A Trainable View-Based Object Detection System

Thesis Proposal

Henry A. Rowley

Thesis Committee:

Takeo Kanade, Chair
Shumeet Baluja
Dean Pomerleau
Manuela Veloso
Tomaso Poggio, MIT

Motivation

Object detection is fundamental to computer vision.

Many potential applications:

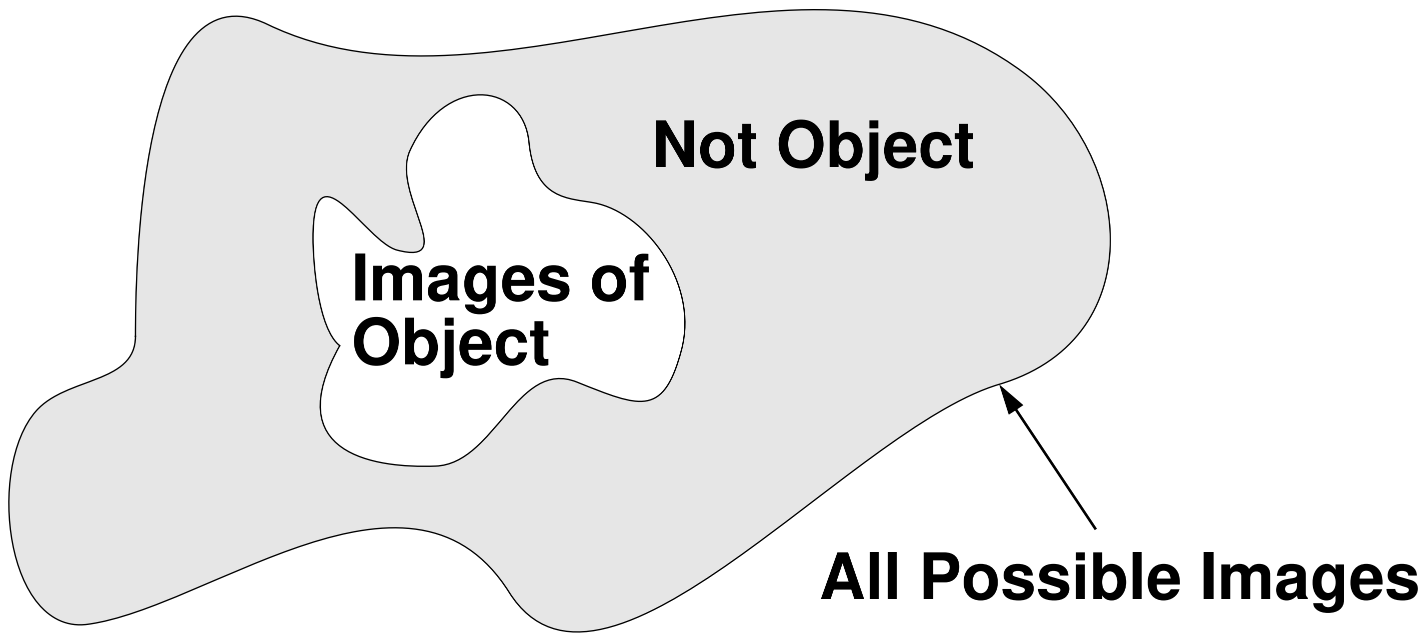
- **Indexing / searching images by content**
- **Video summarization**
- **Face recognition / security systems**
- **Mobile robotics**

Outline of Talk

- Motivation
- What is Object Detection?
- Results to Date:
    - Frontal Face Detection
    - Car (Tire) Detection
- Related Work
- Expected Contributions
- Timetable

What is Object Detection?

Formally: Object detection is deciding if a new image belongs to the set of images of an object.



Conservative assumption: Increasing object variability increases difficulty of detection problem.

Sources of Variability

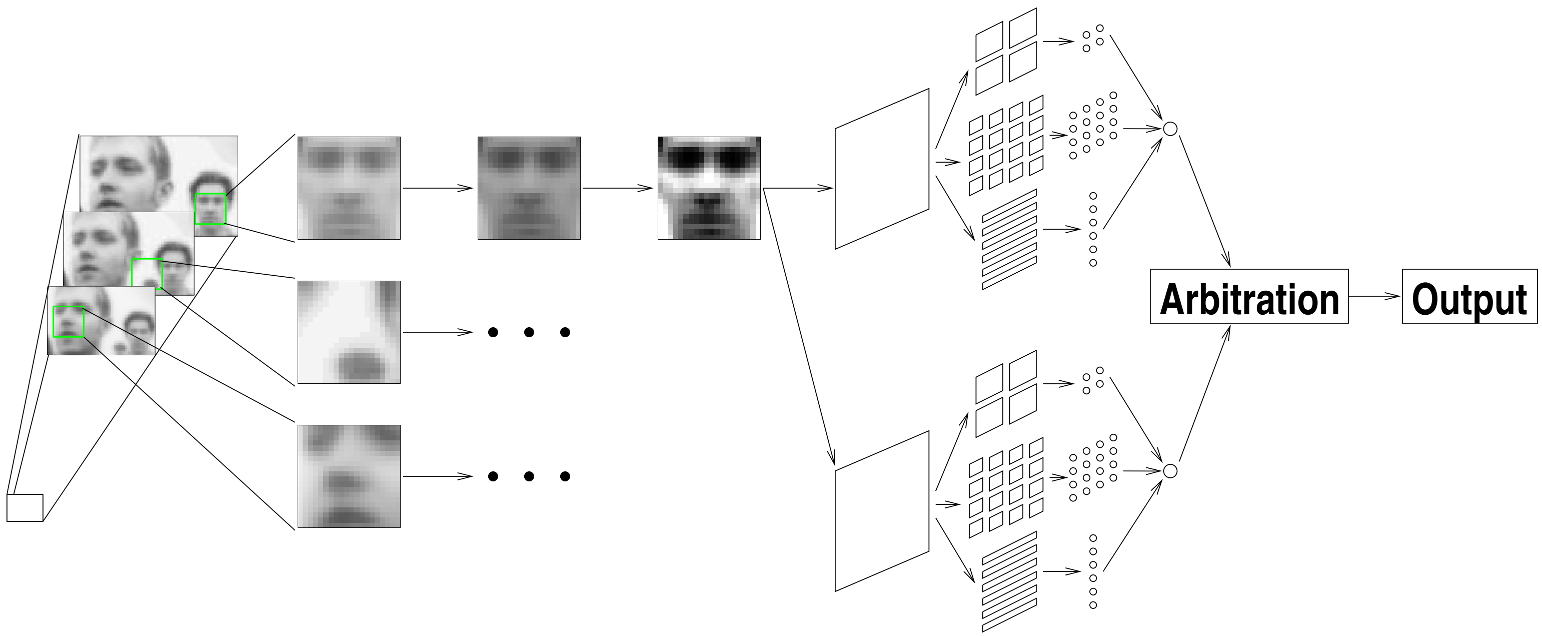
- Image plane variation (rotation, translation, scale)
- Object pose (3D rotation, distance from camera)
- Lighting and surface appearance / texture
- Background variation
- Shape variation (within class: cars or chairs)
- Shape variation (within object: articulated motion)

Building an Object Detector

- How to partition problem?
  - Separating images by pose (profile / frontal face)
  - Sub-features of object (eyes, nose, mouth)
- How to do classification?
  - Preprocessing of images
  - Type of classifier
  - Training procedure
- How to merge results?
  - Graph matching
  - Statistical methods

Architecture of Frontal Face Detector

- Extract 20 x 20 pixel windows from the image
- Preprocess windows to improve contrast / lighting
- Apply (multiple) neural networks for classification
- Arbitrate among networks



Extracting 20 x 20 Windows

To make detection simpler, just detect faces centered in, and filling, 20 x 20 pixel windows.

Preprocessing Windows

Lighting and contrast may be poor in the images.

Original window:

Best fit plane:

Original minus best fit plane:

Apply histogram equalization:

Positive Training Examples

Randomizing Positive Examples

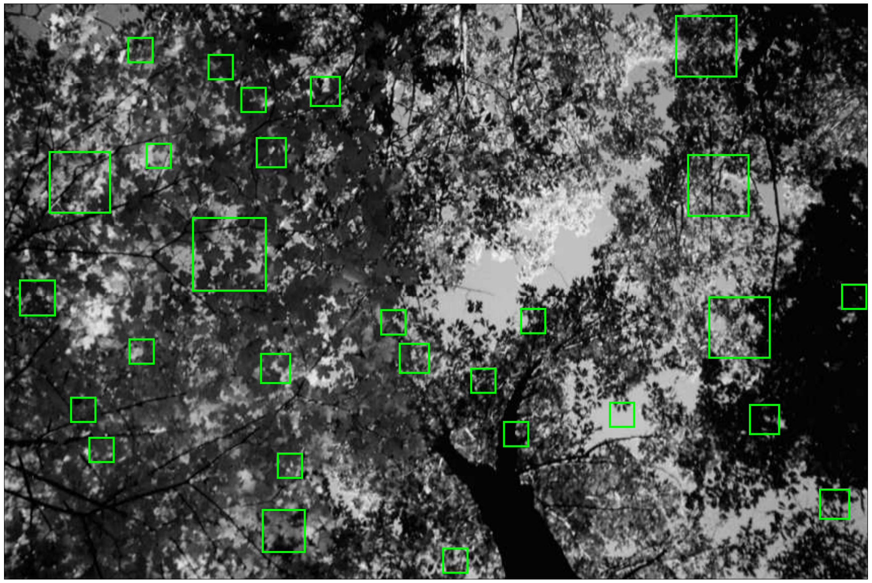
Selecting Negative Examples

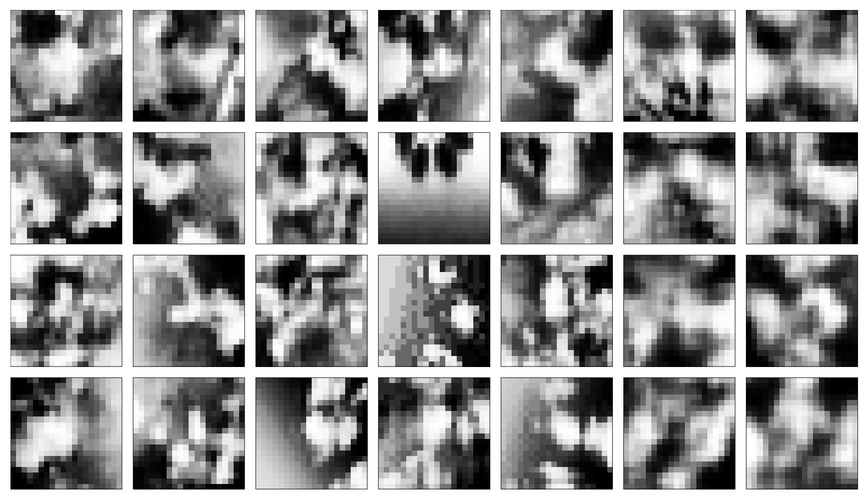
Selecting a representative sample of non-faces is hard.

Active Learning

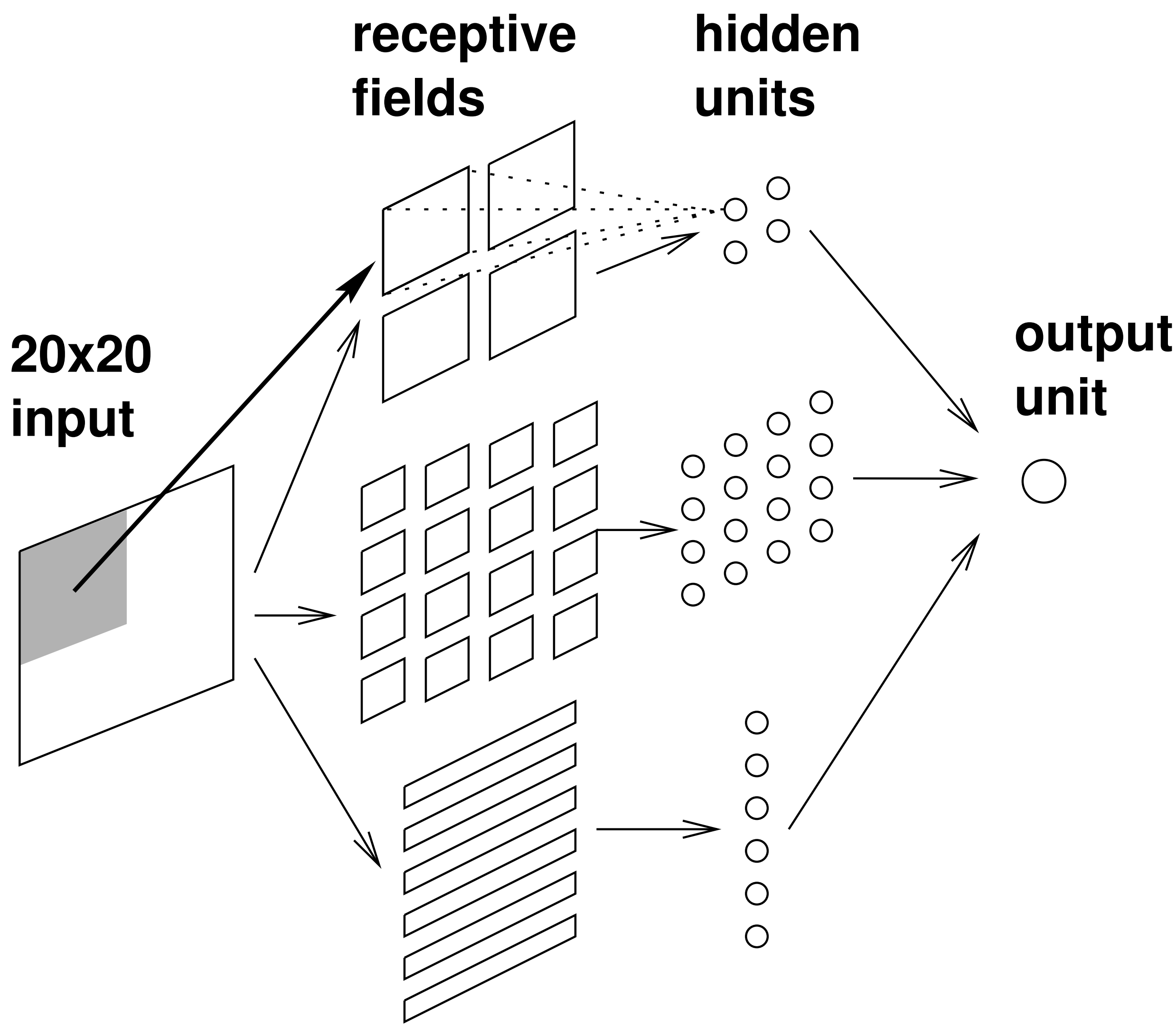
1. Train network on training set.
2. Present an image which contains no faces.
3. See where it makes mistakes.
4. Add mistakes to training set as negative examples.
5. Repeat.

Negative Examples





Network Architecture

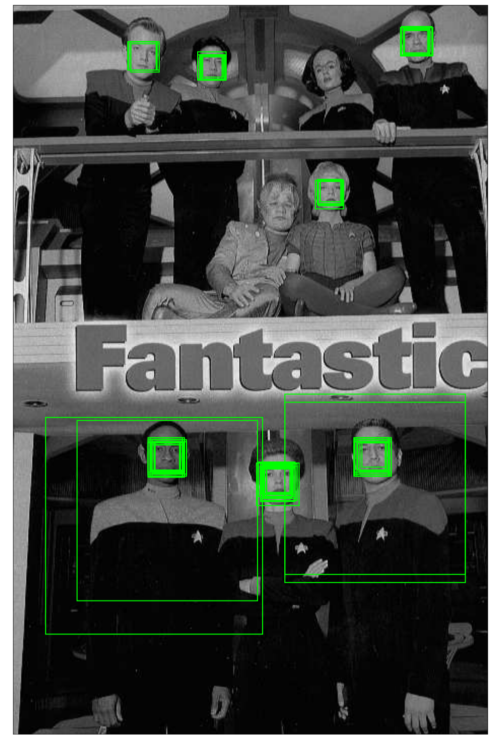


Arbitration Among Multiple Networks

Network 1

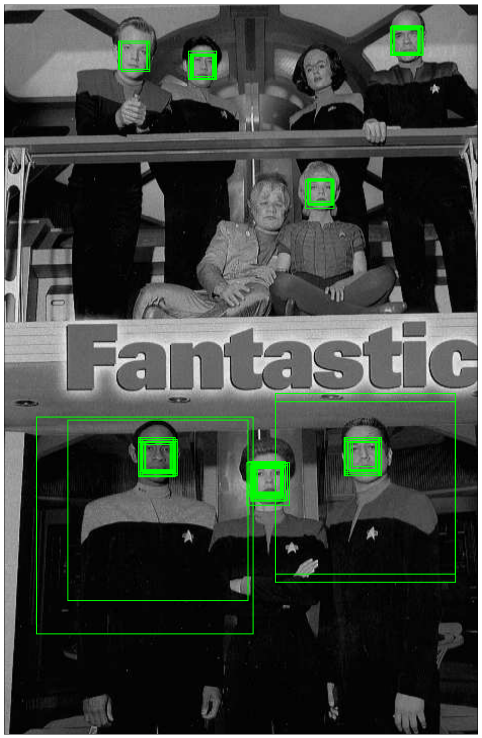
Network 2

AND of networks



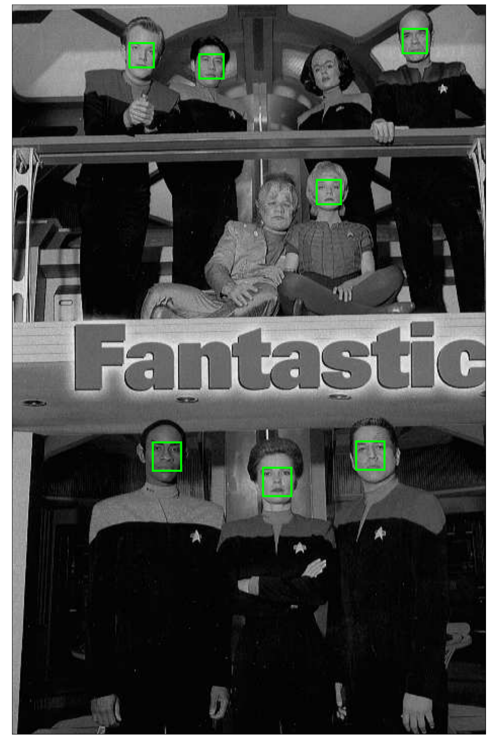
Clean-Up Heuristics

AND of networks



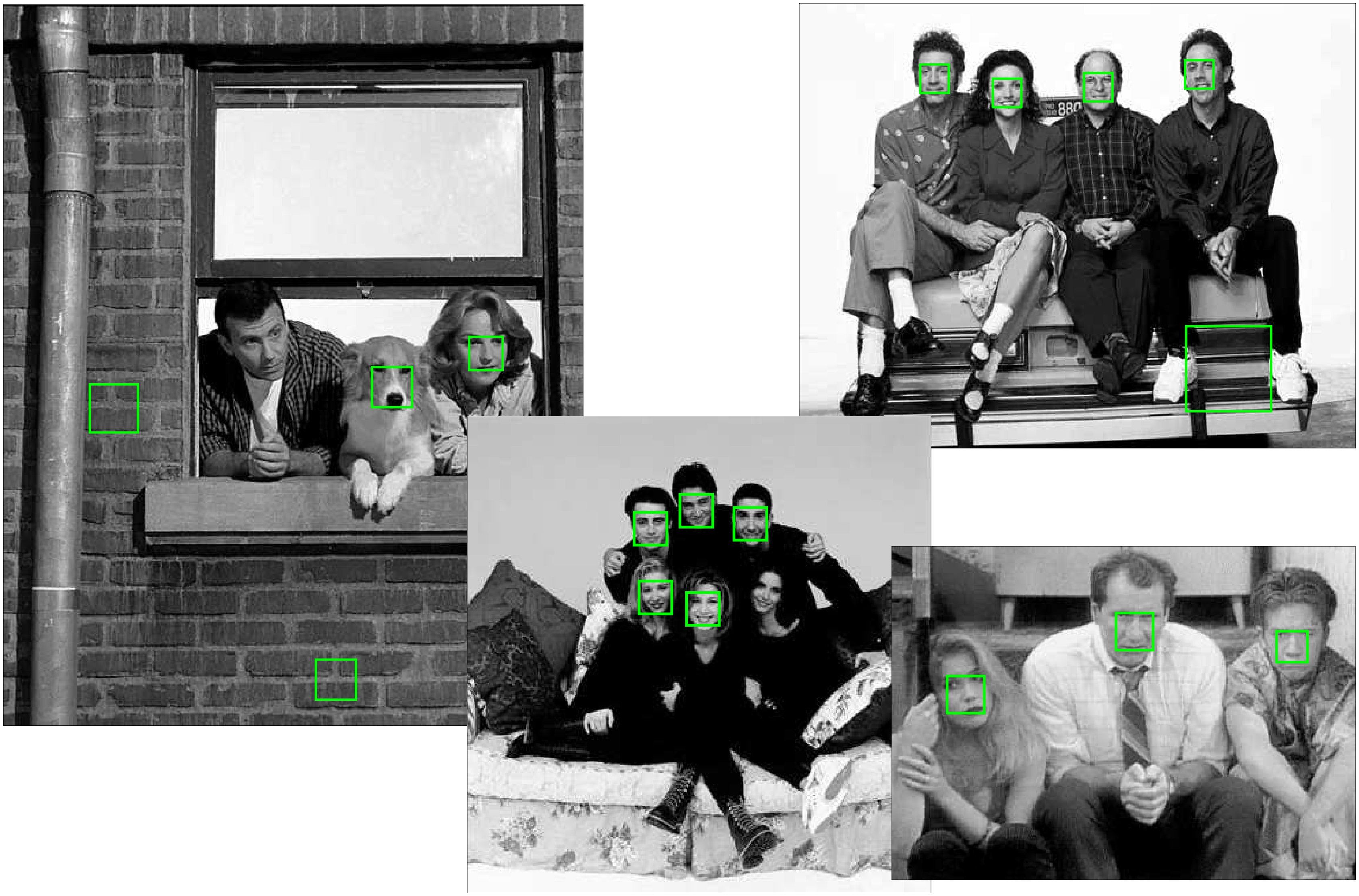
AND + merge detections

AND + merging + remove overlaps

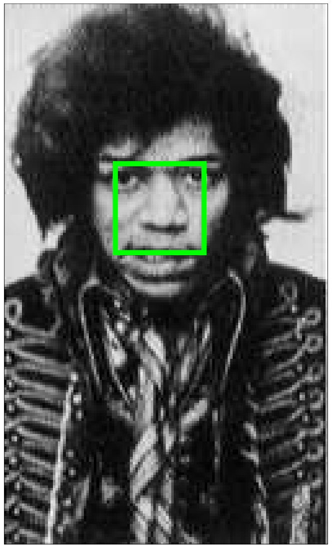


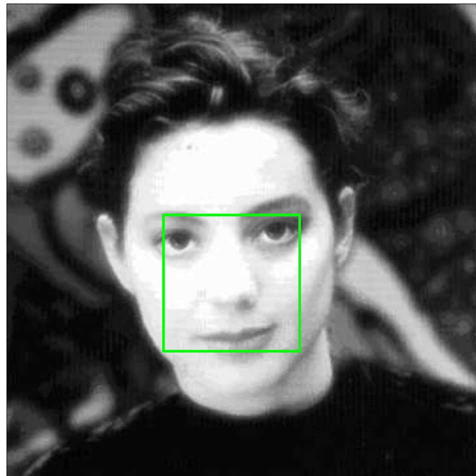
Results: Digitized TV Images

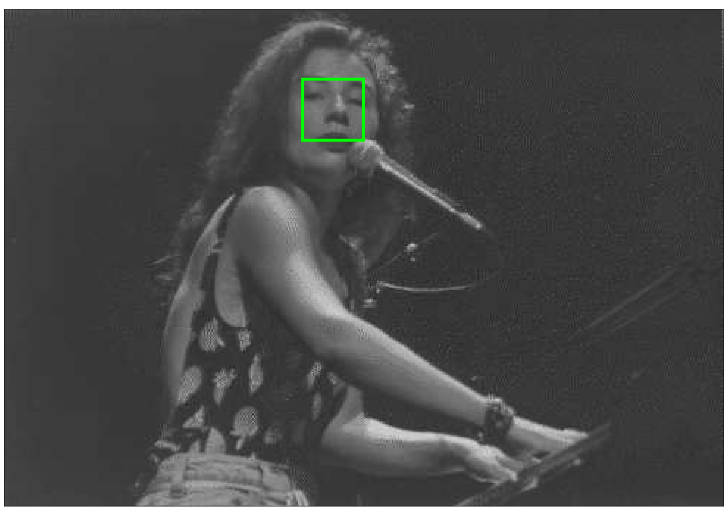
Results: Sitcom Casts



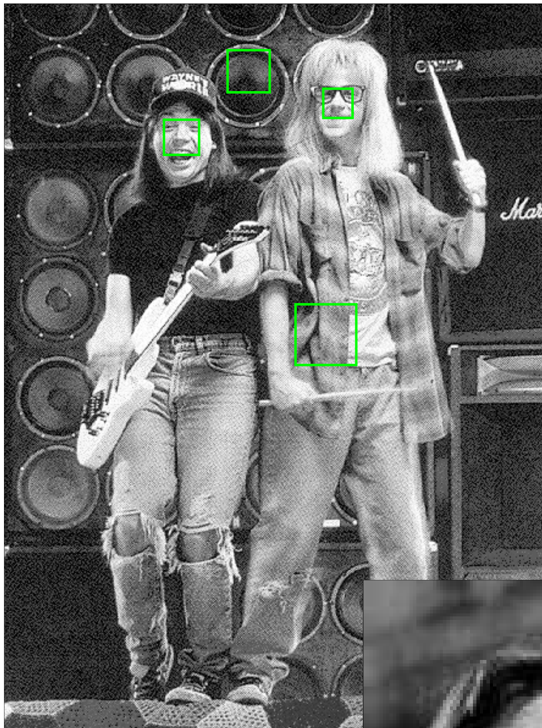
Results: Musicians

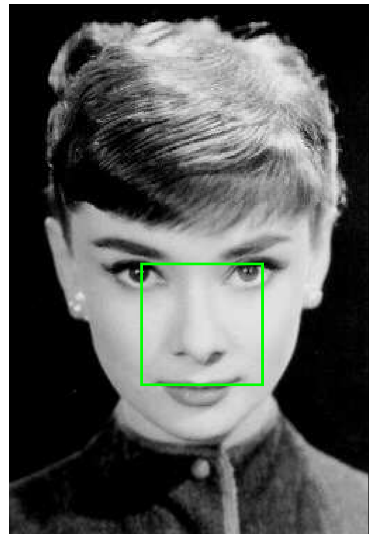


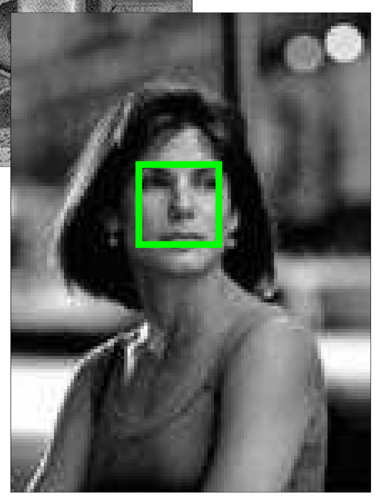


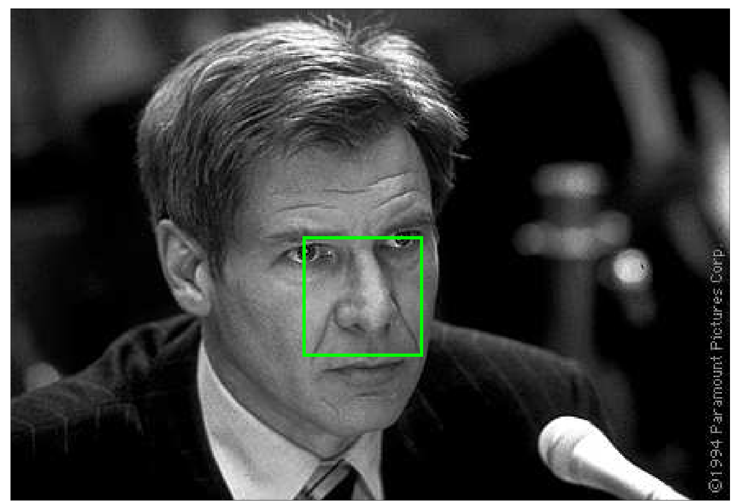


Results: Movie Stars

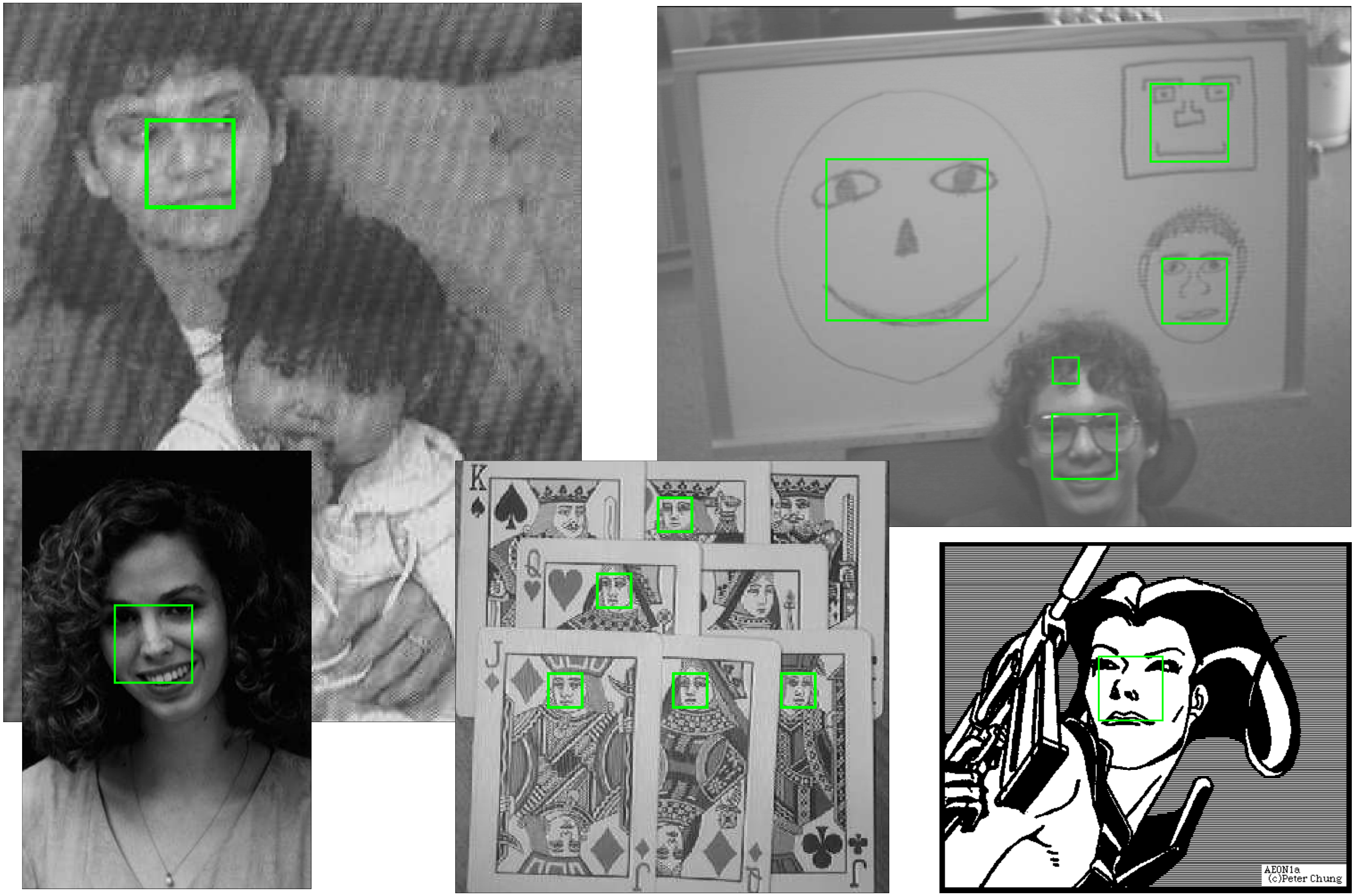




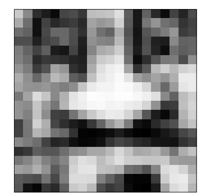


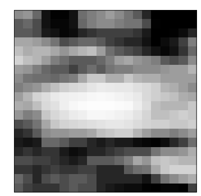


Results: Random Images



Results: Class Picture





Accuracy

Sung & Poggio: 23 images, 155 faces, 9678084 windows

System	Detect Rate %	False Detects
Single network	92.9	353
Single network + heuristics	92.3	126
Two networks (version 1)	78.1	3
Two networks (version 2)	87.1	15
Two networks (version 3)	92.9	64
MIT: PCA/Clustering/MLP	76.8	5
MIT: PCA/Clustering/Perceptron	81.9	13
Fast Version	72.9	3

Version 1: AND network outputs, then apply a threshold and overlap elimination
Version 2: Apply heuristics to networks separately, then AND the results
Version 3: Apply heuristics to networks separately, OR the results, then remove overlaps

Variations on Face Detection

- Speed improvements
- Sub-features of face: Eye detection
- Profiles and other views

Speed Improvements

Processing time for a 320x240 image:
5.5 minutes on an SGI Indigo 2.

Where is the bottleneck?
Must extract 20x20 pixel window from every pixel position and scale.

Solution from license plate detector, Umezaki [1995]:
Do not examine each pixel location.

Speed Improvements

Use the same training procedure, different data:

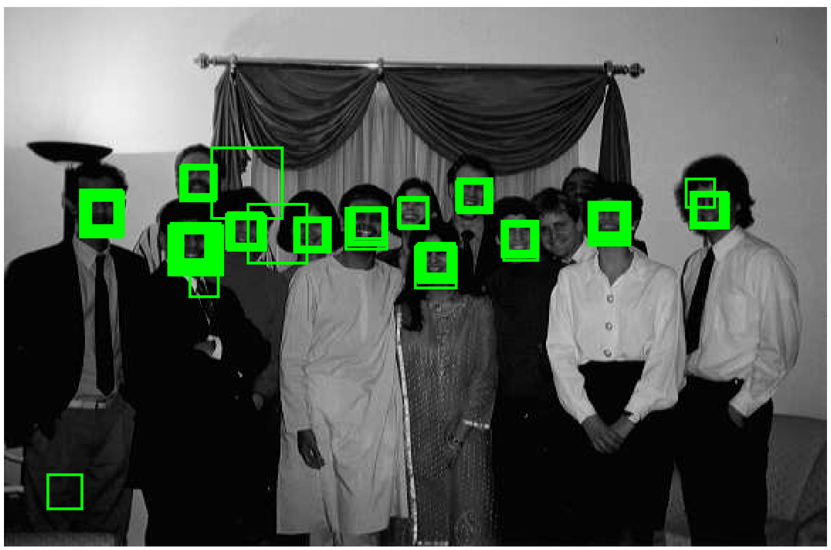
- **Larger input window: 30x30 pixels**
- **Positive examples no longer centered:**

- **Detector moves in steps of 10 pixels over image**
- **Single output indicates presence of a face**

Accuracy of Large Window Detector

Many more false detections than Small Window detector

Small Window Version



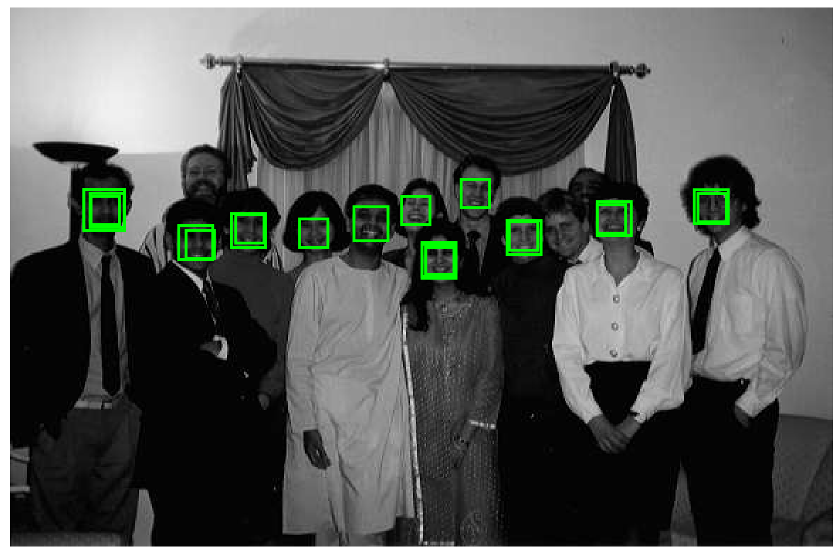
Large Window Version

Improving Accuracy

Treat Large Window detections as candidates.
Verify candidates with Small Window detector.

Candidates

Verified Detections



Speedup

320x240 image, with Large Window detector:

- **9 seconds on an SGI Indigo 2.**
- **Faster by a factor of 35.**

Further Speedups

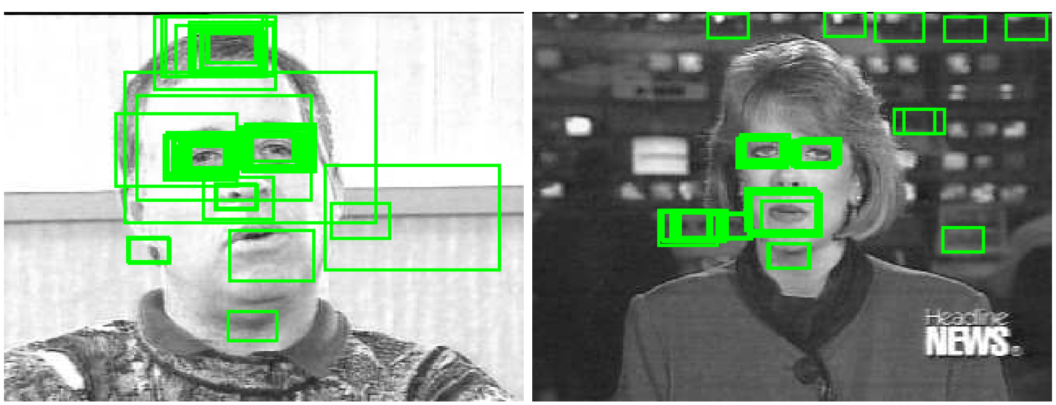
• **Motion**	**3 - 5 seconds**
• **Skin color detection, Yang & Waibel [1995]**	**1 - 2 seconds**
• **Temporal coherence / tracking**	**0.2 seconds**

Sub-features of Face: Eyes

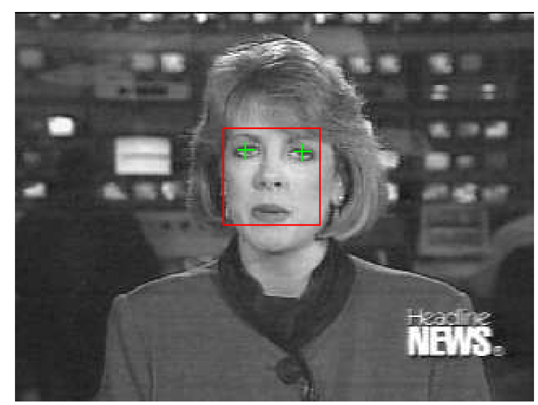
Same training procedure, new data (25x15 windows):

More false detections than for faces:

Eye detector alone

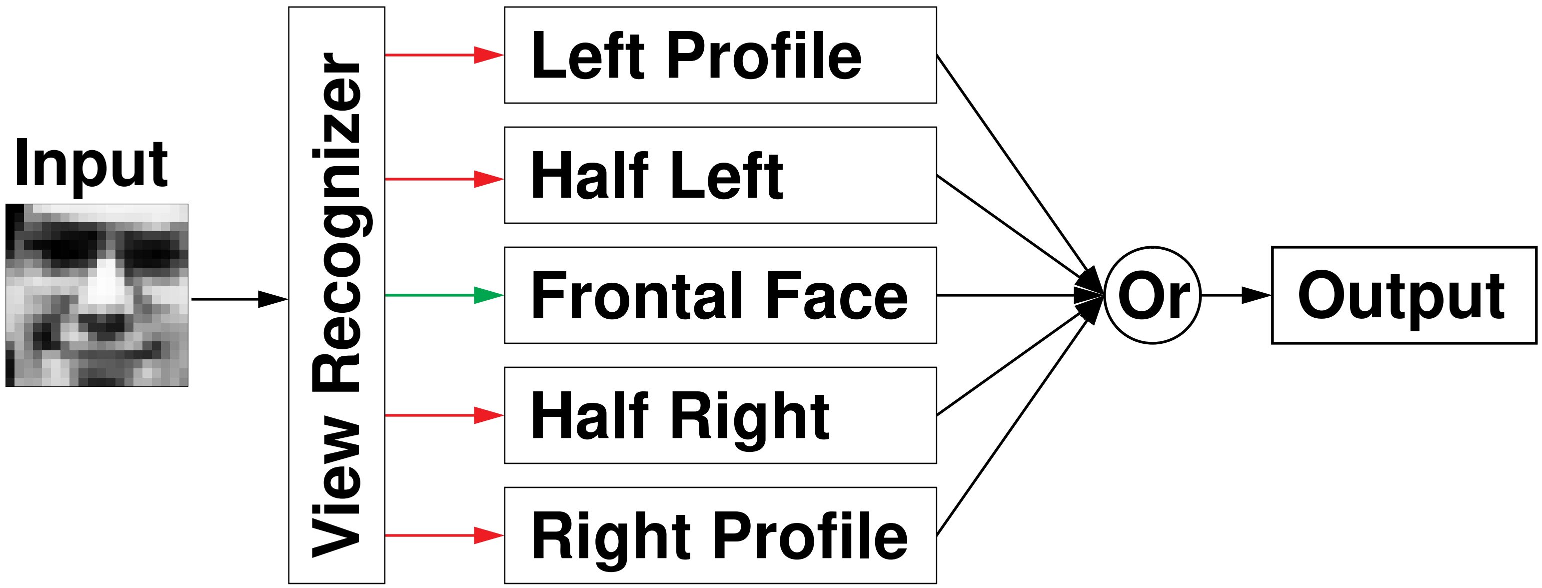


With face location



Partitioning / Merging Views of Faces

Architecture suggested by Baluja [1996]:



View Recognizer accuracy of 99% is easily achieved.

Pose Invariant Face Detection

Applications

Applications for face detector prototype:

- **Associating names with faces in video [Sato, 1996]**
- **Summarizing video [Smith & Kanade,1996]**
- **Image search engine for the WWW**
  http://webseer.cs.uchicago.edu/
  [Frankel, Swain & Athitsos, 1996]
- **Security camera**

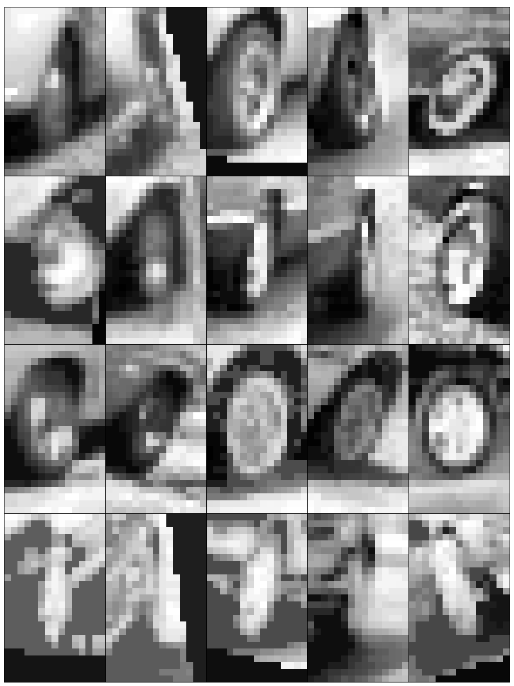
Car Detection

- Goal: Detect cars
- Difficult: Wide variety of shapes
- Solution: Select smaller features
  - Should be present in most views
  - Should be present in most cars
  - Should have stable appearance
- Chosen feature: Tires
- Two ways to partition tire detection:
  - By view
  - By resolution

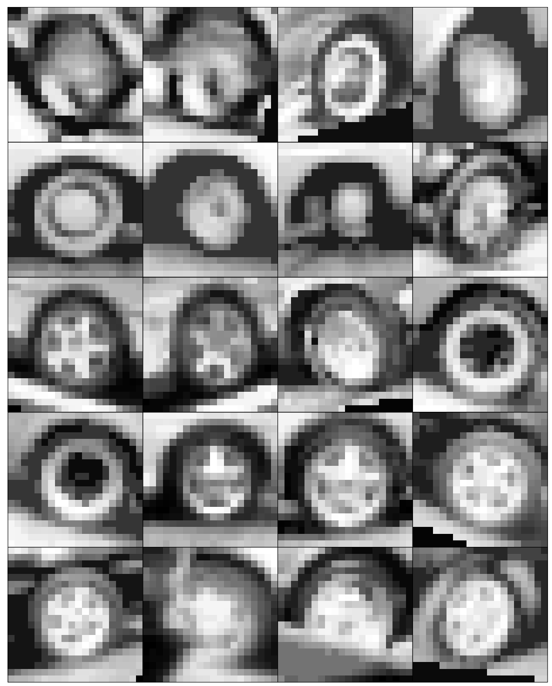
Tire Detection: View Partitioning

Tire images are divided into three classes:

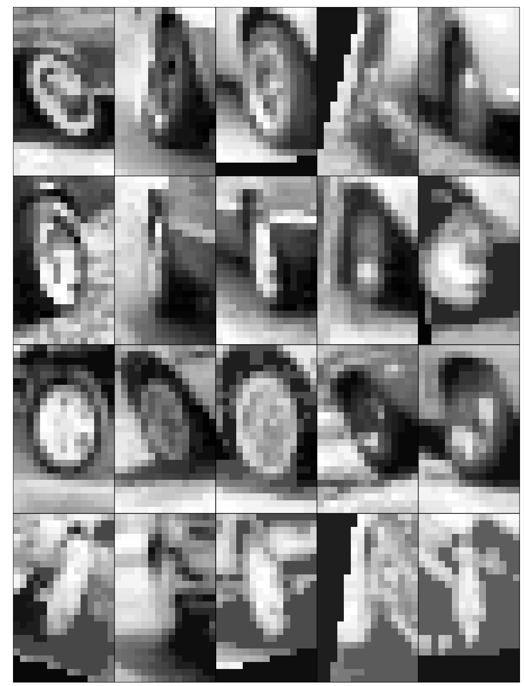
Front View
(15x25 window)



Side View
(20x20 window)

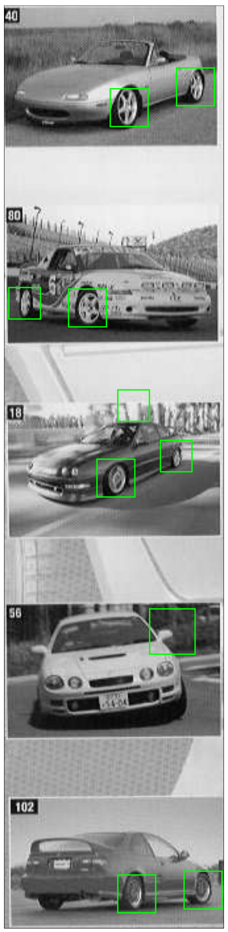


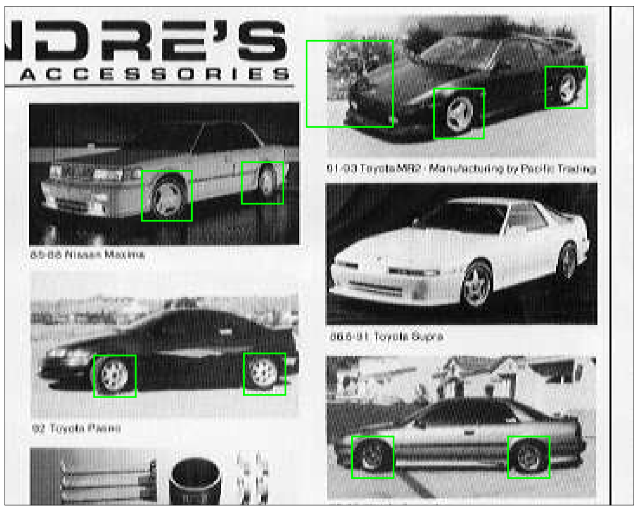
Back View
(15x25 window)



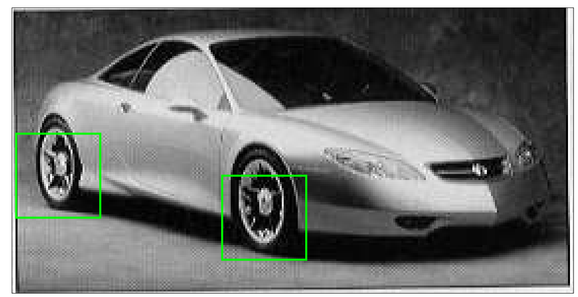
Tire Detection Results (Views)

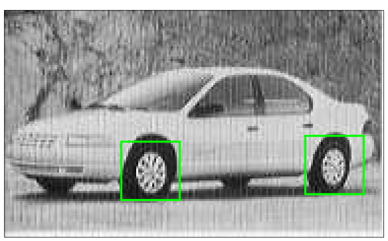
For each view, AND the results of two networks.
To combine views, OR those results:

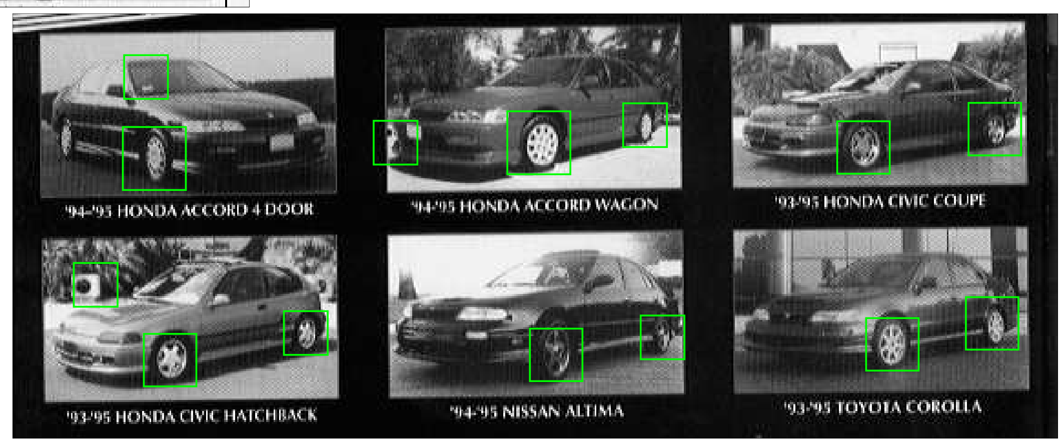




Detects 66.7% of 126 tires
195 false detect/47 images



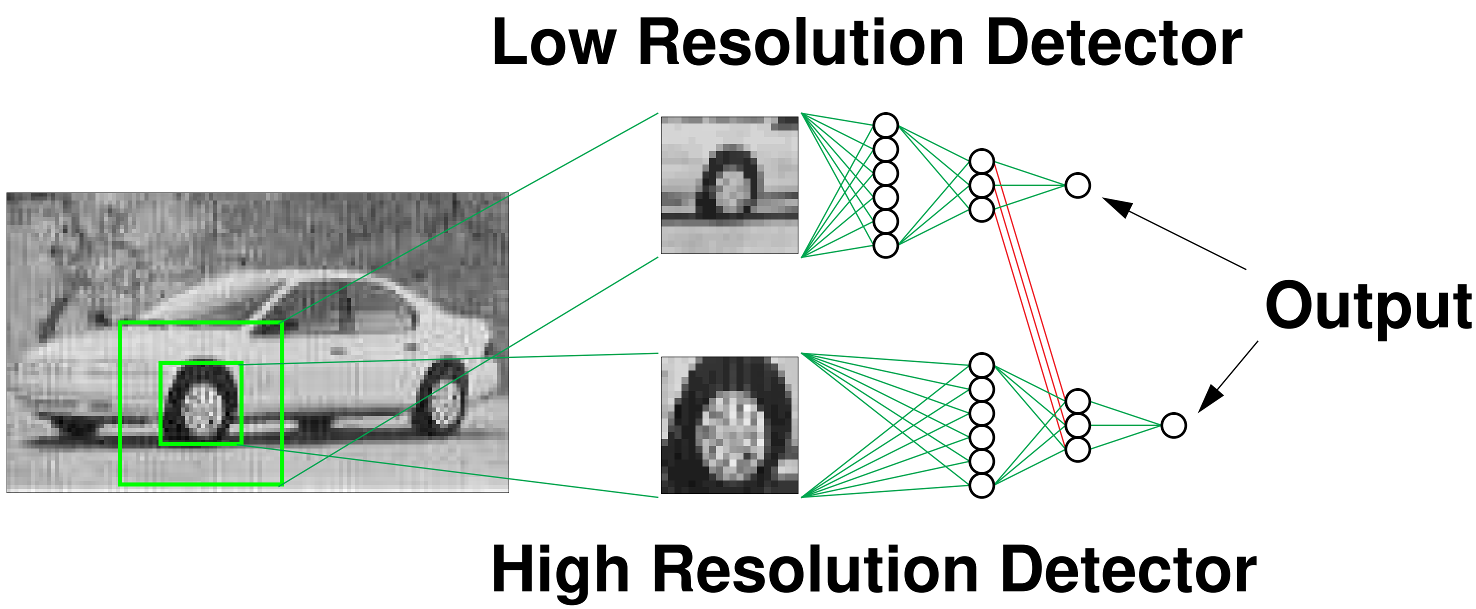




Tire Detection: Hierarchical Method

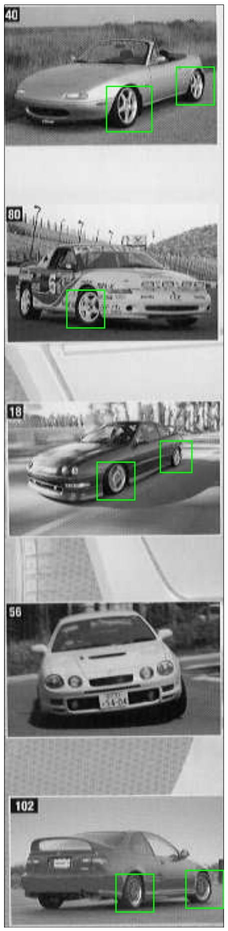
Sajda, Spence & Pearson [Sarnoff]:

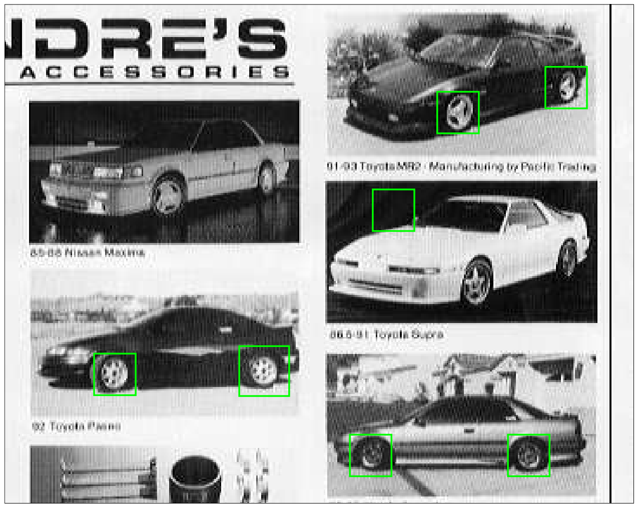
- Train a low resolution network to detect tires.
- Train high resolution network, with extra info:

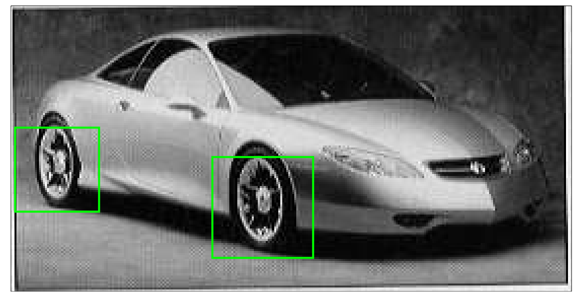


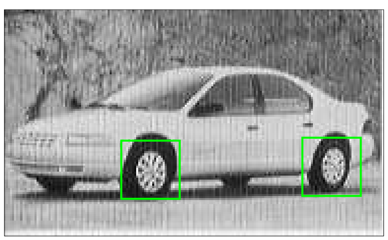
Tire Detection Results (Hierarchy)

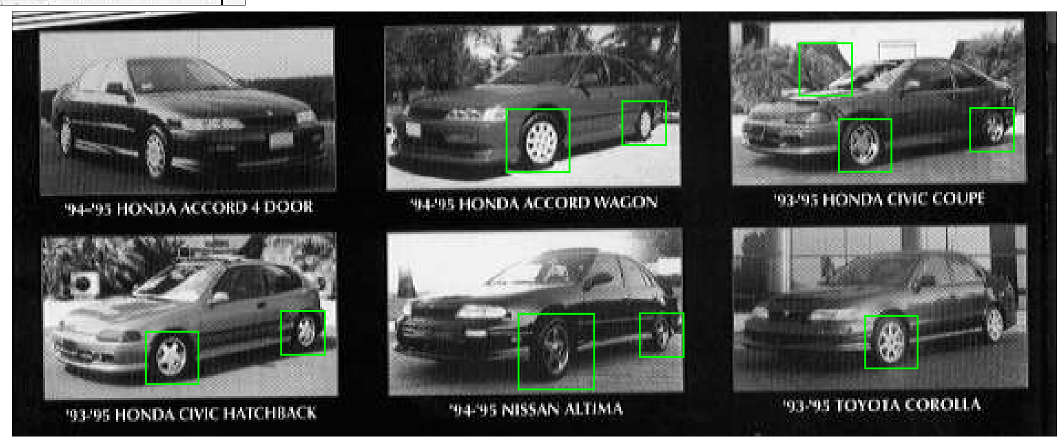
Detects 68.3% of 126 tires
38 false detect / 47 images











Related Work

Partitioning:

Viola [MIT]: algorithms for selecting sub-features

Classifying:

Belhumeur & Kriegman [Yale]: Lighting variation
Sung & Poggio [MIT]: Clustering, distance measures
Pentland *et al* [MIT]: Eigenfaces for face recognition
Cortes & Vapnik [AT&T]: Support vector classifier
Sajda, Spence & Pearson [Sarnoff]: Hierarchical NNs

Merging:

Leung, Burl & Perona [CalTech]: Graph matching
Yow & Cipolla [Cambridge]: Bayesian evidence

Expected Contributions

Methods, heuristics, and algorithms for:

- **Assigning variability to parts of detector**
- **Partitioning views / features**
- **Collecting and aligning training examples**
- **Merging detector outputs**

To demonstrate methods:

- **Pose invariant face detector**
- **Pose invariant car detector (to extent possible)**
- **Possibly other objects to demonstrate detector (fruits, license plates, clock faces, text, *advisors*)**

Approximate Timetable

Activity	Months
Example alignment, lighting variation	1
Data collection, implement parallel NN trainer	2
Industrial internship	4
Experiments on object pose variation	1
Evaluation of classifier types	1
Speed up techniques	1
Feature / view selection and detection	2
Merging and arbitration heuristics	2
Writing and defense	4
Total	18