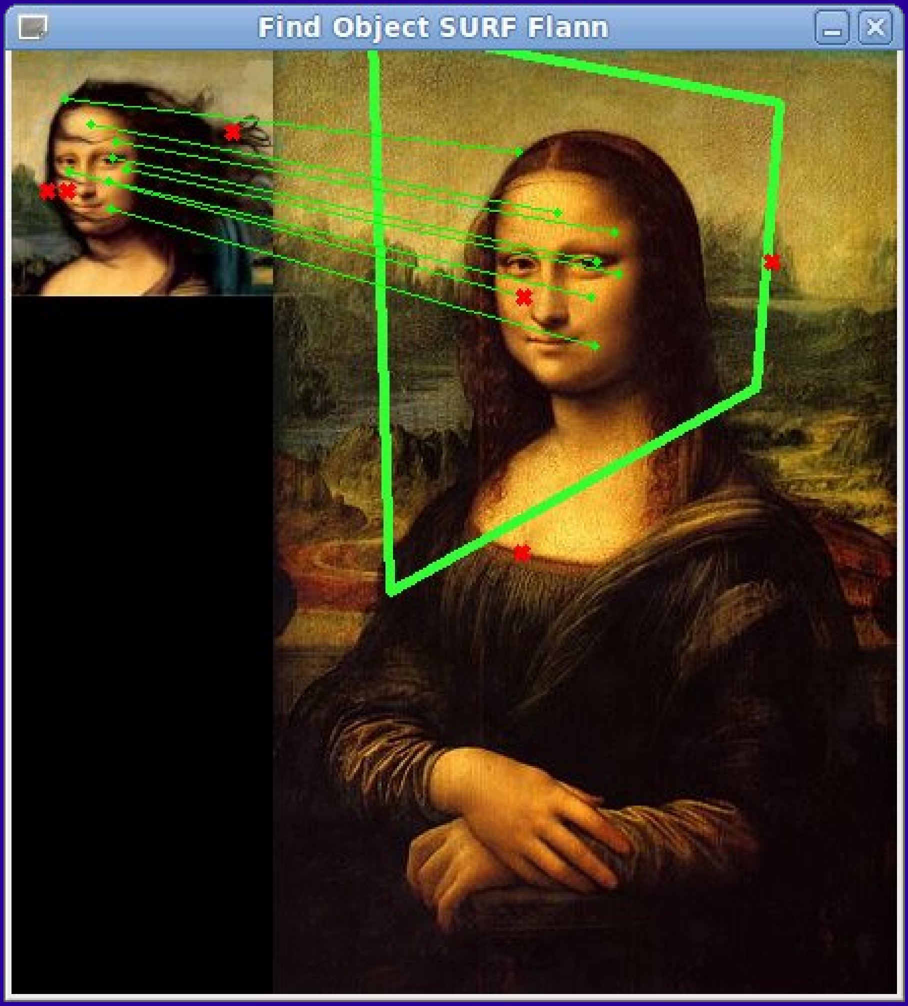
474 keypoints
89 ms



SURF Keypoint/Descriptor Matching

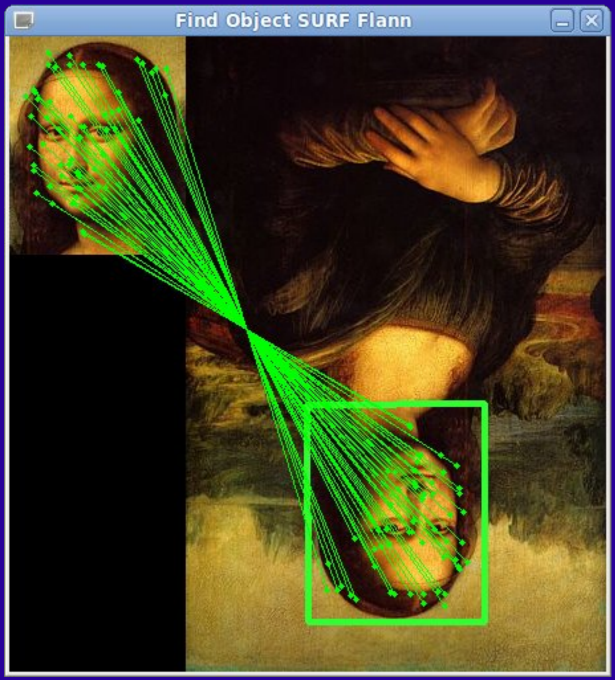
FLANN + RANSAC, *85 ms*

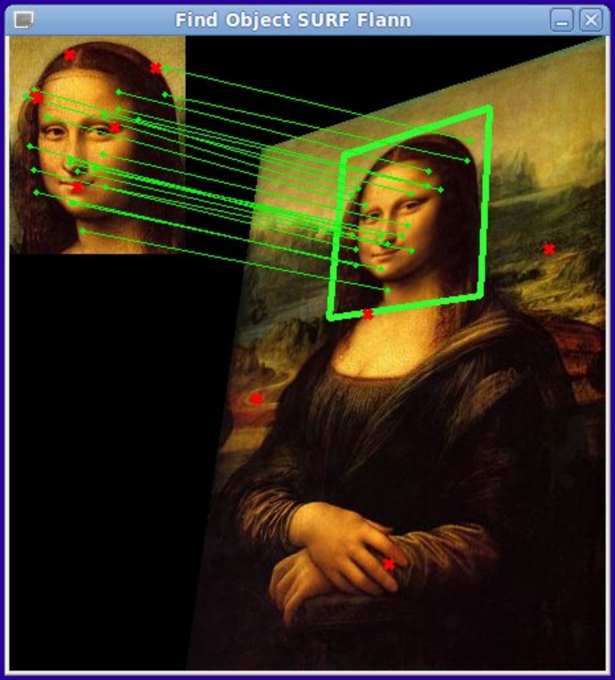


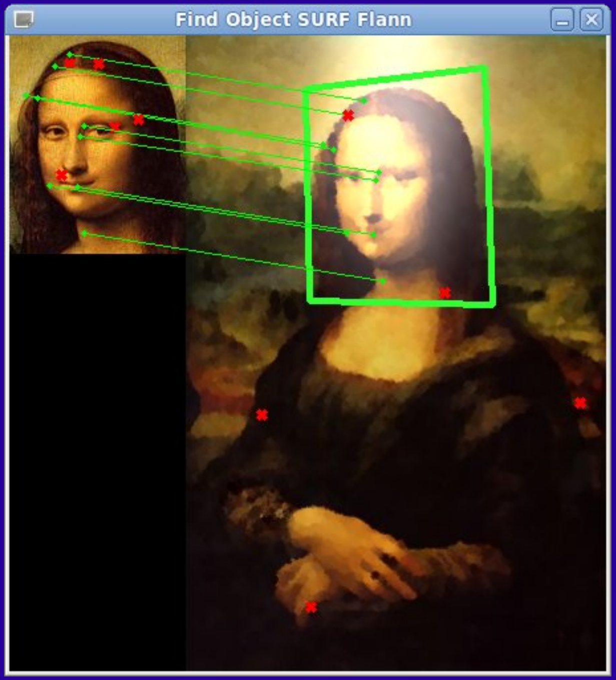


SURF Matching Examples

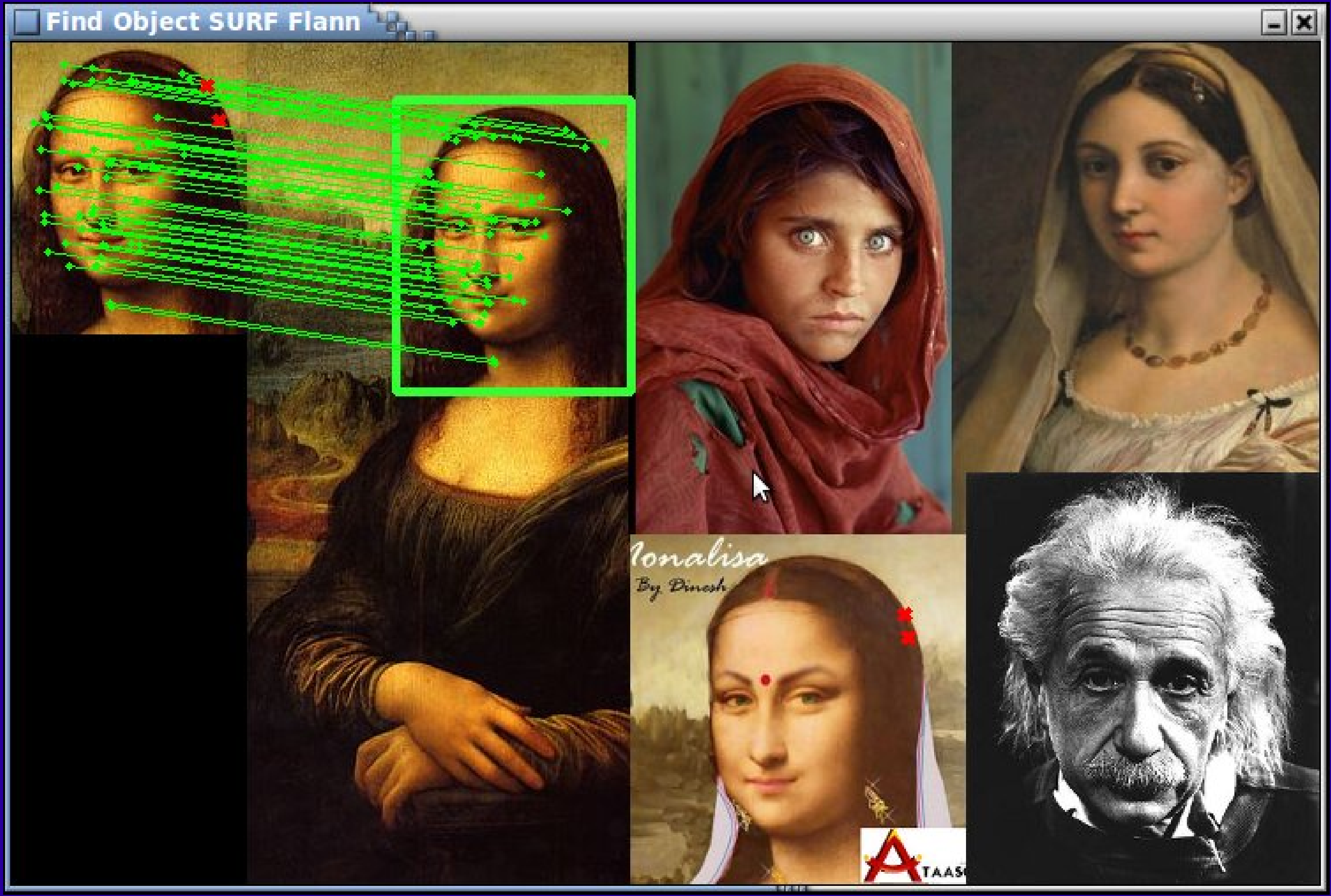
Upright = False, 180 ms



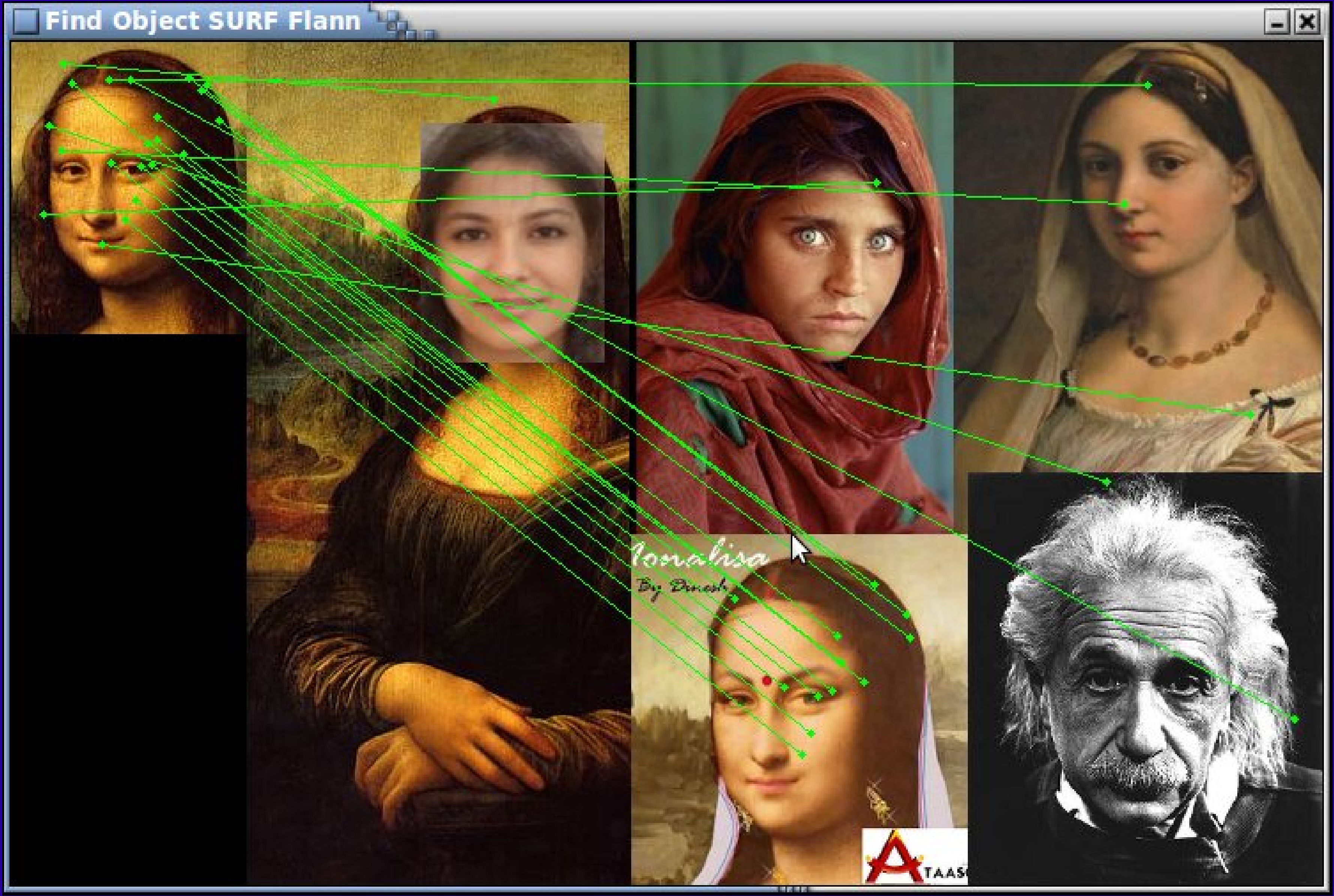




SURF is (Partially) Object Specific

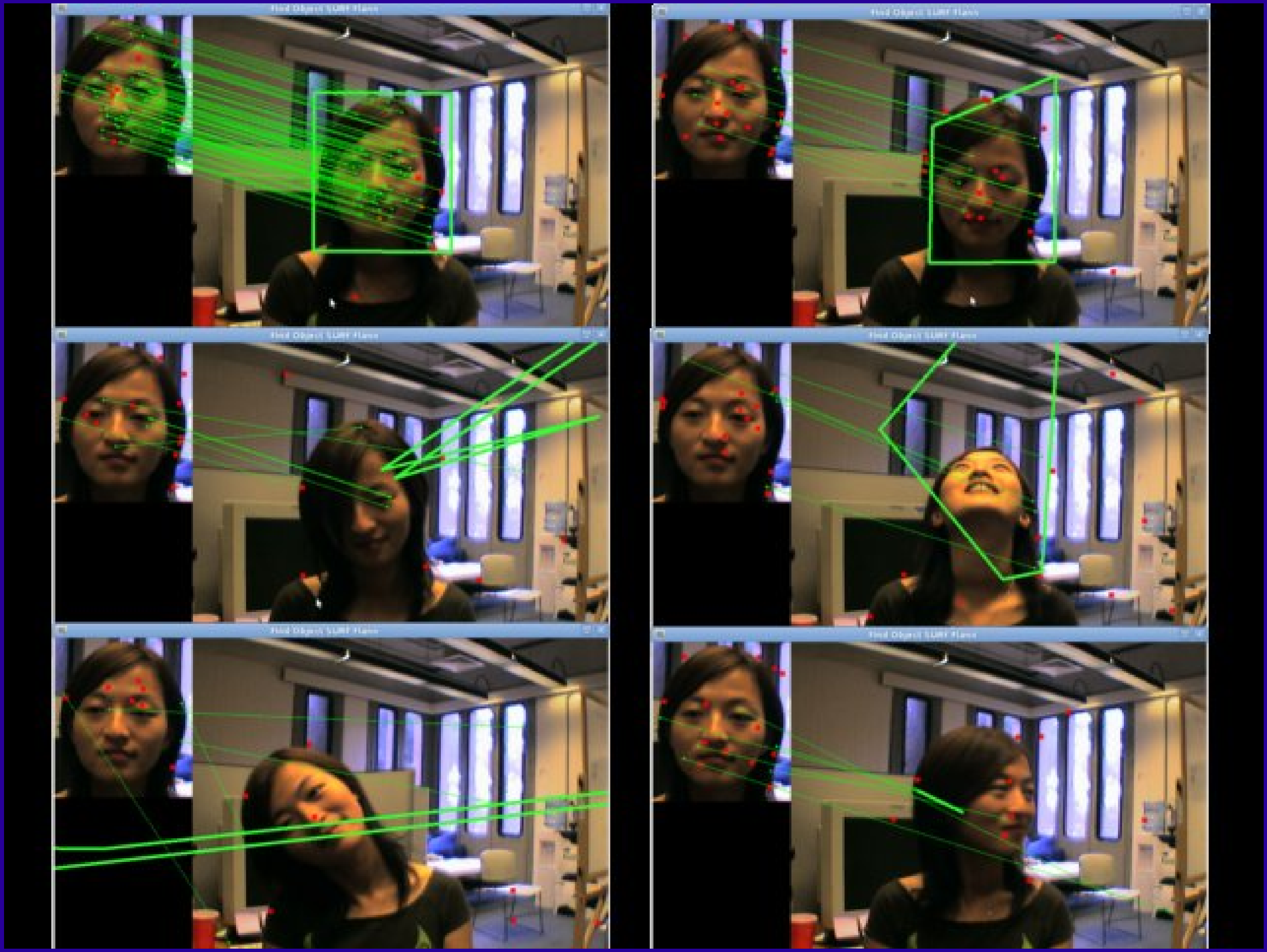


Partial SURF Match

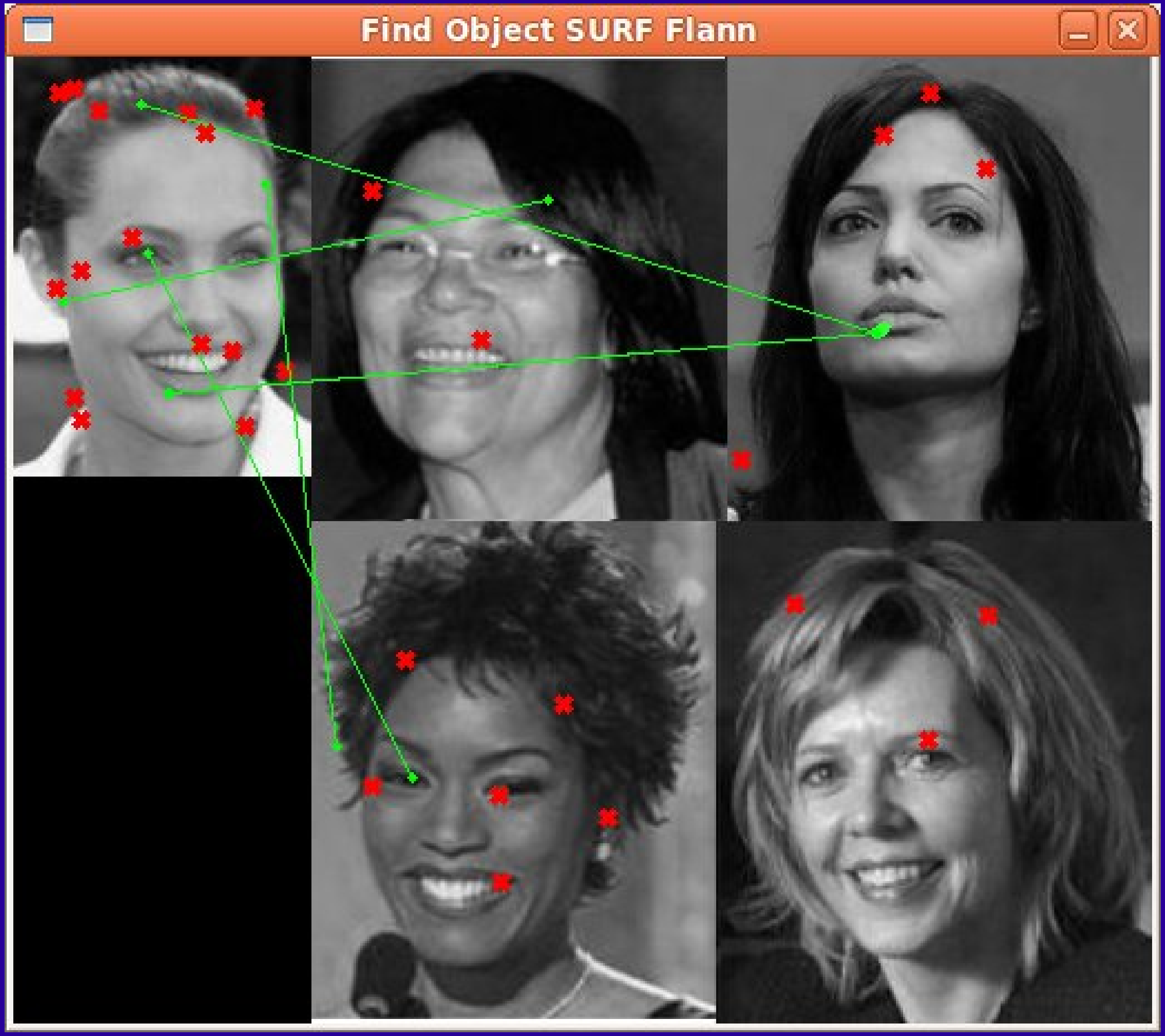


Finding Waldo with SURF

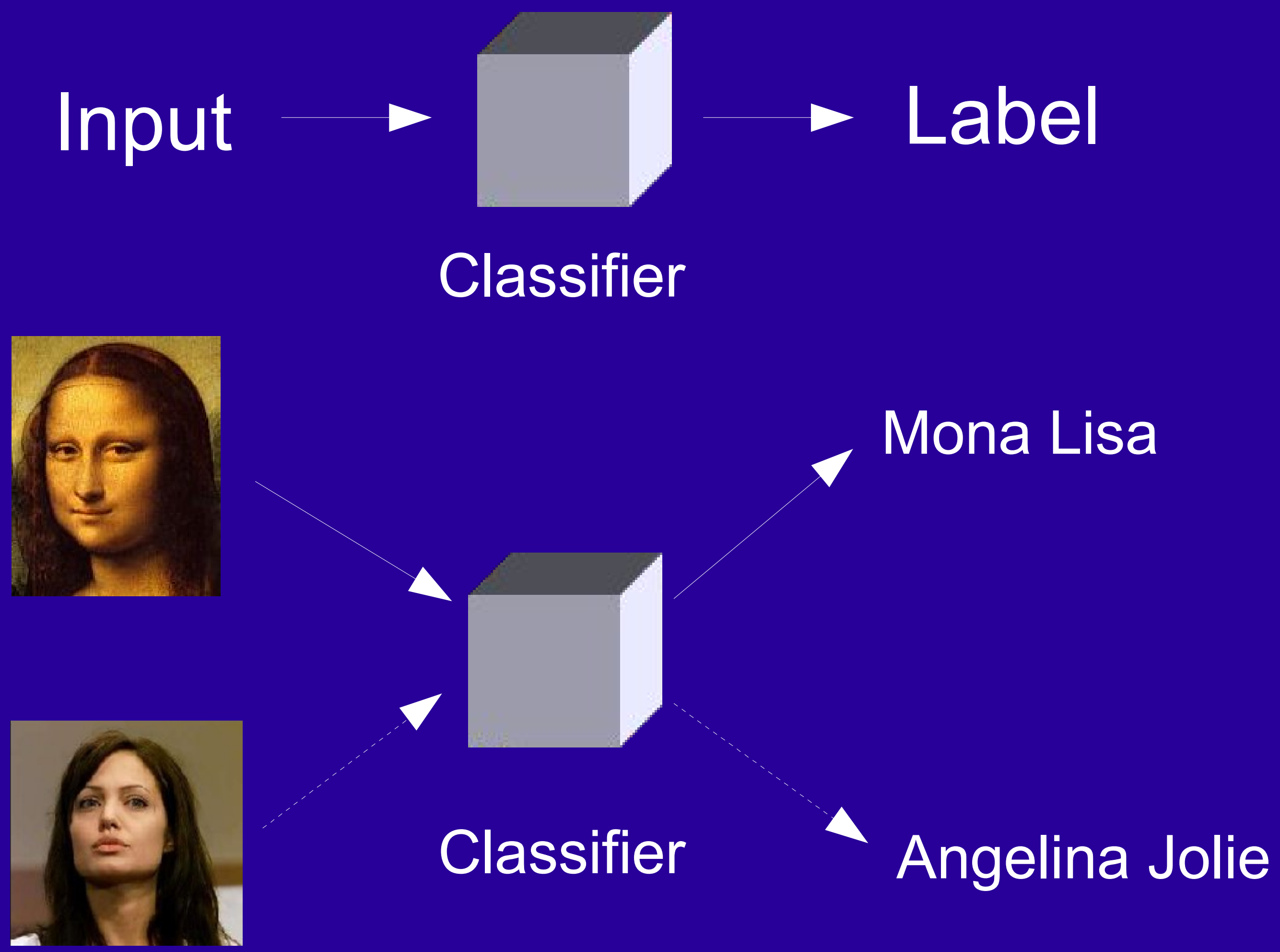
SURF Matching on Video



Keypoint Matching Limitations



Machine Learning & Classification



Building a Classifier

- Acquire training samples; e.g. Google Images, research databases (AT&T, Yale), recorded or live video
  - Single class: positive and negative samples
  - Multi-class: samples for each class
- Choose a model (e.g. SVM vs ANN)
- Train the classifier on a subset of the samples
- Test the classifier (cross-validate) on the remaining samples
- Apply classifier to new samples

Machine Learning Software

- OpenCV (http://opencv.willowgarage.com)
  - Linux, Windows, MacOS, Android; C, C++, Python
  - kNN, SVM, Random Trees, Neural Networks, PCA, Normal Bayes, Boosting

- Orange (http://orange.biolab.si)
  - Linux, Windows, MacOS X; Python
  - Visual Programming, Cross-Validation, Learner Comparison, Feature Selection, Ensembles
  - (No Neural Networks)

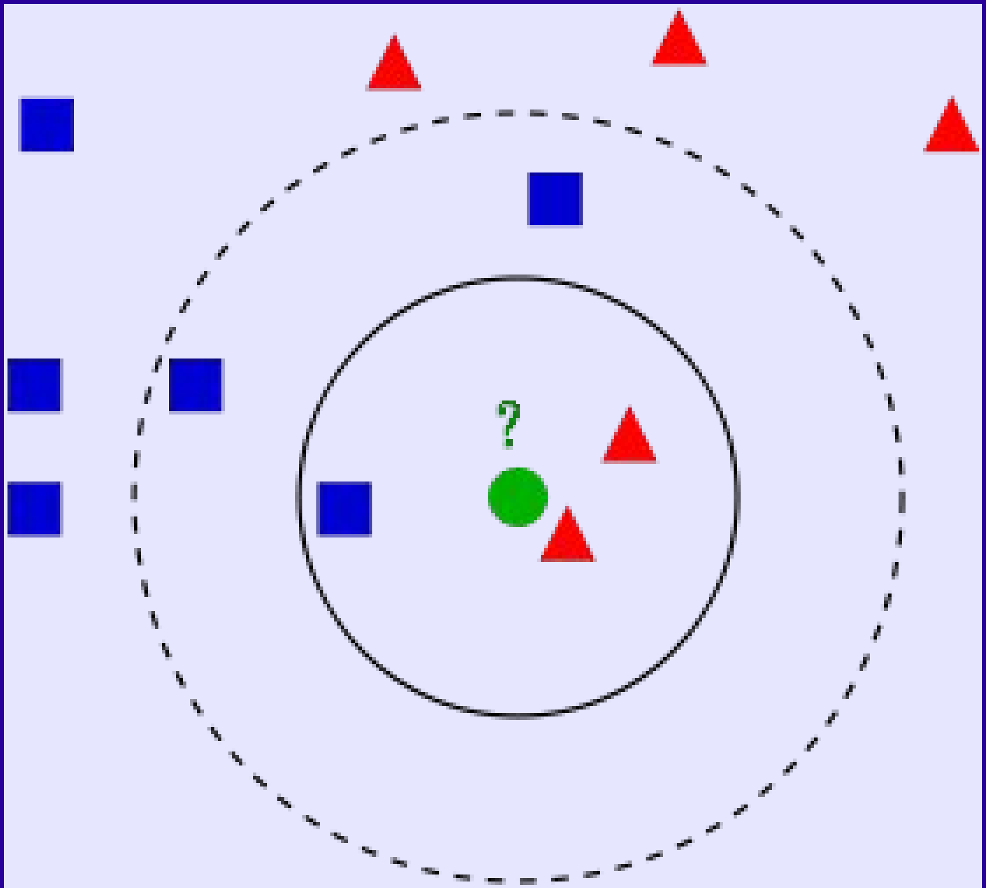
Learning Software Cont'd...

- Scikit-Learn (http://www.scikit-learn.org)
  - Integrated with numpy, scipy, mathplotlib
  - Linux, Windows, MacOS X; Python
  - Stochastic Gradient Descent, Gaussian Mixture Models, (No Neural Networks)
- PyBrain (http://www.pybrain.org)
  - Linux; Python
  - Reinforcement Learning, Sequential Learning, Q-Learning, Recurrent Neural Networks, Belief Networks

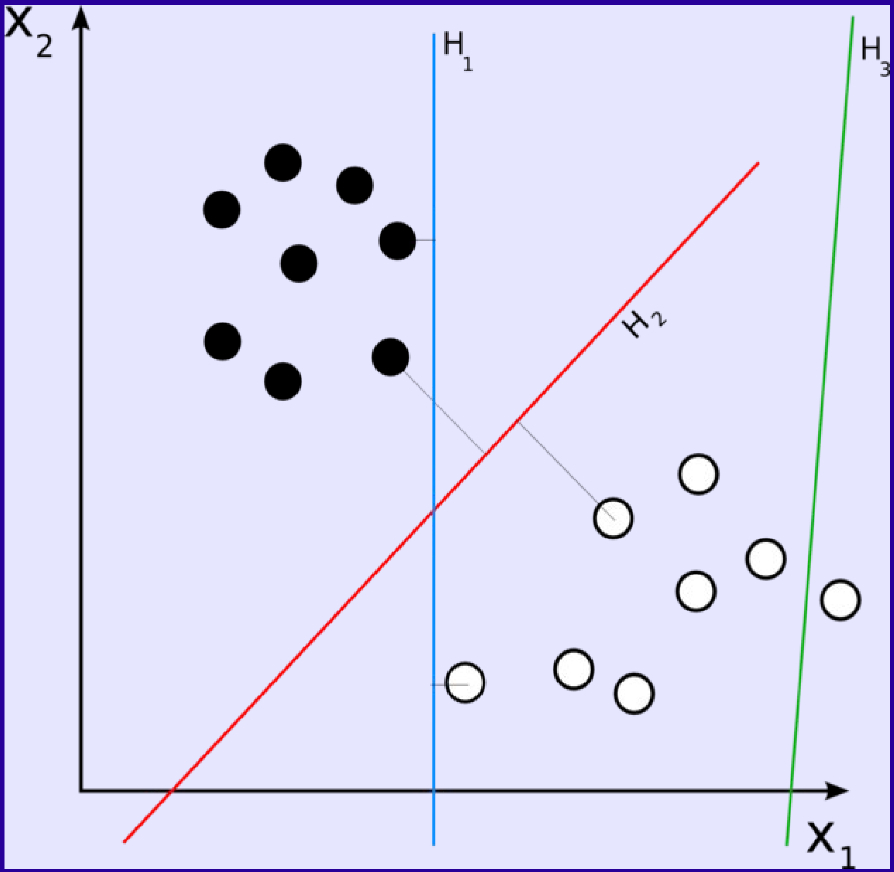
Most Popular Classifiers

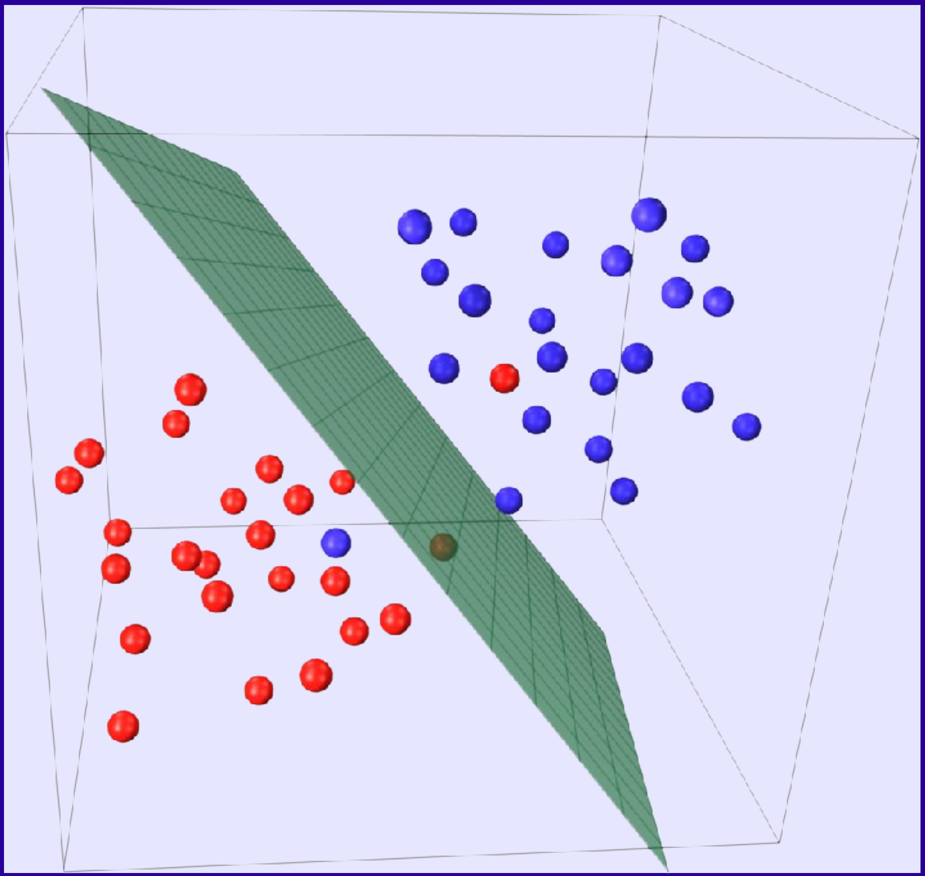
- K Nearest Neighbors (kNN)
- Support Vector Machines (SVM)
- Decision Trees (DT)
- Random Forests (Ensemble of DTs)
- Cascades (e.g. Haar face detector)
- Artificial Neural Networks (ANN)
- Adaptive Boosting (AdaBoost)

k-Nearest Neighbors (kNN) (no training required!)



Support Vector Machines (SVM)

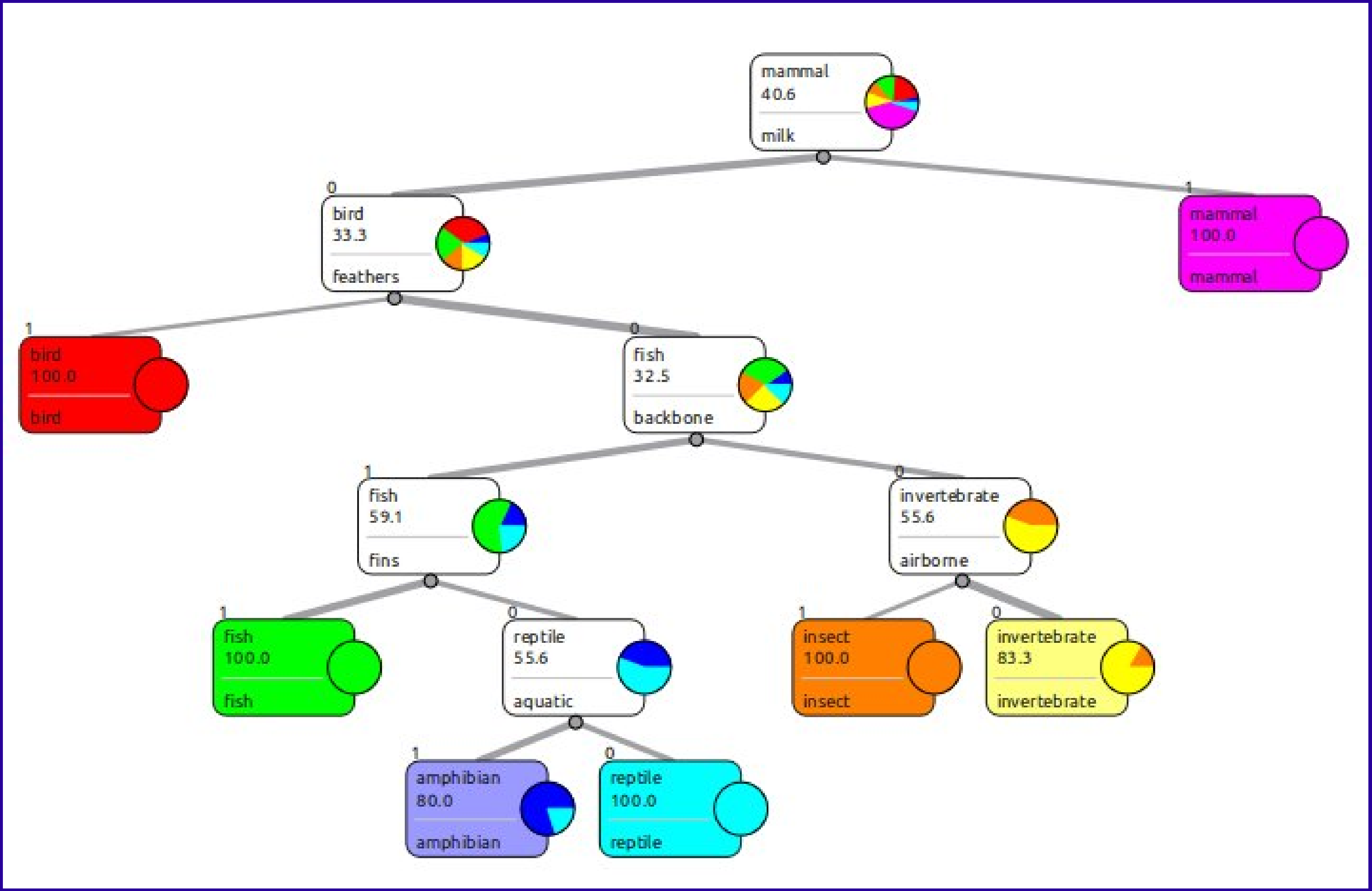




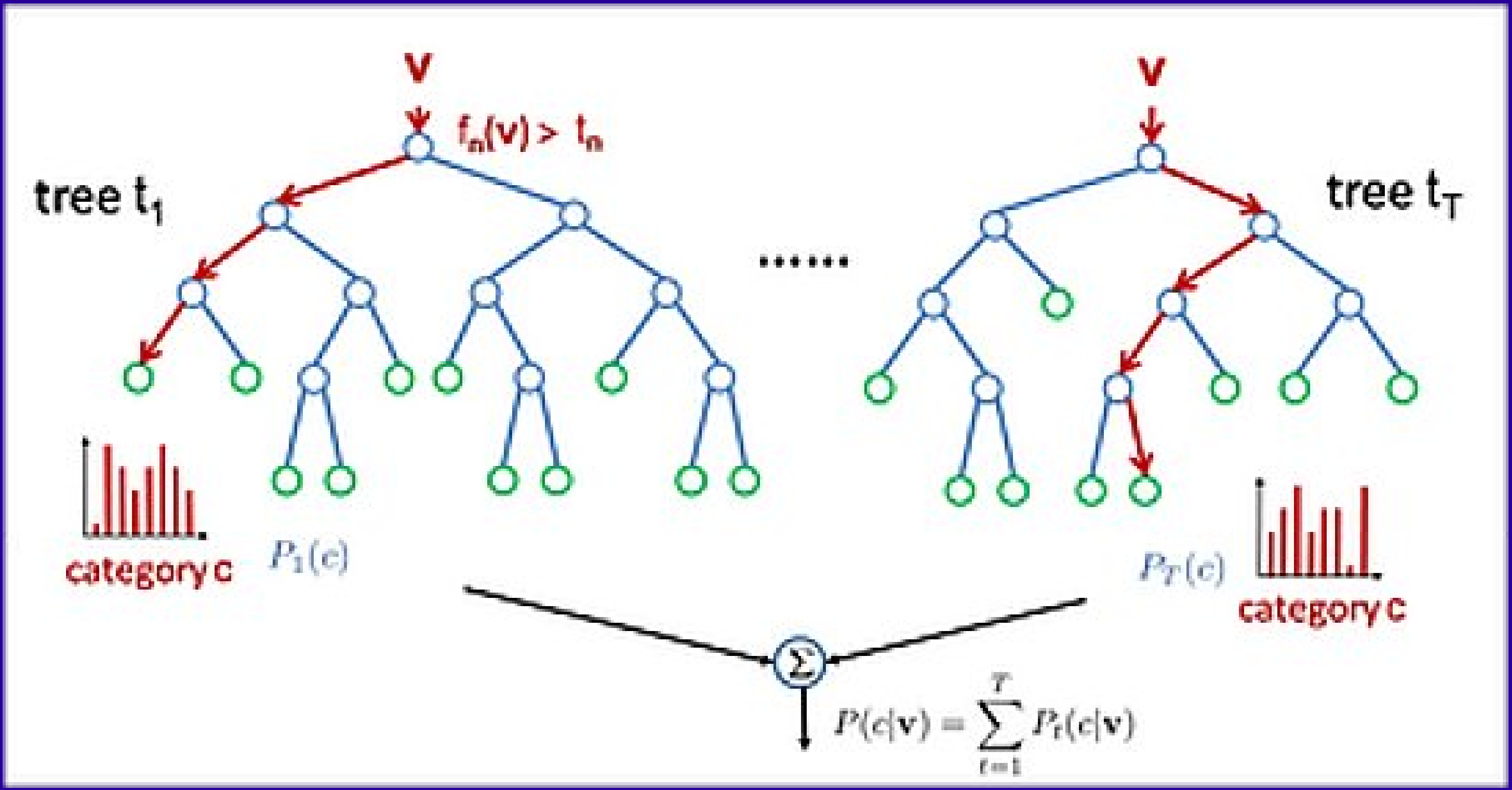
Decision Tree Data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Name	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed	backbone	breathes	venomous	fins	tail	domestic	catsize	type
2	aardvark	1	0	0	1	0	0	1	1	1	1	0	0	0	0	1	mammal
3	antelope	1	0	0	1	0	0	0	1	1	1	0	0	1	0	1	mammal
4	bass	0	0	1	0	0	1	1	1	1	0	0	1	1	0	0	fish
5	bear	1	0	0	1	0	0	1	1	1	1	0	0	0	0	1	mammal
6	boar	1	0	0	1	0	0	1	1	1	1	0	0	1	0	1	mammal
7	buffalo	1	0	0	1	0	0	0	1	1	1	0	0	1	0	1	mammal
8	calf	1	0	0	1	0	0	0	1	1	1	0	0	1	1	1	mammal
9	carp	0	0	1	0	0	1	0	1	1	0	0	1	1	1	0	fish
10	catfish	0	0	1	0	0	1	1	1	1	0	0	1	1	0	0	fish
11	cavy	1	0	0	1	0	0	0	1	1	1	0	0	0	1	0	mammal
12	cheetah	1	0	0	1	0	0	1	1	1	1	0	0	1	0	1	mammal
13	chicken	0	1	1	0	1	0	0	0	1	1	0	0	1	1	0	bird
14	chub	0	0	1	0	0	1	1	1	1	0	0	1	1	0	0	fish
15	clam	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	invertebrate
16	crab	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	invertebrate
17	crayfish	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	invertebrate
18	crow	0	1	1	0	1	0	1	0	1	1	0	0	1	0	0	bird
19	deer	1	0	0	1	0	0	0	1	1	1	0	0	1	0	1	mammal
20	dogfish	0	0	1	0	0	1	1	1	1	0	0	1	1	0	1	fish
21	dolphin	0	0	0	1	0	1	1	1	1	1	0	1	1	0	1	mammal
22	dove	0	1	1	0	1	0	0	0	1	1	0	0	1	1	0	bird
23	duck	0	1	1	0	1	1	0	0	1	1	0	0	1	0	0	bird
24	elephant	1	0	0	1	0	0	0	1	1	1	0	0	1	0	1	mammal

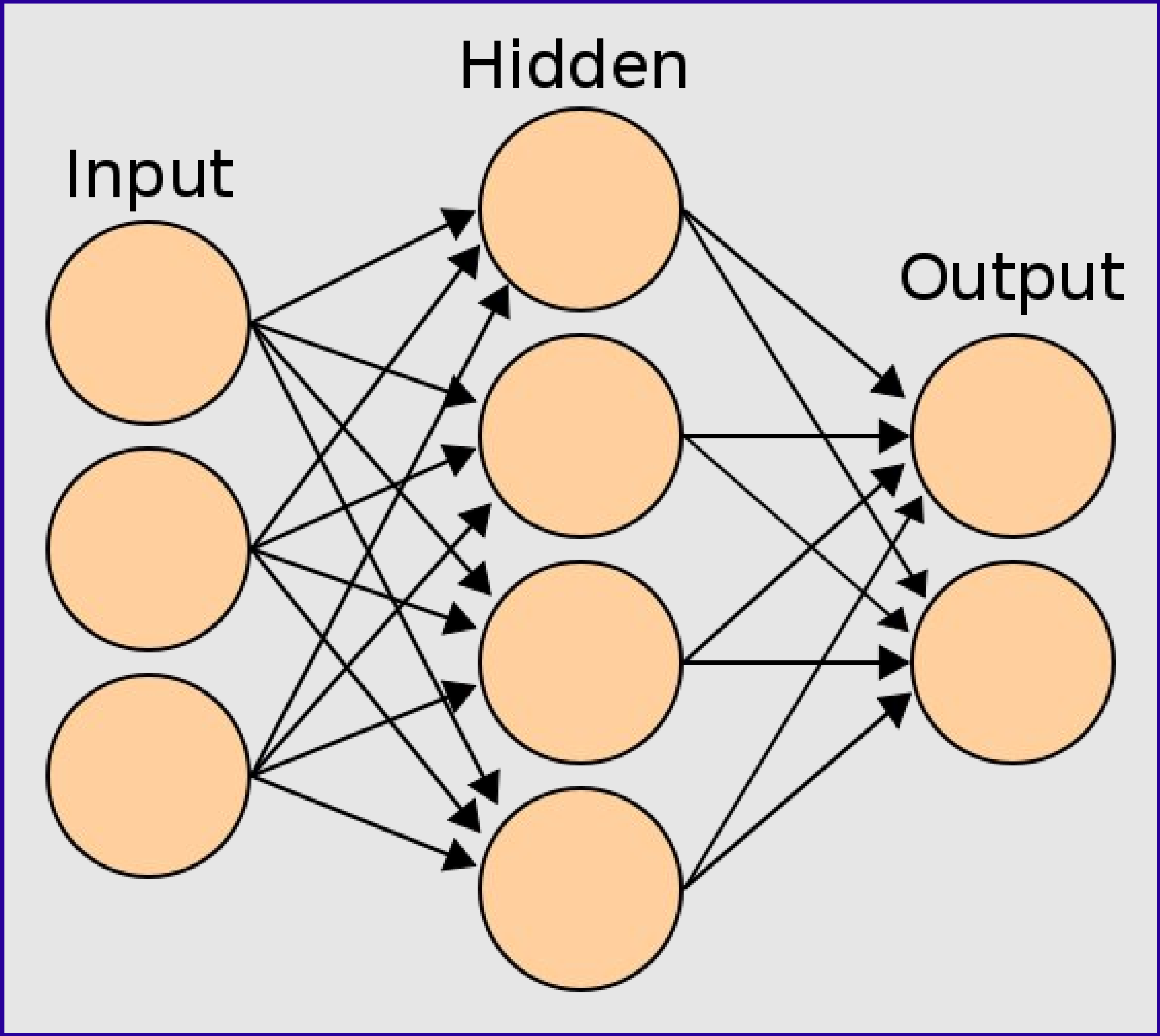
Decision Tree Classifier



Random Forests



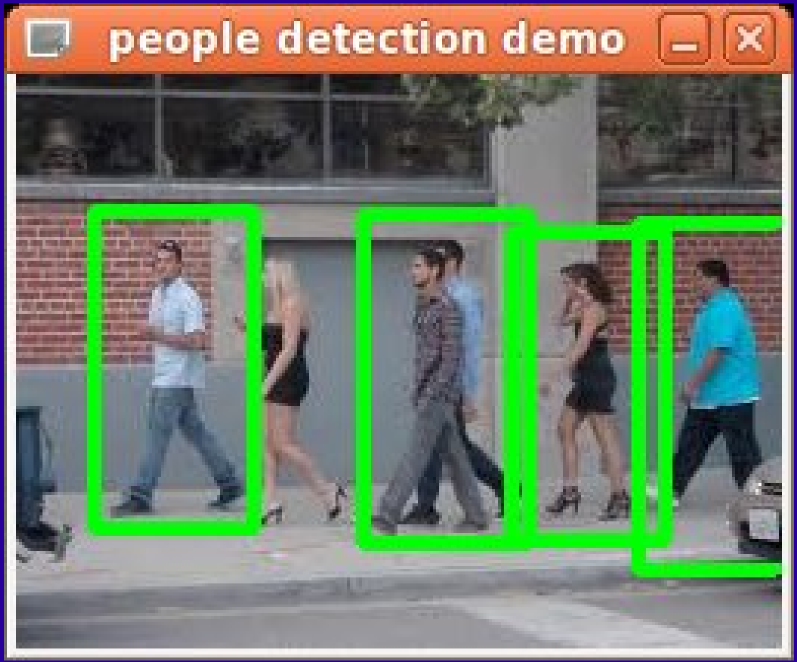
Artificial Neural Networks (ANN)



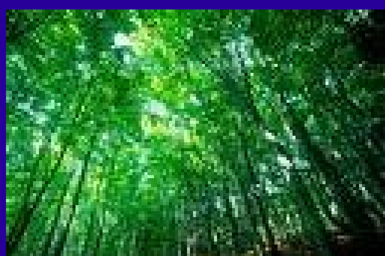
Prebuilt Classifiers in OpenCV
Face & People Detectors

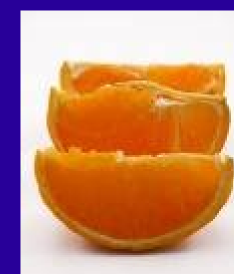
- Haar Face Detector

- HOG Person Detector

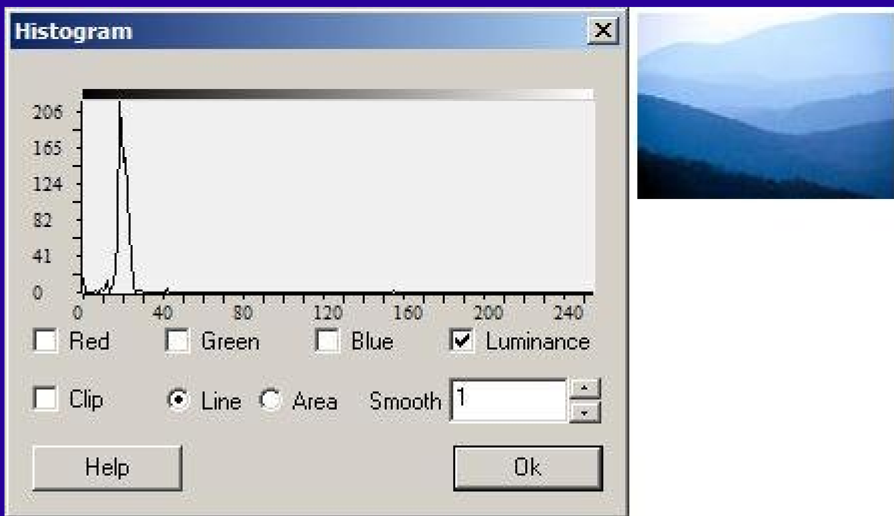


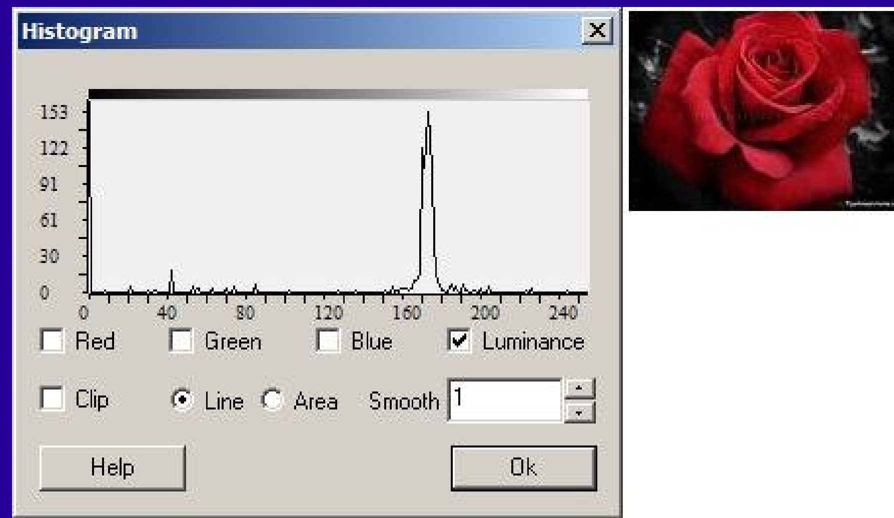
Automated Color Naming

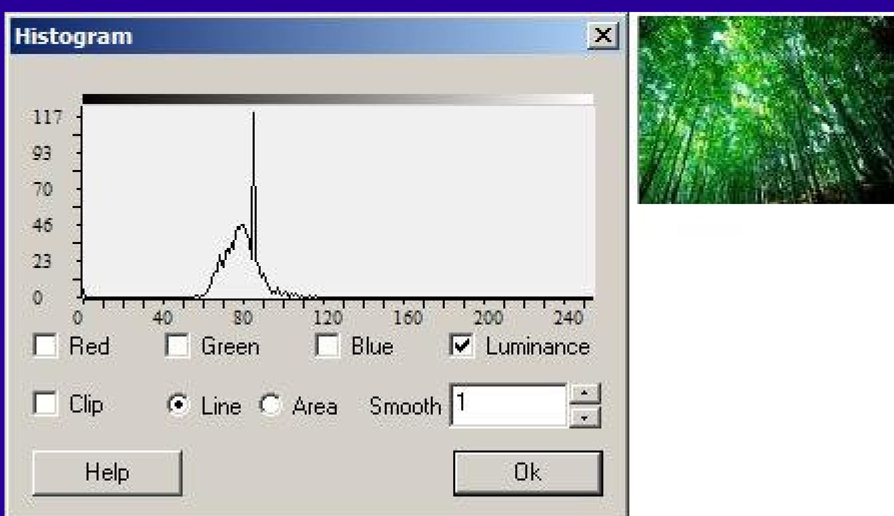


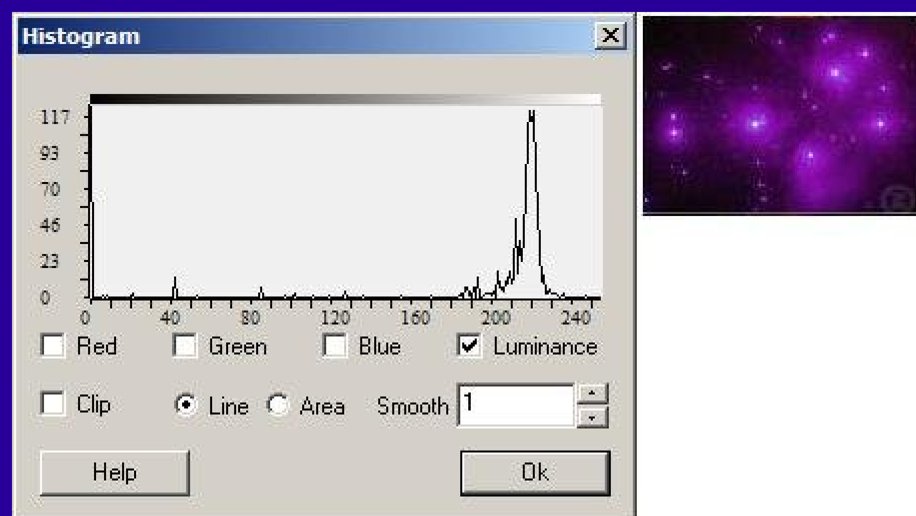


Hue Histograms
RGB → HSV → Hue





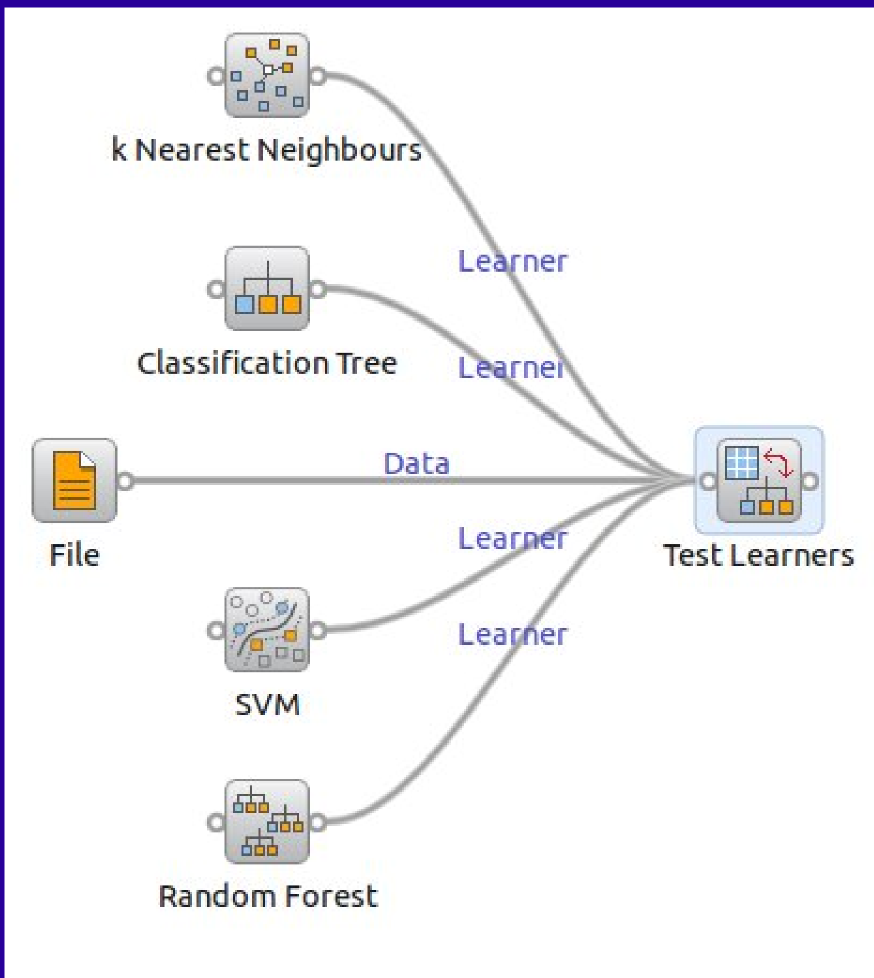


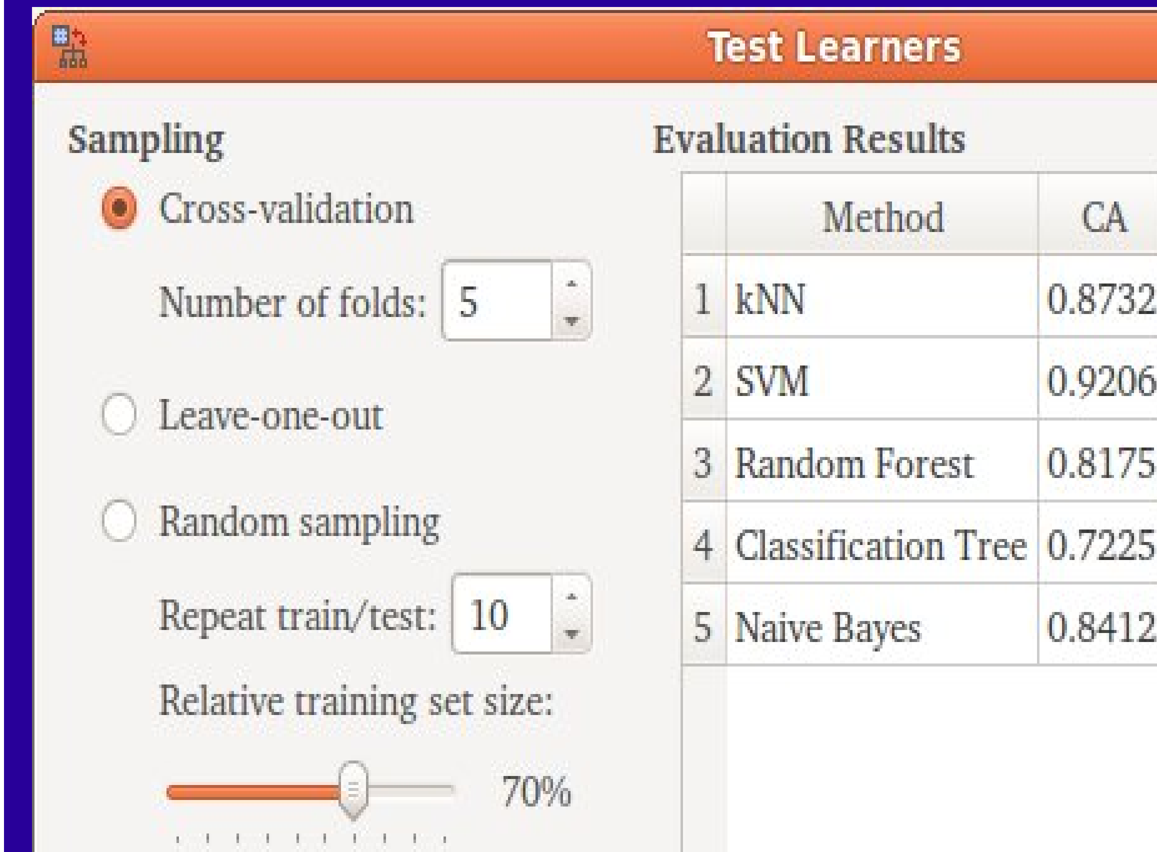


Preparing the Color Data

- Collect color images, say 20 images per color
- Store images in folders named after each color
- Run training script:
  - For each color folder
    - Read image
    - Convert to HSV
    - Compute Hue histogram
    - Add color label and histogram values to tab-delimited data file

Test Multiple Learners with Orange

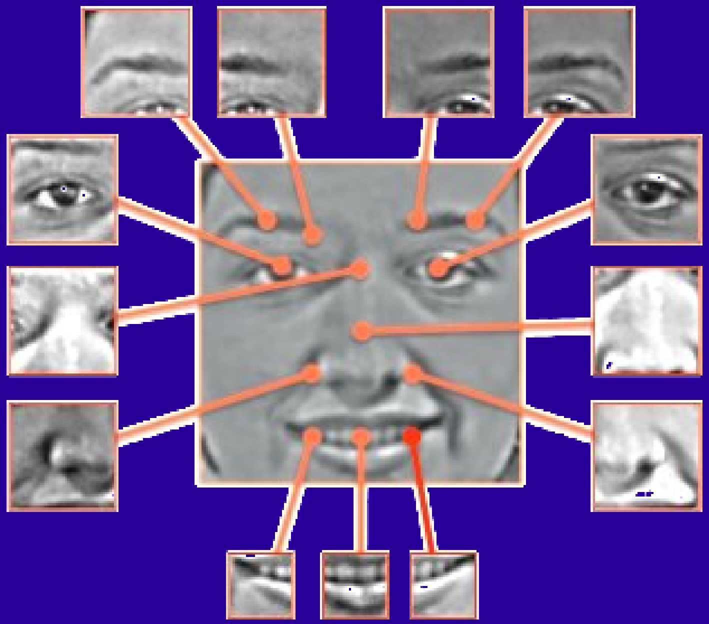


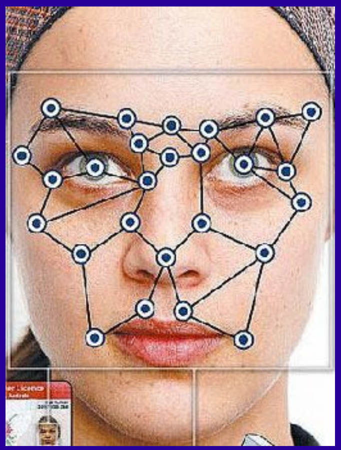


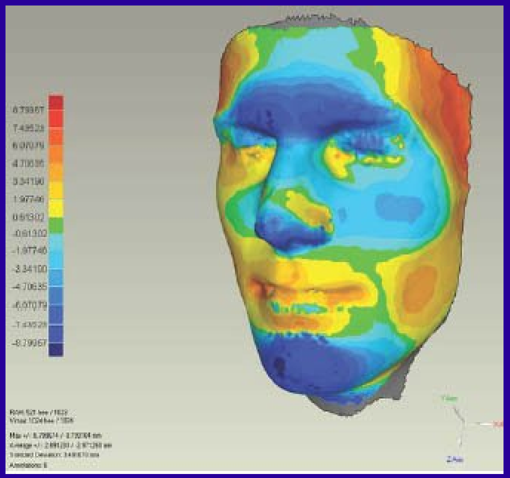
What's in a Face?

What are face features?

- Eyes, ears, nose, mouth?
- Outline, hair color?
- Geometry?
- Template?
- SURF descriptors?
- Eigenfaces?
- Fisher Faces?





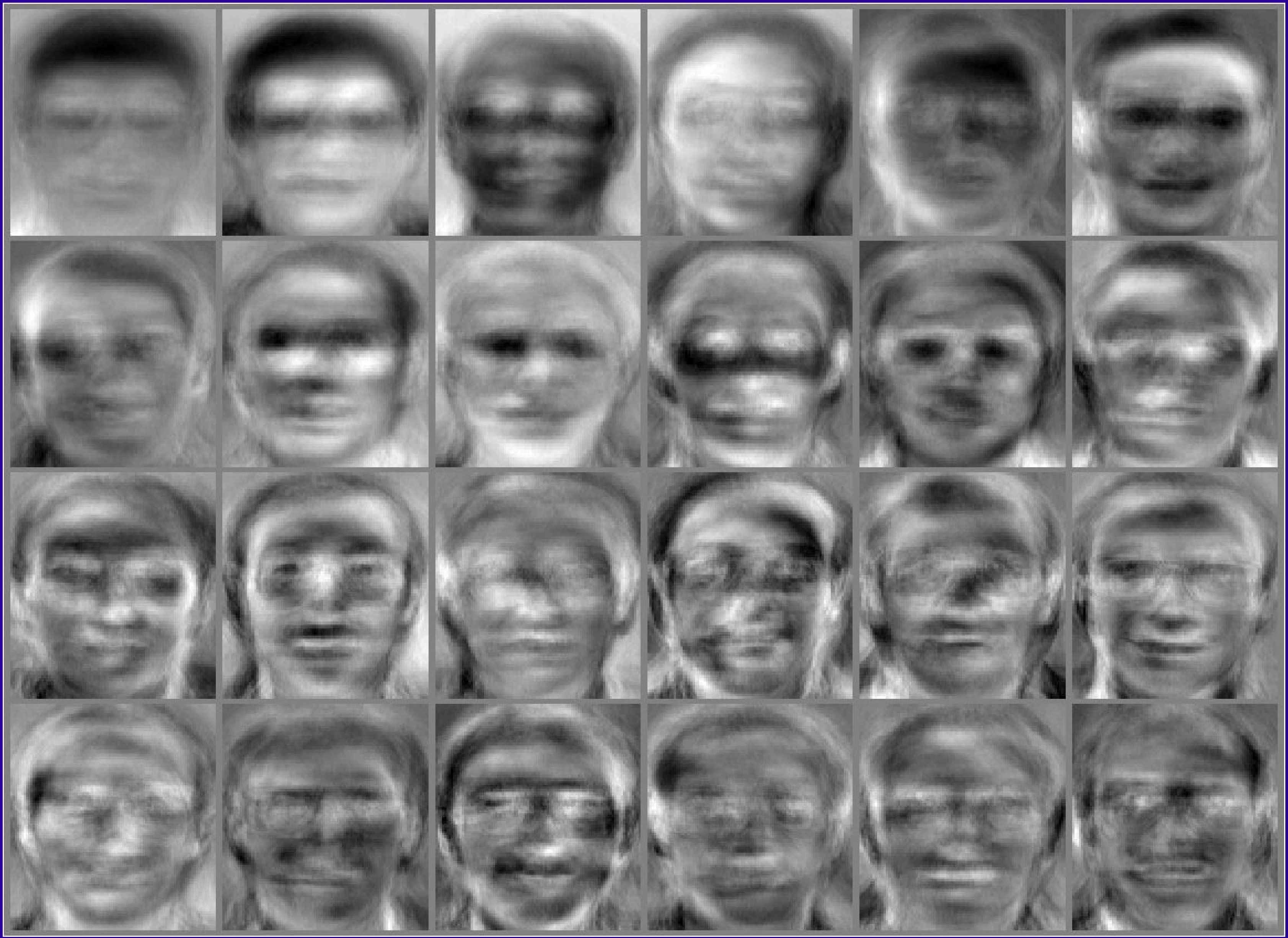


The AT&T / ORL Face Database

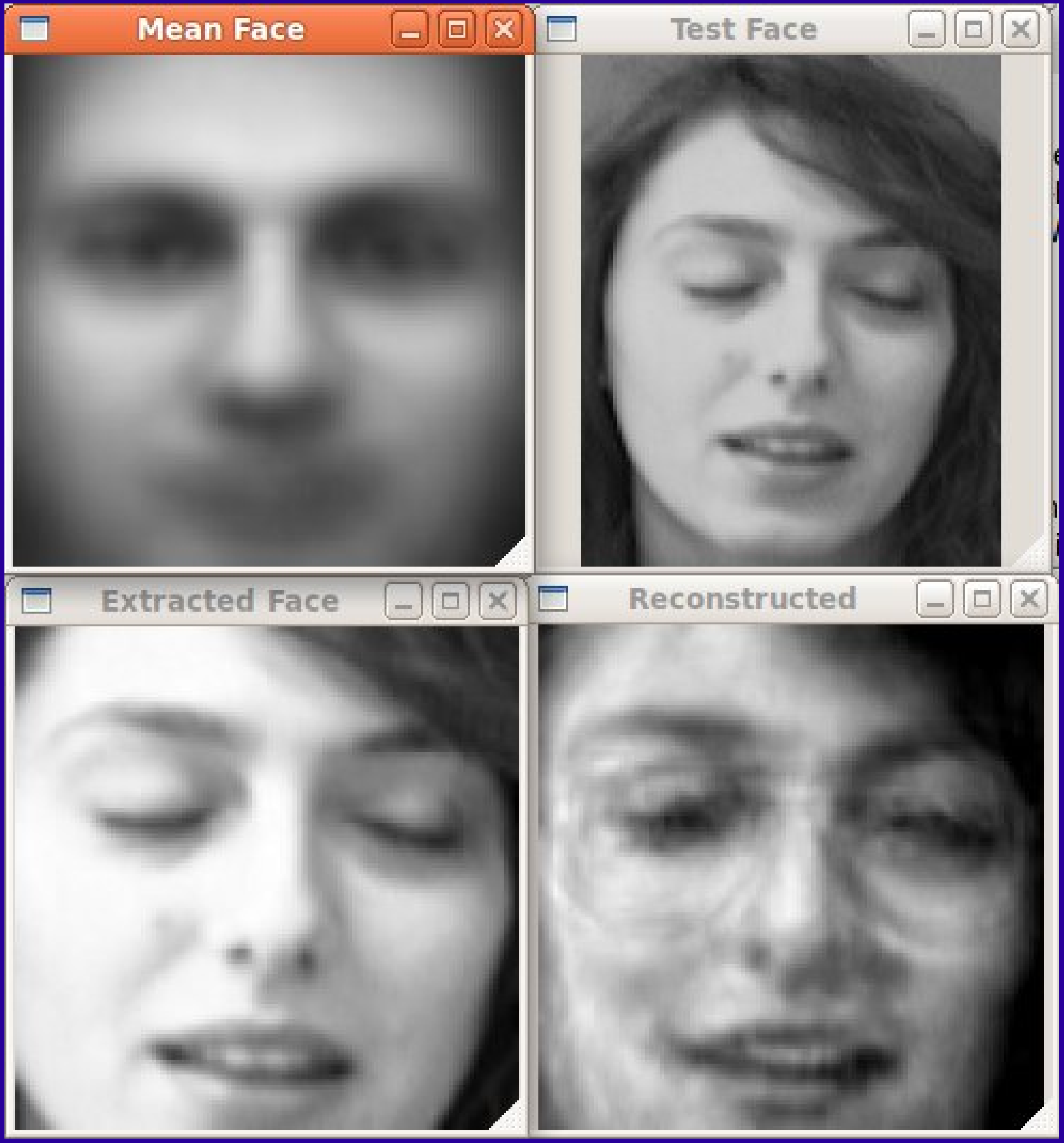
Using Eigenfaces for Face Recognition

- Convert each W x H image to a (W x H)-d vector by concatenating the rows; e.g. 100x100 image → 10,000-d vector
- Run a Principle Components Analysis (PCA) on all training vectors → returns eigenvectors
- Keep N eigenvectors with largest eigenvalues; e.g. N = 64
- Project sample images onto this feature space: converts each 10,000 element vector to a 64 element vector
- Build a classifier using these feature vectors

Eigenfaces



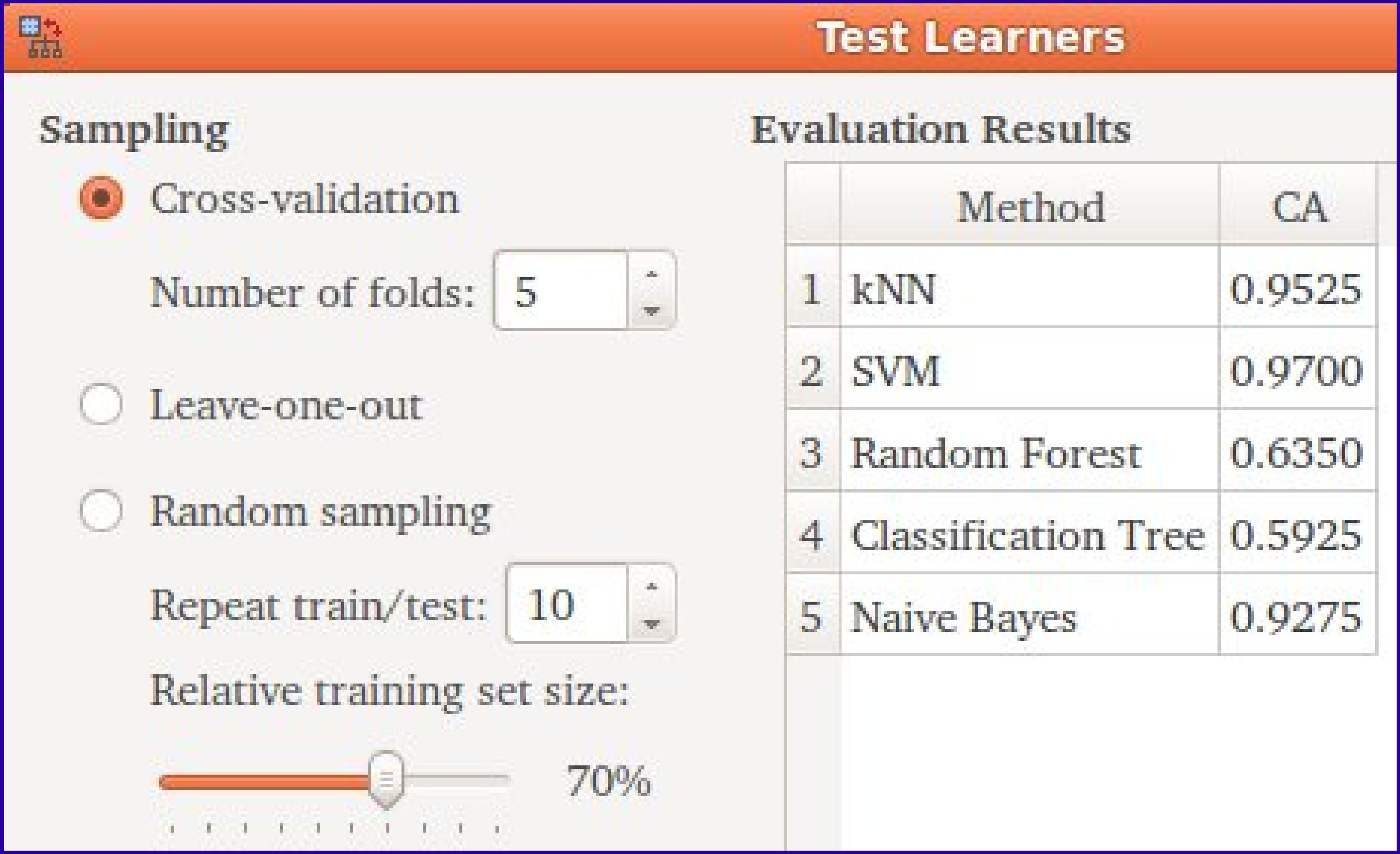
Average and Reconstructed Faces using PCA (64 eigenvectors)



Eigenface Performance on AT&T faces

40 people, 10 images per person

(ANN = 94% correct)



	Method	CA
1	kNN	0.9525
2	SVM	0.9700
3	Random Forest	0.6350
4	Classification Tree	0.5925
5	Naive Bayes	0.9275

The U Sheffield Faces

20 people, 20-40 images per person

Test Learners

Sampling

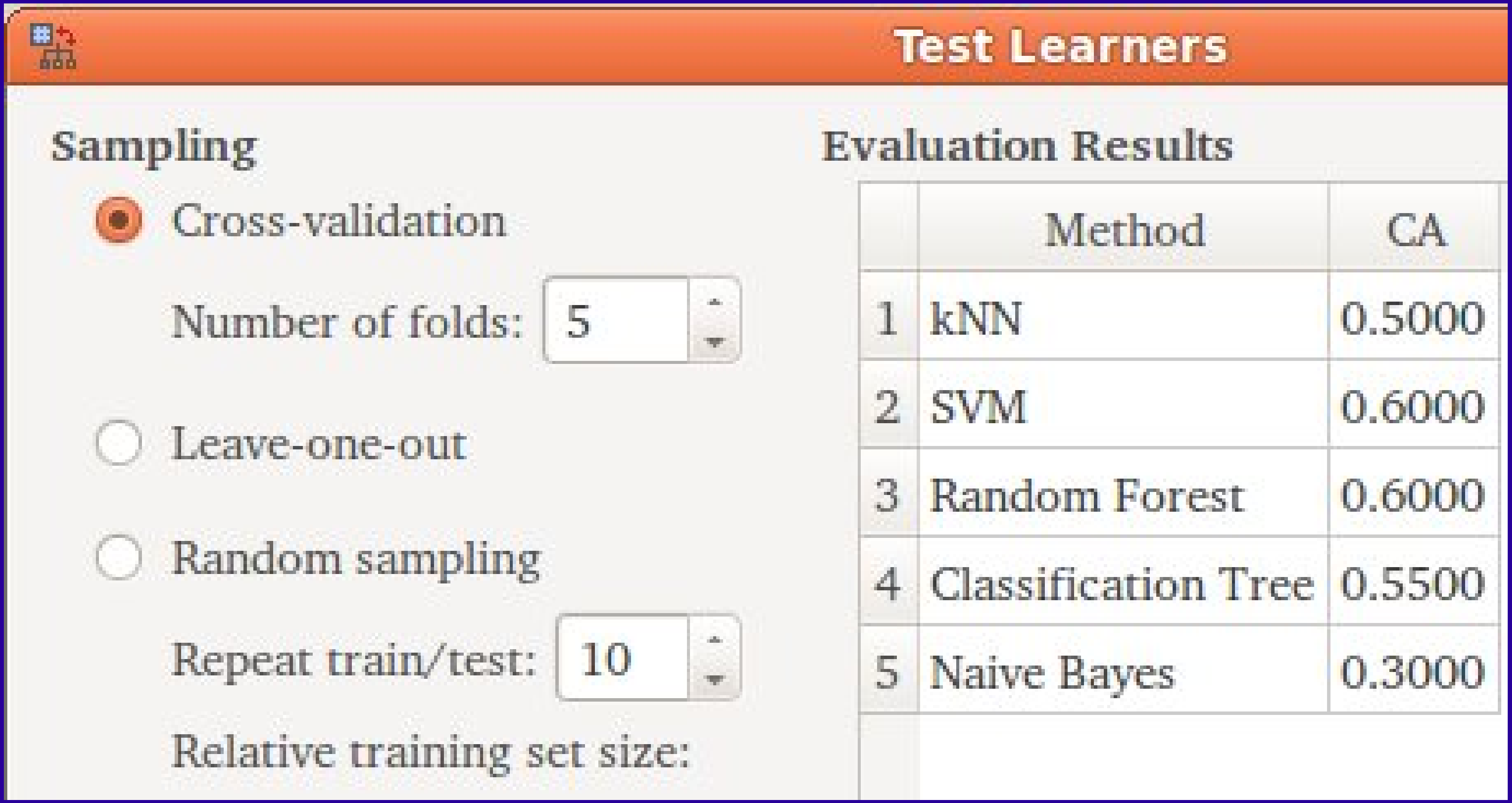
- (selected) Cross-validation
  - Number of folds: 5
- Leave-one-out
- Random sampling
  - Repeat train/test: 10
  - Relative training set size:

Evaluation Results

	Method	CA
1	kNN	0.9878
2	SVM	0.9913
3	Random Forest	0.8557
4	Classification Tree	0.7826
5	Naive Bayes	0.9496

Labeled Faces in the Wild

Eigenface Failure...



Alternatives to Eigenfaces

Your Contribution Goes Here :-)

- PCL: Viewpoint Feature Histograms (VFH)
- SIFT/SURF + PCA
- Biometrics
- 3D Modeling

Real Time Classifier Learning

- Select an object any way you can
  - manual selection
  - existing detector (e.g. Haar face detector)
  - motion detection
- Track keypoints using Optical Flow
- Grab positive and negative samples as you go
- Build a custom classifier from current samples
- Re-detect object using classifier
- Store custom classifier for later detection

Face Tracking using Optical Flow

Video courtesy of Honda/UCSD Video Database

Pi Face Tracker Video

http://youtu.be/Yw_zkLaZNsQ

Predator Algorithm (OpenTLD) Zdenek Kalal, Jan 2011

[Predator Track-Learn-Detect (TLD)](http://youtu.be/1GhNXHCQGsM)

http://youtu.be/1GhNXHCQGsM

Thanks for Listening!