Neural network applications

To date:

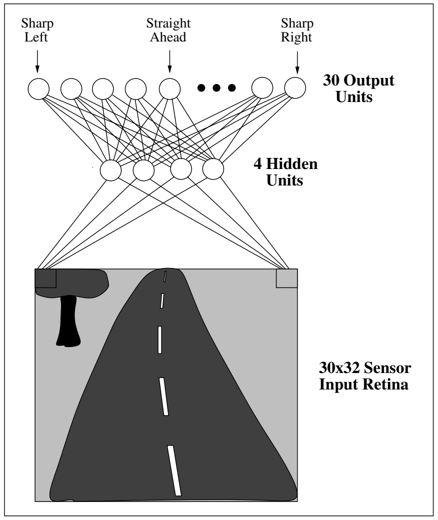
- Neural networks: what are they
- Backpropagation: efficient gradient computation
- Advanced training: (scaled) conjugate gradient
- Adaptive architectures: cascade NN w/NDEKF

Today:

- Neural network applications

ALVINN (Pomerleau, mid 1990s)

Autonomous Land Vehicle in Neural Network



ALVINN overview

Basics:

- Map image of road ahead to steering direction
- Training data: watch (person) and learn

Performance:

- Demonstrated for 100+ continuous miles at 70+ mph (10Hz)
- Neither rain nor sleet nor snow...
- One-lane dirt paths to interstate highways

So is that all there is to it?

ALVINN: input representation

Typical hi-res camera image: $500 \times 500 = 250,000$

- Too many inputs
- Solution: sub-sample image ($32 \times 30 = 960$ — whew!)
- Color/intensity normalization — reduce lighting variability

Questions: Why choose 32×30 **?**

ALVINN: input image example #1

ALVINN: input image example #2

ALVINN: output representation

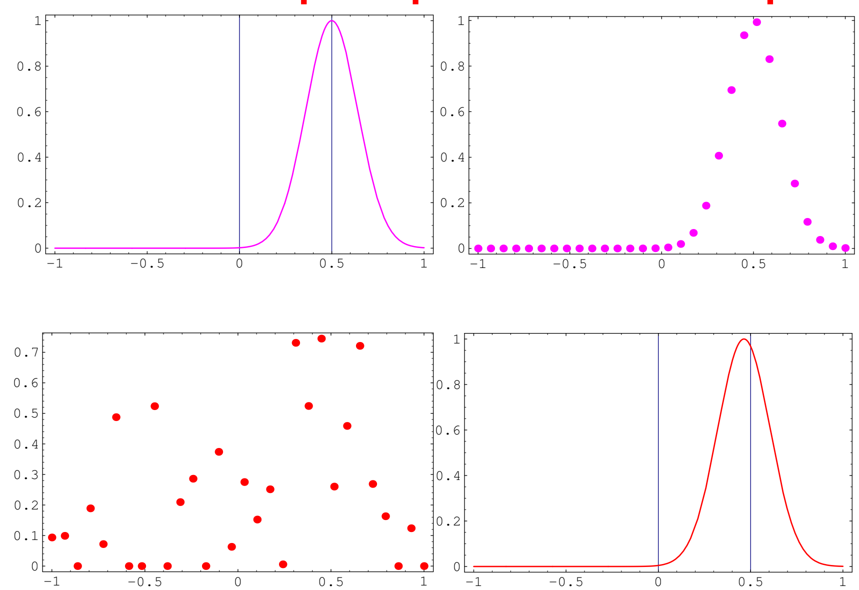
Output representation: two choices

- Single linear output
- Multiple outputs: Gaussian fit

Questions:

- Why choose particular output representation?

Gaussian output representation example



ALVINN: neural network architecture

Tried everything from one to 70 hidden units

Four to five hidden units worked best

Questions:

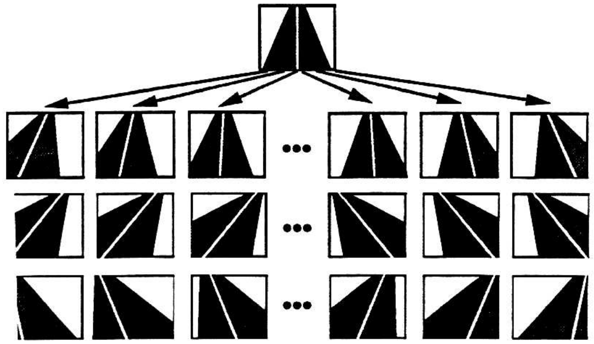
- Why no direct input/output connections?
- Why did larger networks not do better?

ALVINN: training data

Problem: Person drives too well!

- Neural network does not learn error recovery

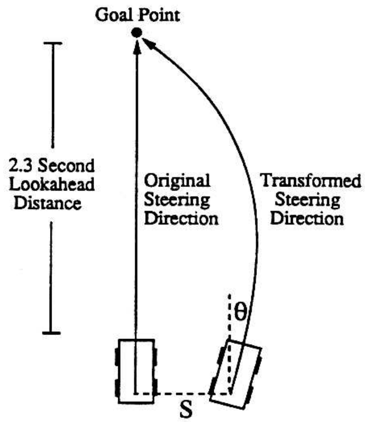
Solution: create synthetic data from real data



ALVINN: synthetic images

Problem: What's the correct steering direction?

- Pure pursuit model of how people driving



ALVINN: spurrious features

Examples of problem data:

- Oil slicks, shadows
- Other cars

Removing spurrious features

Solution #1: Add Gaussian noise to image ***(problems?)***

Solution #2: Model spurrious features ***(problems?)***

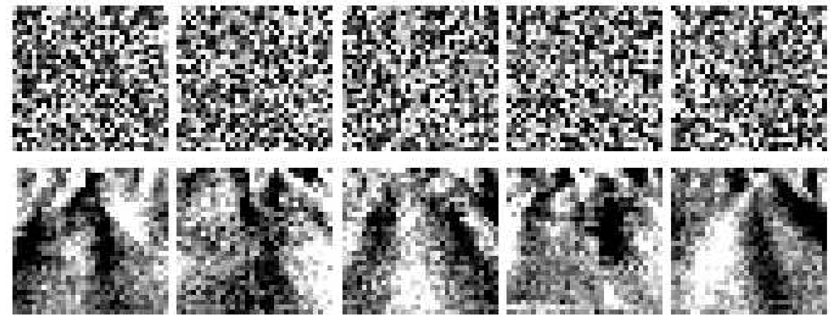
Solution #3: Use neural network's internal model

- "Structured noise"
- Learns to ignore peripheral features

ALVINN: other issues

- **Balance data (left/right/straight samples)** ***(why?)***
- **Training on-line (*vs.* batch)**
- **Hidden unit weights: a closer look**

Hidden Unit 1	Hidden Unit 2	Hidden Unit 3	Hidden Unit 4	Hidden Unit 5



ALVINN: conclusions

- **ALVINN represented a huge step forward in autonomous driving (mid 1990s)**
- **Probably most well-known NN application**
- **Extensively tested at high speeds in real traffic**
- **Next step: learning from ALVINN**

RALPH: learning from ALVINN

Rapid Lateral Position Handler:

- Understanding ALVINN let to RALPH
- Took several years of analysis
- Easy to understand technique

Question:

- Which is better approach?

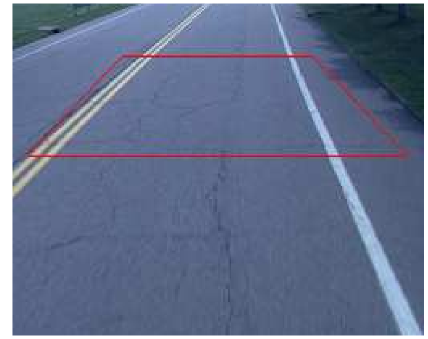
RALPH: basic algorithm

For a given image:

- Trapezoidal subsampling of image
- Hypothesize a road curvature
- *Horizontally* shift pixels to correspond to curvature hypothesis
- *Vertically* add pixel intensities
- Compute measure of curvature hypothesis correctness

Trapezoidal subsampling

Key insight: don't look at whole image



- Function of speed
- Camera orientation w/respect to road (perspective)
- No spurrious feature problem

Trapezoidal subsampling: example #1

Why do trapezoidal subsampling?

Trapezoidal subsampling: example #2

Note how key features line up to indicate curvature...

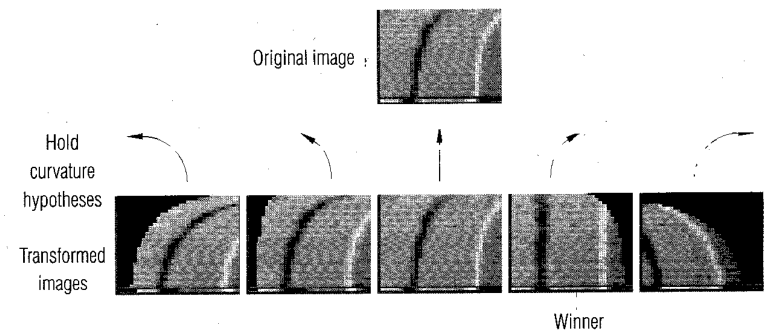
RALPH: basic algorithm

For a given image:

- Trapezoidal subsampling of image
- Curvature hypothesis
- *Horizontally* shift pixels to correspond to curvature hypothesis
- *Vertically* add pixel intensities
- Compute measure of curvature hypothesis correctness

RALPH: curvature hypothesis

- Curvature hypothesis
- *Horizontally* shift pixels to correspond to curvature hypothesis



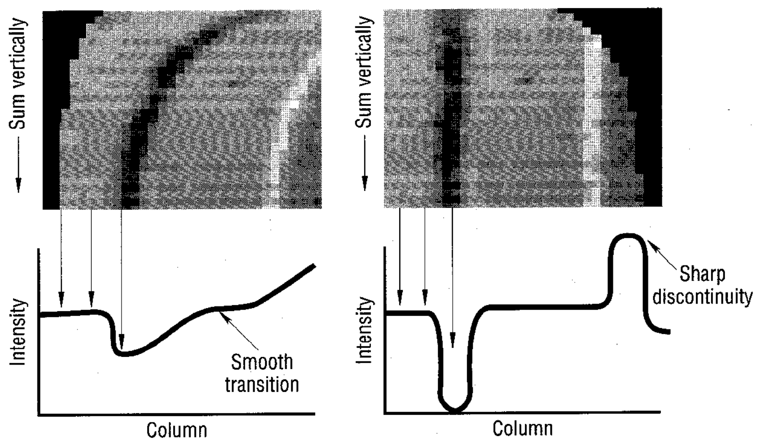
RALPH: basic algorithm

For a given image:

- Trapezoidal subsampling of image
- Hypothesize a road curvature
- *Horizontally* shift pixels to correspond to curvature hypothesis
- *Vertically* add pixel intensities
- Compute measure of curvature hypothesis correctness

RALPH: curvature hypothesis evaluation

- *Vertically* add pixel intensities
- Compute measure of curvature hypothesis correctness



RALPH performance

"No Hands across America"

- Washington, D.C. to San Diego (2,850 miles)
- 98.1% autonomous (2,796 miles)
- 70 mph top speed (officially)
- 110 mph top speed (unofficially)

Lines are useful, but RALPH doesn't need them...

Failure modes...

ALVINN *vs.* RALPH

Which is better?

Neural network applications

Road following

- ALVINN: Road following
- RALPH: learning from neural networks

Face detection

Robot control

Face detection (Kanade, late 1990s)

Basics:

- Map 20×20 image to ± 1 (face/non-face)

Performance:

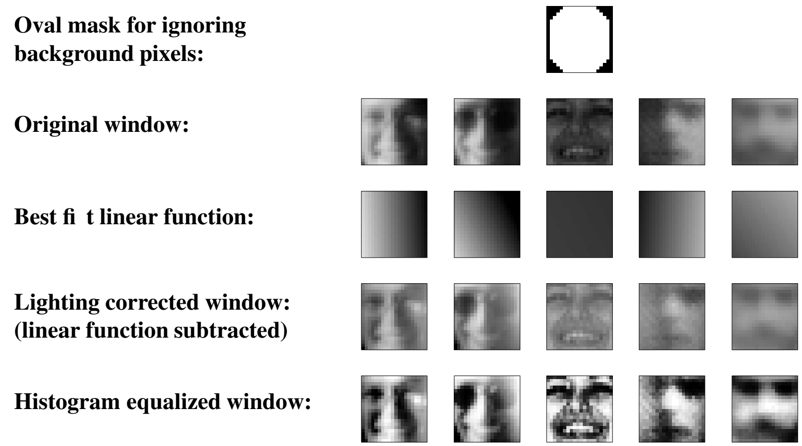
- Face detection results: 85%-90%, few false detects
- 1.5Hz - 3.5Hz on PII/450 (320×240)

Face detection

Outline:

- Which part of image to look at?
- Image pre-processing
- Specialized neural network architecture
- Training data
- Overlap detection
- Committee of experts: multiple neural networks
- Results

Image preprocessing

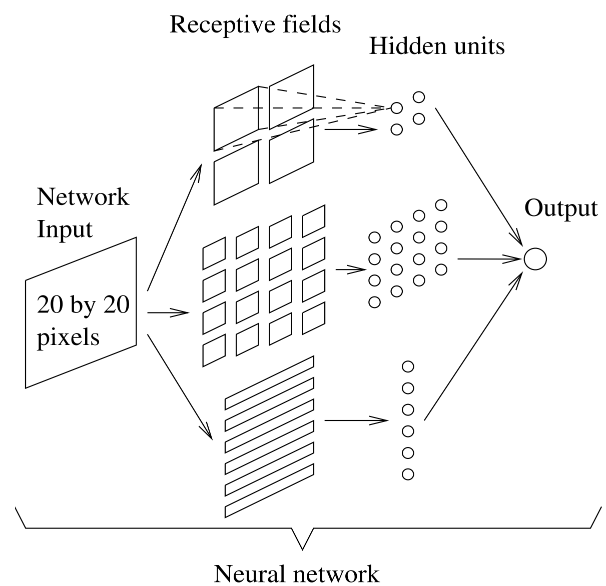


Face detection

Outline:

- Which part of image to look at?
- Image pre-processing
- Specialized neural network architecture
- Training data
- Overlap detection
- Committee of experts: multiple neural networks
- Results

Specialized neural network architecture



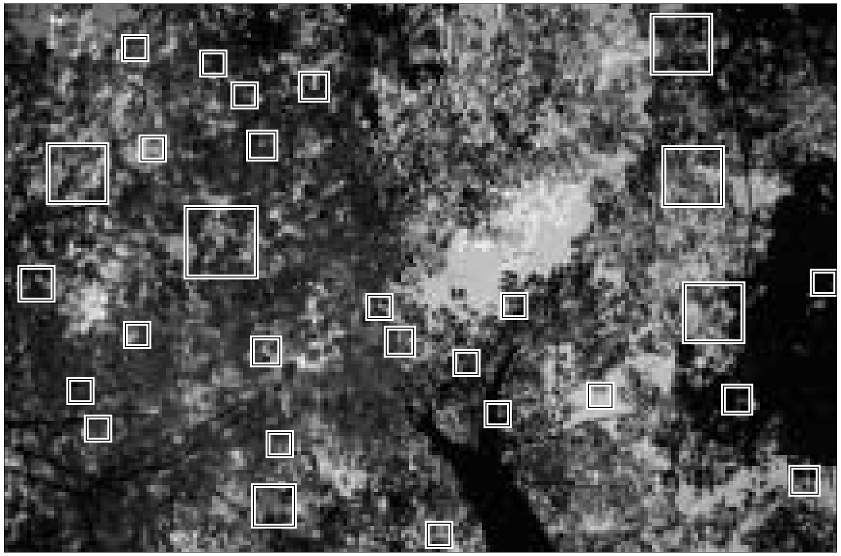
Face detection

Outline:

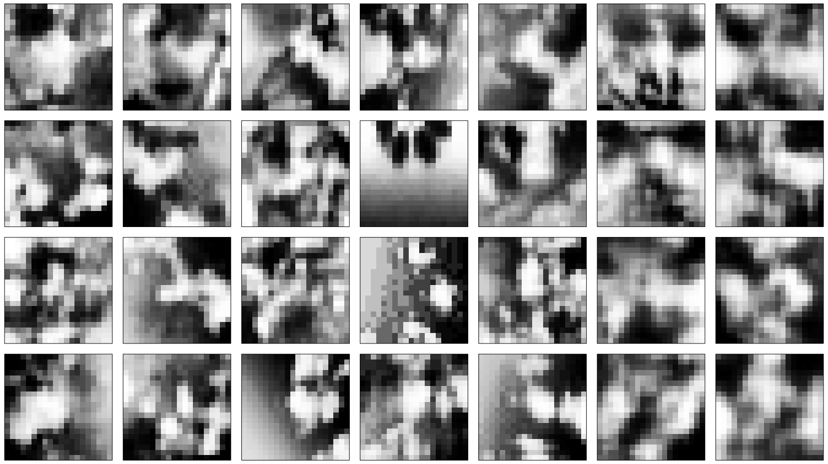
- Which part of image to look at?
- Image pre-processing
- Specialized neural network architecture
- Training data
- Overlap detection
- Committee of experts: multiple neural networks
- Results

NN training data: face examples

Generating non-face examples



NN training data: nonface examples



Basic NN detection results

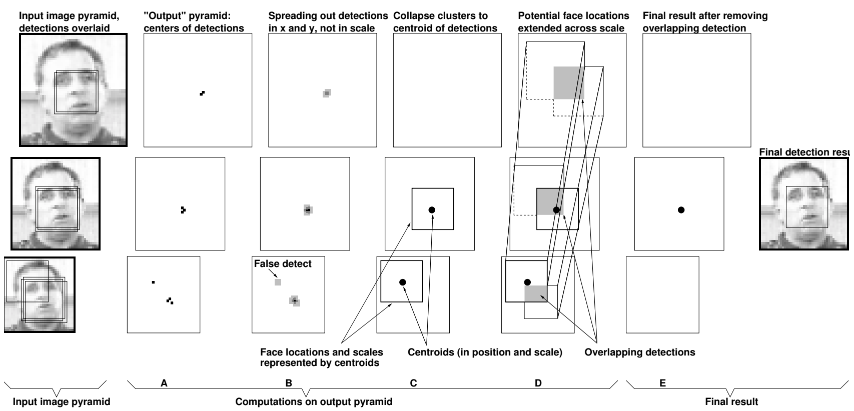
Type	System	Missed faces	Detect rate	False detects
Single network, no heuristics	1) Network 1 (2 copies of hidden units (52 total), 2905 connections)	45	91.1%	945
	2) Network 2 (3 copies of hidden units (78 total), 4357 connections)	38	92.5%	862
	3) Network 3 (2 copies of hidden units (52 total), 2905 connections)	46	90.9%	738
	4) Network 4 (3 copies of hidden units (78 total), 4357 connections)	40	92.1%	819

Face detection

Outline:

- Which part of image to look at?
- Image pre-processing
- Specialized neural network architecture
- Training data
- Overlap detection
- Committee of experts: multiple neural networks
- Results

Overlap detection



NN results w/overlap detection

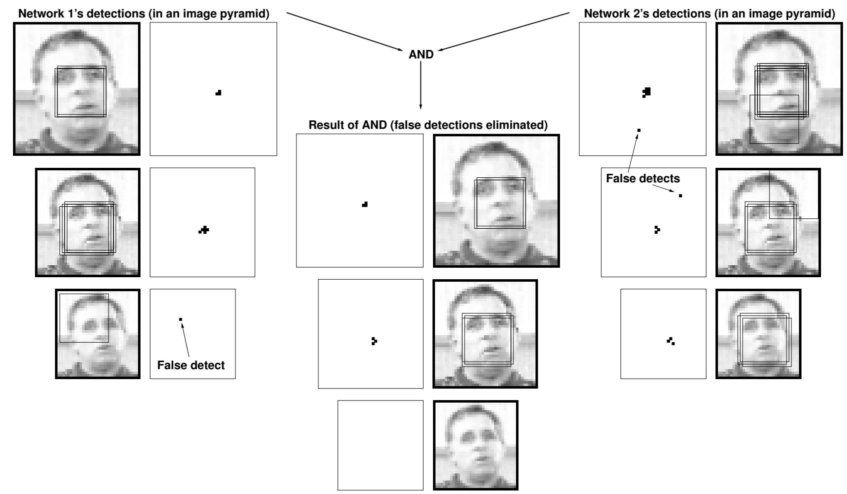
Type	System	Missed faces	Detect rate	False detects
Single network, no heuristics	1) Network 1 (2 copies of hidden units (52 total), 2905 connections)	45	91.1%	945
	2) Network 2 (3 copies of hidden units (78 total), 4357 connections)	38	92.5%	862
	3) Network 3 (2 copies of hidden units (52 total), 2905 connections)	46	90.9%	738
	4) Network 4 (3 copies of hidden units (78 total), 4357 connections)	40	92.1%	819
Single network, with heuristics	5) Network 1 $\rightarrow$ threshold(2,1) $\rightarrow$ overlap elimination	48	90.5%	570
	6) Network 2 $\rightarrow$ threshold(2,1) $\rightarrow$ overlap elimination	42	91.7%	506
	7) Network 3 $\rightarrow$ threshold(2,1) $\rightarrow$ overlap elimination	49	90.3%	440
	8) Network 4 $\rightarrow$ threshold(2,1) $\rightarrow$ overlap elimination	42	91.7%	484

Face detection

Outline:

- Which part of image to look at?
- Image pre-processing
- Specialized neural network architecture
- Training data
- Overlap detection
- Committee of experts: multiple neural networks
- Results

Committee of experts



NN results w/multiple networks

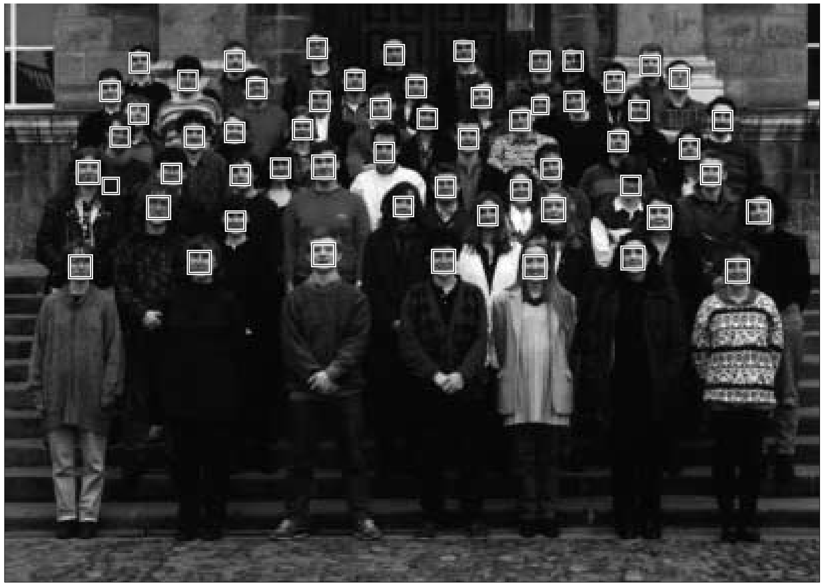
Single network, with heuristics	5) Network 1 → threshold(2,1) → overlap elimination	48	90.5%	570
	6) Network 2 → threshold(2,1) → overlap elimination	42	91.7%	506
	7) Network 3 → threshold(2,1) → overlap elimination	49	90.3%	440
	8) Network 4 → threshold(2,1) → overlap elimination	42	91.7%	484
Arbitrating among two networks	9) Networks 1 and 2 → AND(0)	68	86.6%	79
	10) Networks 1 and 2 → AND(0) → threshold(2,3) → overlap elimination	112	77.9%	2
	11) Networks 1 and 2 → threshold(2,2) → overlap elimination → AND(2)	70	86.2%	23
	12) Networks 1 and 2 → thresh(2,2) → overlap elim → OR(2) → thresh(2,1) → overlap elimination	49	90.3%	185
Arbitrating among three networks	13) Networks 1, 2, 3 → voting(0) → overlap elimination	59	88.4%	99
	14) Networks 1, 2, 3 → network arbitration (5 hidden units) → thresh(2,1) → overlap elimination	79	84.4%	16
	15) Networks 1, 2, 3 → network arbitration (10 hidden units) → thresh(2,1) → overlap elimination	83	83.6%	10
	16) Networks 1, 2, 3 → network arbitration (perceptron) → thresh(2,1) → overlap elimination	84	83.4%	12

Face detection

Outline:

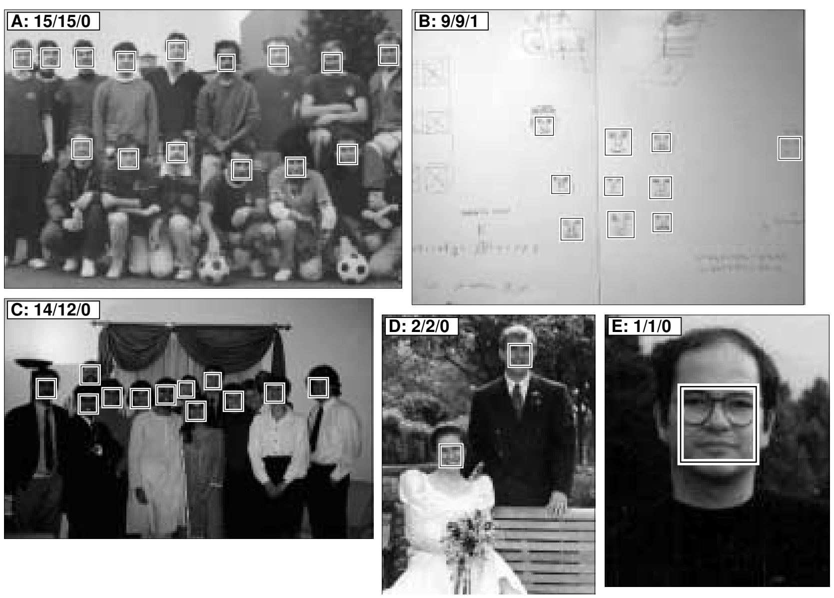
- Which part of image to look at?
- Image pre-processing
- Specialized neural network architecture
- Training data
- Overlap detection
- Committee of experts: multiple neural networks
- Results

Sample detection results



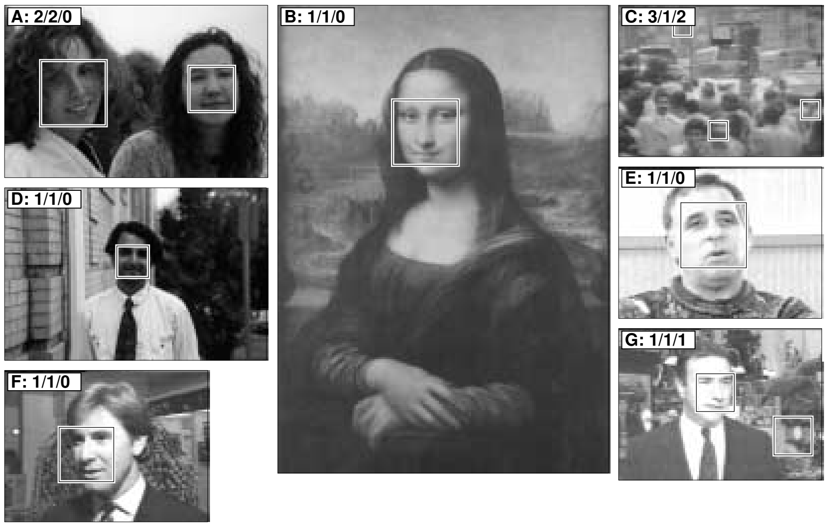
Sample detection results

Sample detection results

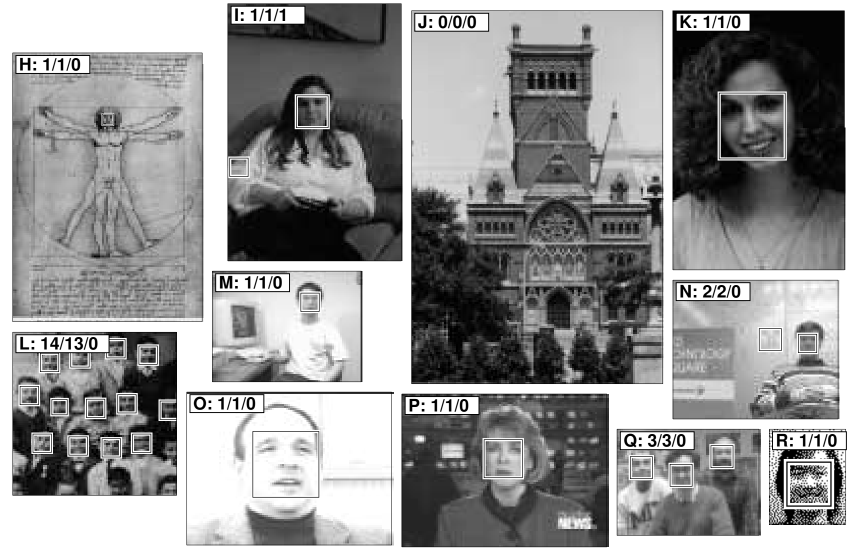


Sample detection results

Sample detection results



Sample detection results



Face detection: concluding thoughts

NN worked as well as anything at the time...

...since then statistical frequency modeling has surpassed accuracy (Schneiderman, 2001)

Comparison (over same test set):

- 95.8% vs. 86.0% detection
- 65 vs. 31 false detections
- slower vs. faster

Commercial system at Superbowl 2001 (Tampa)

Neural network applications

Road following

- ALVINN: Road following
- RALPH: learning from neural networks

Face detection

Robot control

Robot control

Analytic model:

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) \quad \textit{(why important?)}$$

What's missing?

- Friction
- Link flexibility
- Unmodeled dynamics (inertia tensors, masses, etc.)

Bottom line: analytic model will not be 100%

Robot control

Analytic model:

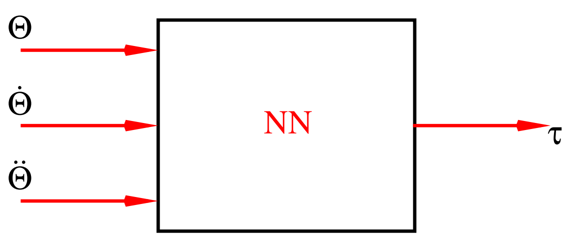
$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta) \quad \textit{(why important?)}$$

What's missing?

- Friction
- Link flexibility
- Unmodeled dynamics (inertia tensors, masses, etc.)

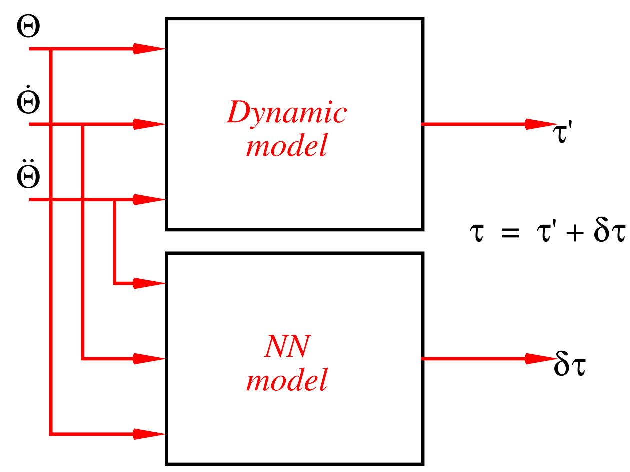
Bottom line: analytic model will not be 100%

Use NN to model robot dynamics



Is this a good idea?

Better idea: complement analytic model



Why is this better?

Neural network applications

Road following

- ALVINN: Road following
- RALPH: learning from neural networks

Face detection

Robot control

Other applications?

Why didn't we use it for horizon tracking?