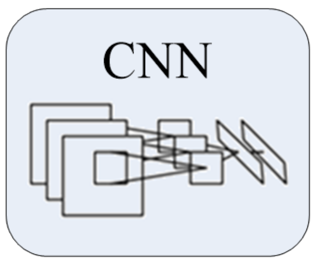
# Direct Perception for Autonomous Driving

Chenyi Chen

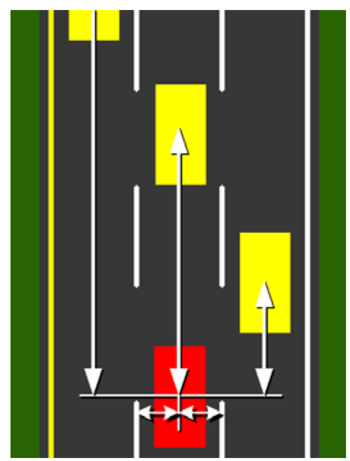
# DeepDriving

- http://deepdriving.cs.princeton.edu/

# Direct Perception

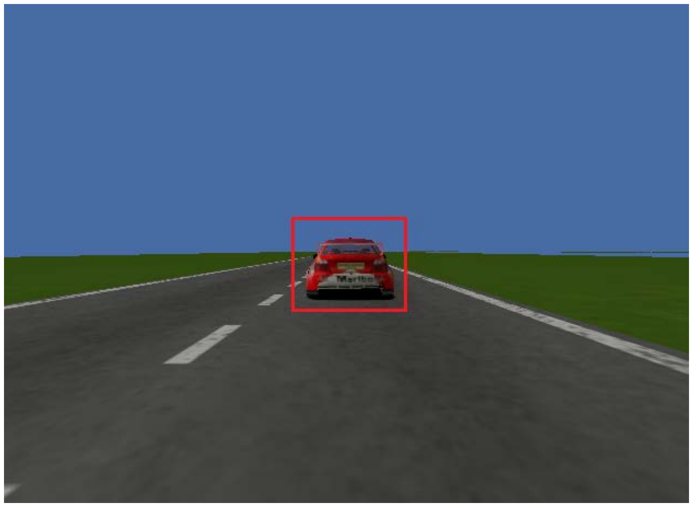



estimate



control

# Direct Perception (in Driving…)



- Ordinary car detection: Find a car! It's localized in the image by a red bounding box.
- Direct perception: The car is in the right lane; 16 meters ahead
- Which one is more helpful for driving task?

# Machine Learning

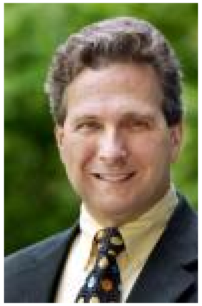
# What's machine learning?

- Example problem: face recognition


Prof. K


Prof. F


Prof. P


Prof. V


Chenyi

- Training data: a collection of images and labels (names)

Who is this guy?

- Evaluation criterion: correct labeling of new images

# What’s machine learning?

- Example problem: scene classification

road

road

sea

mountain

city

- a few labeled training images

What’s the label of this image?

- goal to label yet unseen image

# Supervised Learning

- System input: X
- System model: $f_\theta()$, with parameter θ
- System output: $\hat{Y} = f_\theta(X)$,
- Ground truth label: Y
- Loss function: L(θ)
- Learning rate: γ

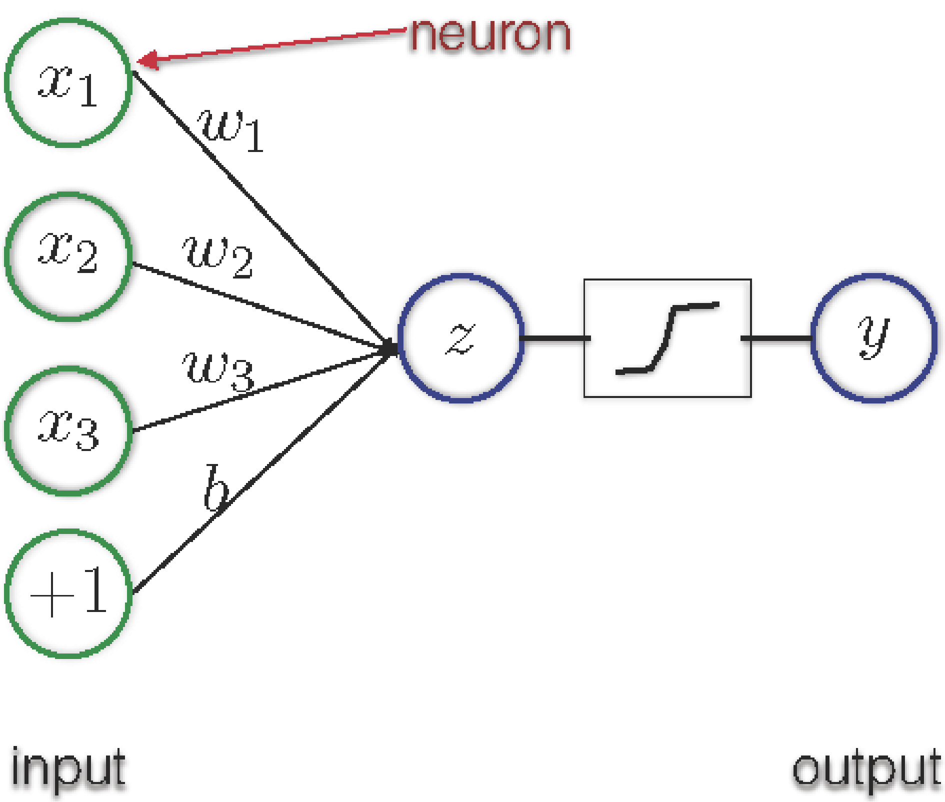
# Why machine learning?

- The world is very complicated
- We don’t know the exact model/mechanism between input and output
- Find an approximate (usually simplified) model between input and output through learning

# Deep Learning:
# A sub-field of machine learning

# Neural Networks

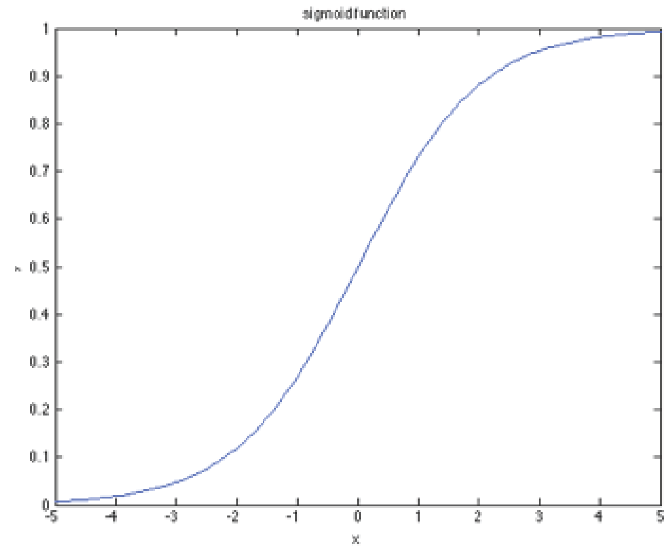
basic building blocks




$$z = \sum_i x_i w_i + b, \quad y = f(z)$$

where f is a activation function:

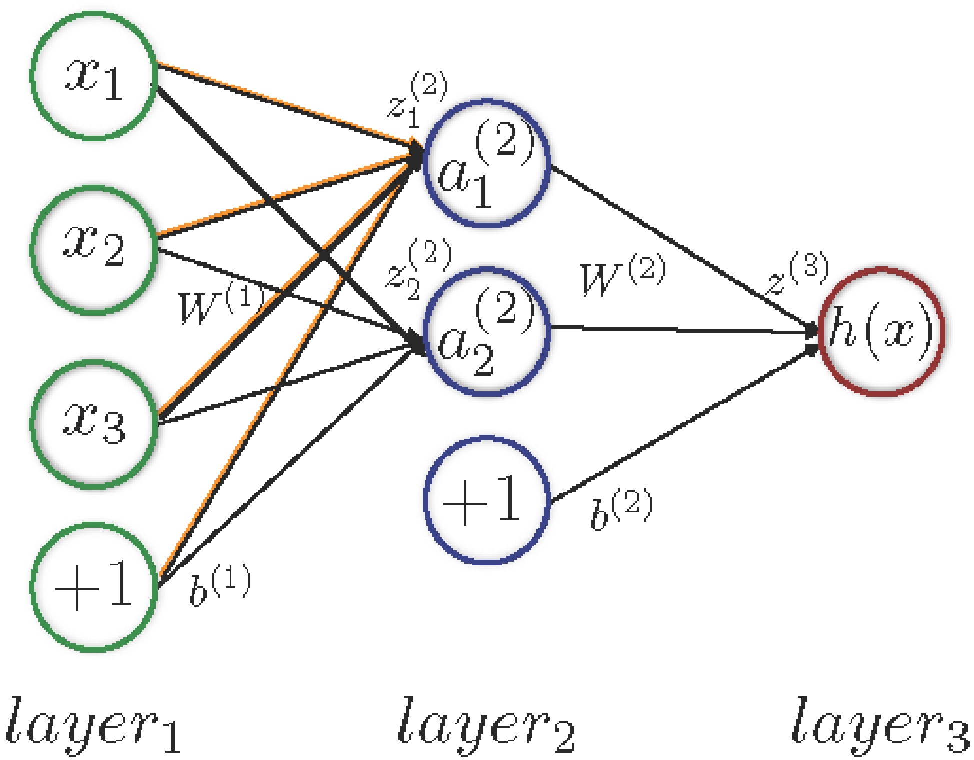
$$f(z) = \sigma(z) = \frac{1}{1 + exp(-z)}$$




- sigmoid is bounded between 0 and 1
- monotonically increasing
- differentiation: $\sigma'(z) = \sigma(z) * (1 - \sigma(z))$

# Neural Networks

## representation




## feed-forward

for each neuron in the next layer:

$$z_i^{(2)} = \sum_{j=1}^{n} w_{ij}^{(1)} x_j + b_i^{(1)}, \quad a_i^{(2)} = f(z_i^{(2)})$$

$$z_1^{(2)} = w_{11}^{(1)} x_1 + w_{21}^{(1)} x_2 + w_{31}^{(1)} x_3 + b_1^{(1)}, \quad a_1^{(2)} = f(z_1^{(2)})$$

compactly:

$$z^{(2)} = W^{(1)} x + b^{(1)}, \quad a^{(2)} = f(z^{(2)})$$

$$z^{(3)} = W^{(2)} x + b^{(2)}, \quad a^{(3)} = f(z^{(3)})$$

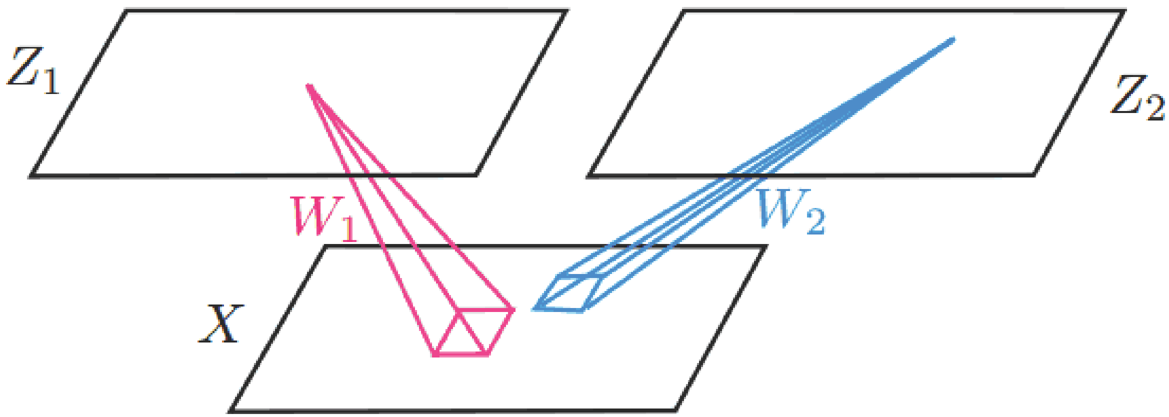
globally models a function:

$$\hat{y} = h_{W,b}(x)$$

where W and b are model parameters

# Convolutional Neural Networks

detection
layers




convolution

$$Z_i = \sigma(W_i * X)$$

# Convolutional Neural Networks

pooling :

- reduce the size of representations
- allow small translation invariance.

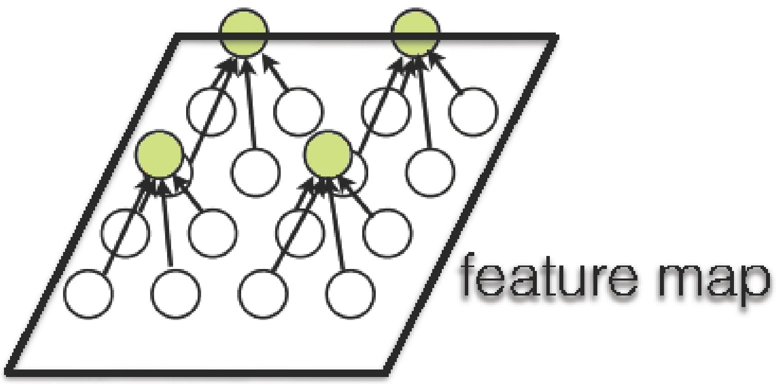
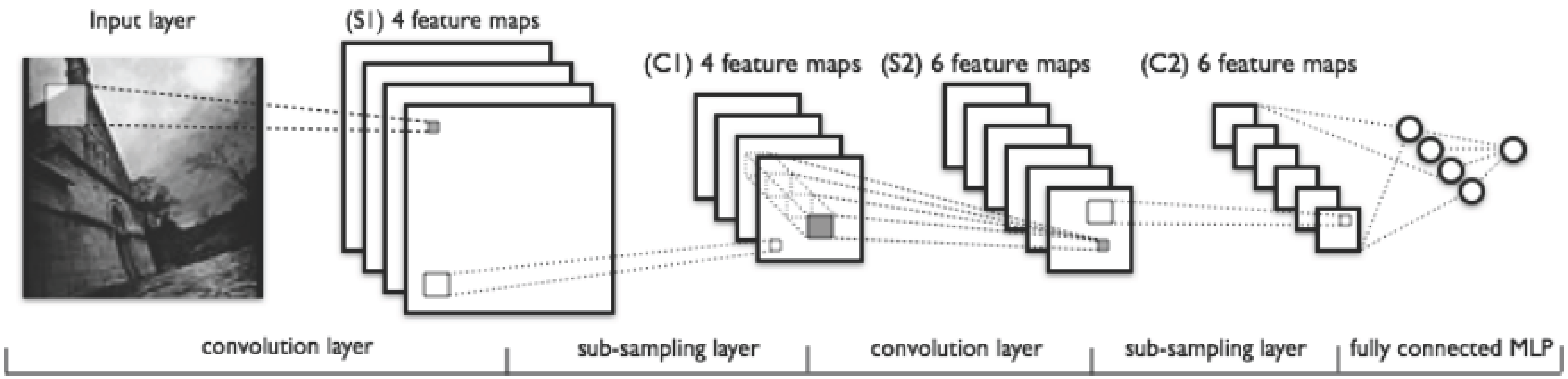



figure from roger grosse tutorial

common pooling techniques:
max pooling, average pooling

# Convolutional Neural Networks

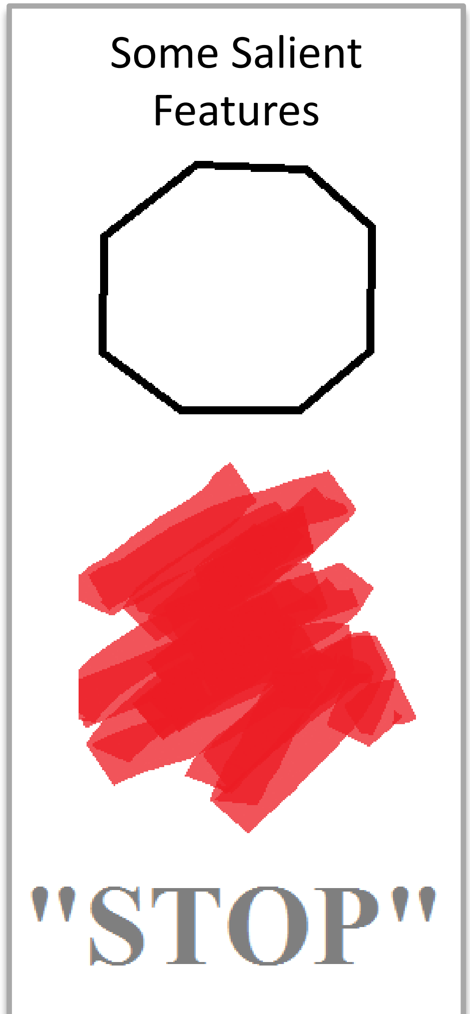
convolution network is just a combination of convolution layers, pooling layers and fully connected layers




(figure from LeNet tutorial, http://deeplearning.net/tutorial/lenet.html)

# Why deep learning?

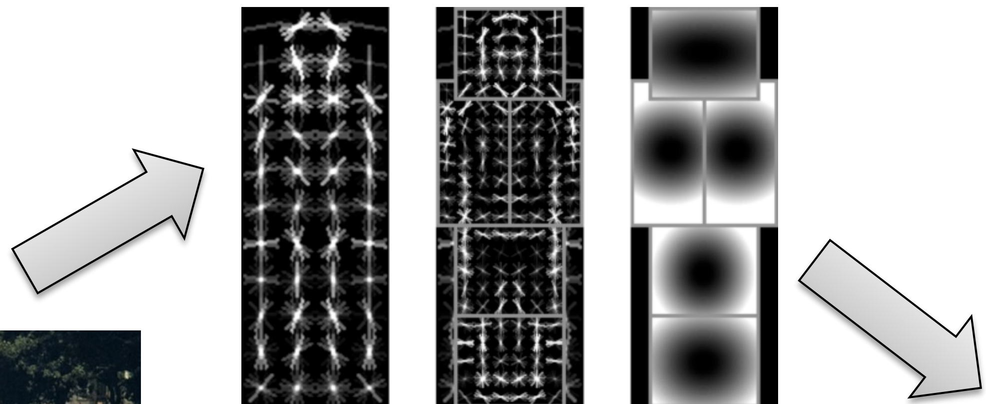
How do we detect a stop sign? It's all about feature!

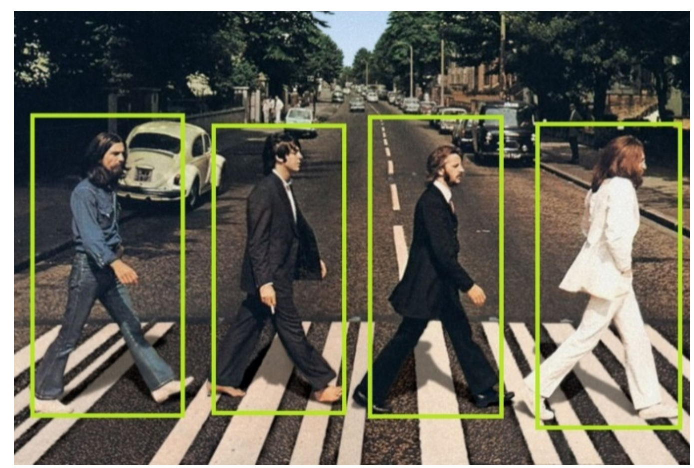

**It's a STOP sign!!**

# Why deep learning?

How does computer vision algorithm work? It’s all about feature!

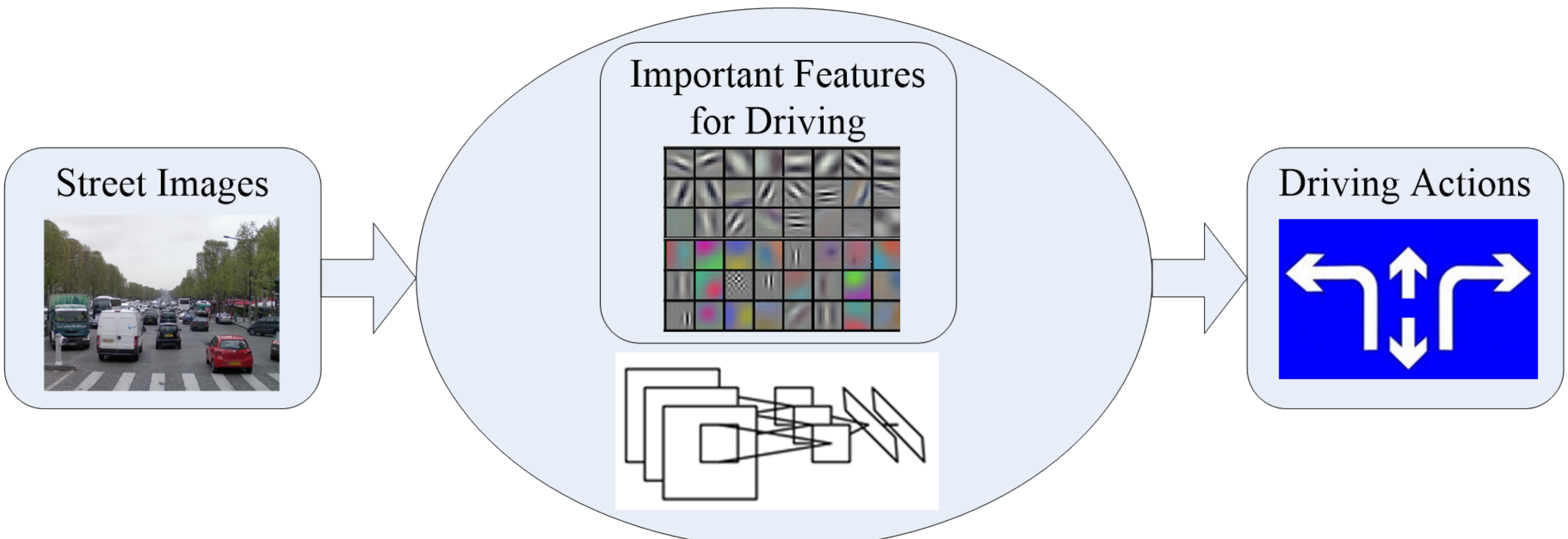


Pedestrian found!!!



# Why deep learning?

- We believe driving is also related with certain features
- Those features determine what action to take next
- Salient features can be automatically detected and processed by deep learning algorithm




Deep Convolutional Neural Network

# Deep Convolutional Neural Network (CNN)

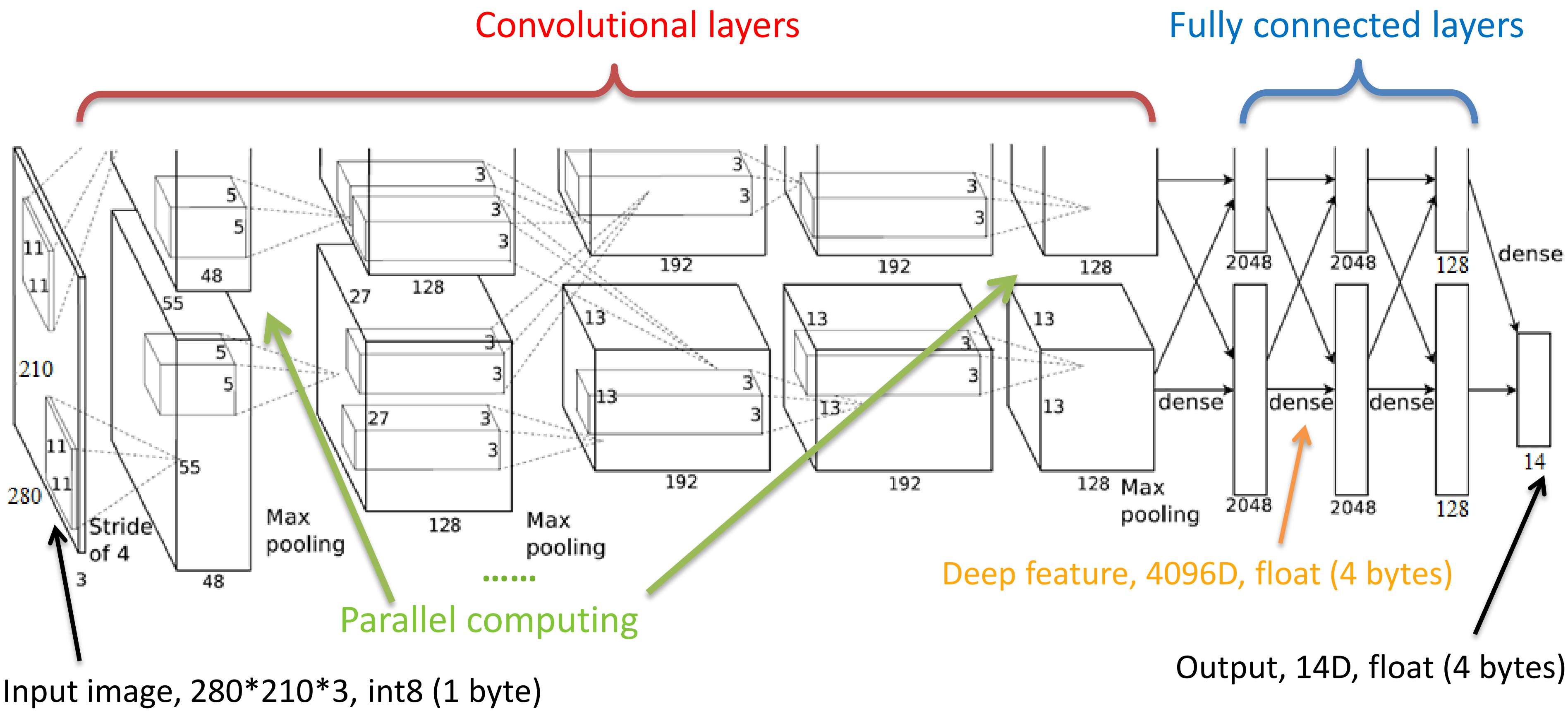
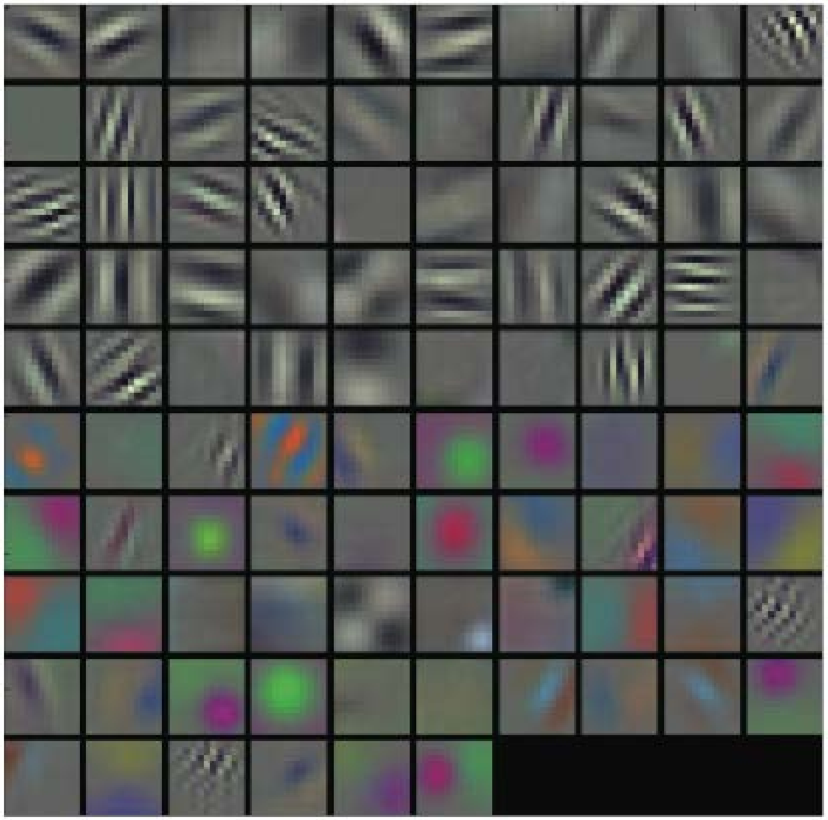



Figure courtesy and modified of Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton

# How does CNN work?

- Input image convolves with a large set of filters, and this process is repeated in many layers.
- Compared to traditional computer vision feature extraction method, e.g. SIFT, HOG, more powerful feature representation of the input image is learnt automatically through convolutions.



A visualized example of the set of filters

# Why deep learning?

- ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012), the most famous challenge in computer vision field
  - 1000 categories
  - 1.2 million training images
  - 50,000 validation images
  - 150,000 testing images
  - Top-5 error rate* of deep learning: 15.3%
  - Top-5 error rate of second best (which is non-deep learning): 26.2%

***Top-5 error rate**: the fraction of test images for which the correct label is not among the five labels considered most probable by the model

# Result Page of ILSVRC2012 Website

**Deep Learning** →

**Deep Learning** →

Only one team!

**Task 1**

| Team name | Filename | Error (5 guesses) | Description |
|---|---|---|---|
| SuperVision | test-preds-141-146.2009-131-137-145-146.2011-145f. | 0.15315 | Using extra training data from ImageNet Fall 2011 release |
| SuperVision | test-preds-131-137-145-135-145f.txt | 0.16422 | Using only supplied training data |
| ISI | pred_FVs_wLACs_weighted.txt | 0.26172 | Weighted sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively. |
| ISI | pred_FVs_weighted.txt | 0.26602 | Weighted sum of scores from classifiers using each FV. |
| ISI | pred_FVs_summed.txt | 0.26646 | Naive sum of scores from classifiers using each FV. |
| ISI | pred_FVs_wLACs_summed.txt | 0.26952 | Naive sum of scores from each classifier with SIFT+FV, LBP+FV, GIST+FV, and CSIFT+FV, respectively. |
| OXFORD_VGG | test_adhocmix_classification.txt | 0.26979 | Mixed selection from High-Level SVM scores and Baseline Scores, decision is performed by looking at the validation performance |
| XRCE/INRIA | res_1M_svm.txt | 0.27058 | |

# Result Page of ILSVRC2014 Website

### Task 2a: Classification+localization with provided training data

**Classification+localization with provided training data: Ordered by classification error**

Deep Learning

| Team name | Entry description | **Classification error** | Localization error |
|---|---|---|---|
| GoogLeNet | No localization. Top5 val score is 6.66% error. | 0.06656 | 0.606257 |
| VGG | a combination of multiple ConvNets, including a net trained on images of different size (fusion weights learnt on the validation set); detected boxes were not updated | 0.07325 | 0.256167 |
| VGG | a combination of multiple ConvNets, including a net trained on images of different size (fusion done by averaging); detected boxes were not updated | 0.07337 | 0.255431 |
| VGG | a combination of multiple ConvNets (by averaging) | 0.07405 | 0.253231 |
| VGG | a combination of multiple ConvNets (fusion weights learnt on the validation set) | 0.07407 | 0.253501 |
| MSRA Visual Computing | Multiple SPP-nets further tuned on validation set (B) | 0.0806 | 0.354924 |
| MSRA Visual Computing | Multiple SPP-nets further tuned on validation set (A) | 0.08062 | 0.354769 |
| Andrew Howard | Combination of Convolutional Nets with Validation set adaptation + KNN | 0.08111 | 0.610365 |
| MSRA Visual Computing | Multiple SPP-nets (B) | 0.082 | 0.355568 |
| Andrew Howard | Combination of Convolutional Nets with Validation set adaptation | 0.08226 | 0.611019 |
| MSRA Visual Computing | Multiple SPP-nets (A) | 0.08307 | 0.3562 |
| VGG | a single ConvNet (13 convolutional and 3 fully-connected layers) | 0.08434 | 0.267184 |
| Andrew Howard | Combination of Convolutional Nets + KNN | 0.0853 | 0.612305 |
| Andrew Howard | Baseline Combination of Convolutional Nets | 0.08919 | 0.614307 |
| MSRA Visual Computing | A single SPP-net | 0.09079 | 0.36118 |
| DeeperVision | Simple average ensemble | 0.09508 | 1.0 |
| DeeperVision | Simple average ensemble and box | 0.09508 | 0.842953 |
| DeeperVision | Weighted ensemble | 0.09556 | 1.0 |

# Result Page of ILSVRC2014 Website

**Task 1b: Object detection with additional training data**

**Ordered by mean average precision**

Deep Learning

| Team name | Entry description | Description of outside data used | **mean AP** | Number of object categories won |
|---|---|---|---|---|
| GoogLeNet | Ensemble of detection models. Validation is 44.5% mAP | Pretraining on ILSVRC12 classification data. | 0.439329 | 142 |
| *CUHK DeepID-Net* | *Combine multiple models described in the abstract without contextual modeling. The training data includes the validation dataset 2.* | *ImageNet classification and localization data* | *0.406998* | --- |
| CUHK DeepID-Net | Combine multiple models described in the abstract without contextual modeling | ImageNet classification and localization data | 0.406659 | 29 |
| Deep Insight | Combination of three detection models | Three CNNs from classification task are used for initialization. | 0.404517 | 27 |
| *CUHK DeepID-Net2* | *Combine multiple models described in the abstract without contextual modeling. The training data includes the validation dataset 2.* | *ImageNet classification and localization data* | *0.40352* | --- |
| *CUHK DeepID-Net2* | *Combine multiple models described in the abstract without contextual modeling* | *ImageNet classification and localization data* | *0.403417* | --- |
| *Deep Insight* | *A single detection model.* | *A CNN from classification task is used for initialization.* | *0.401568* | --- |
| *Deep Insight* | *Another single detection model.* | *A CNN from classification task is used for initialization.* | *0.396982* | --- |
| *GoogLeNet* | *Single detection model. Validation is 38.75% mAP* | *Pretraining on ILSVRC12 classification data.* | *0.380277* | --- |
| *CUHK DeepID-Net2* | *Multi-stage deep CNN without contextual modeling* | *ImageNet classification and localization data* | *0.377471* | --- |
| UvA-Euvision | Deep learning with outside data | ImageNet 1000 | 0.354213 | 1 |
| *CUHK* [illegible] | *A single deep CNN with deformation layers* | *ImageNet classification and* | [illegible] | |

# Why deep learning?

- Deep learning first impacted the computer vision field in ILSVRC2012
- Now it’s almost dominating the computer vision field
- It only takes less than three years!

# How does our system work?

# Basic Ideas

- Extract key parameters from driving scenes images with deep learning CNN

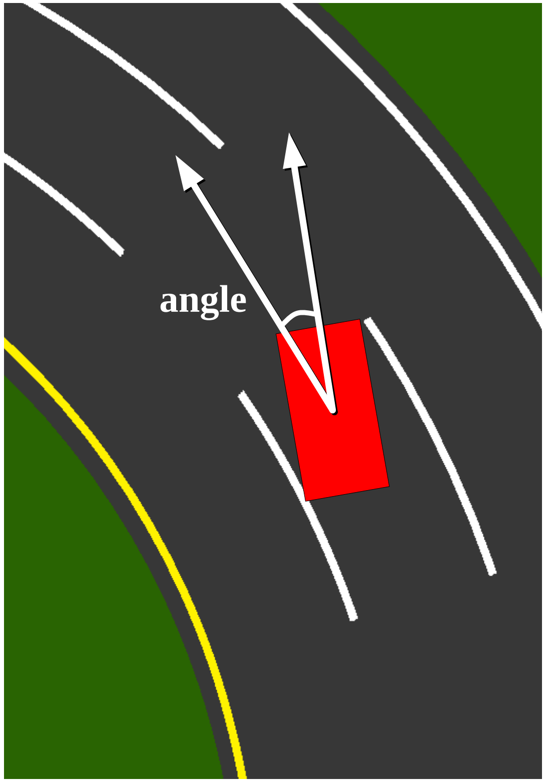
- Compute driving control (optimal control) based on those parameters

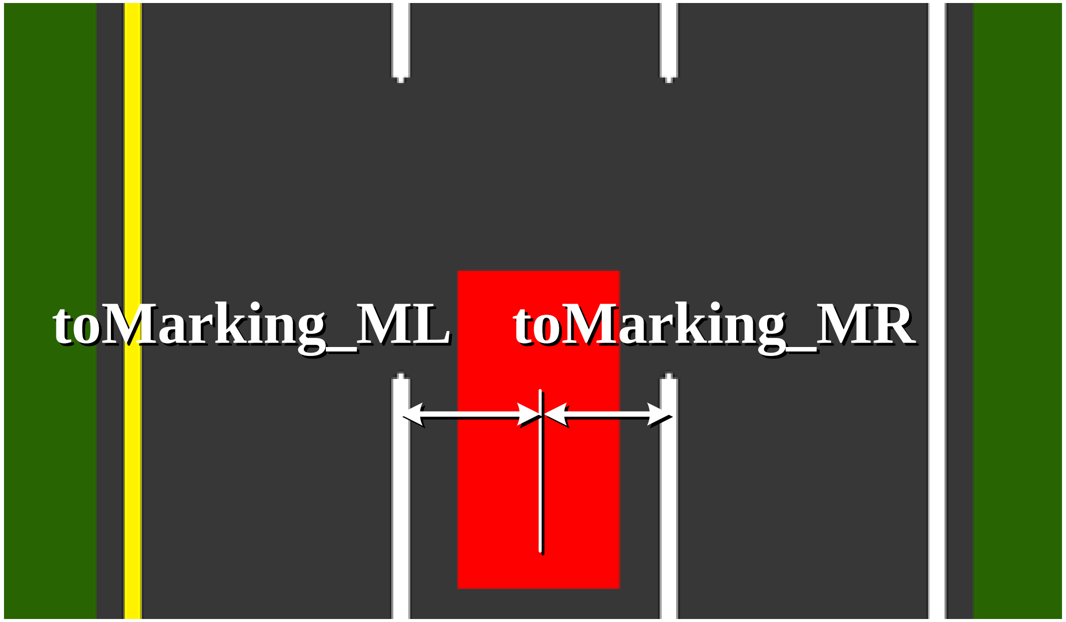
# In our specific case…

- Let the deep learning algorithm tell us:
- angle: the angle between the car’s heading and the tangent of the track;
- toMarking: the distance between the center of the car and each lane marking;
- dist: the distance between the car and the preceding car in each lane;
- … 14 parameters in total

# In our specific case…

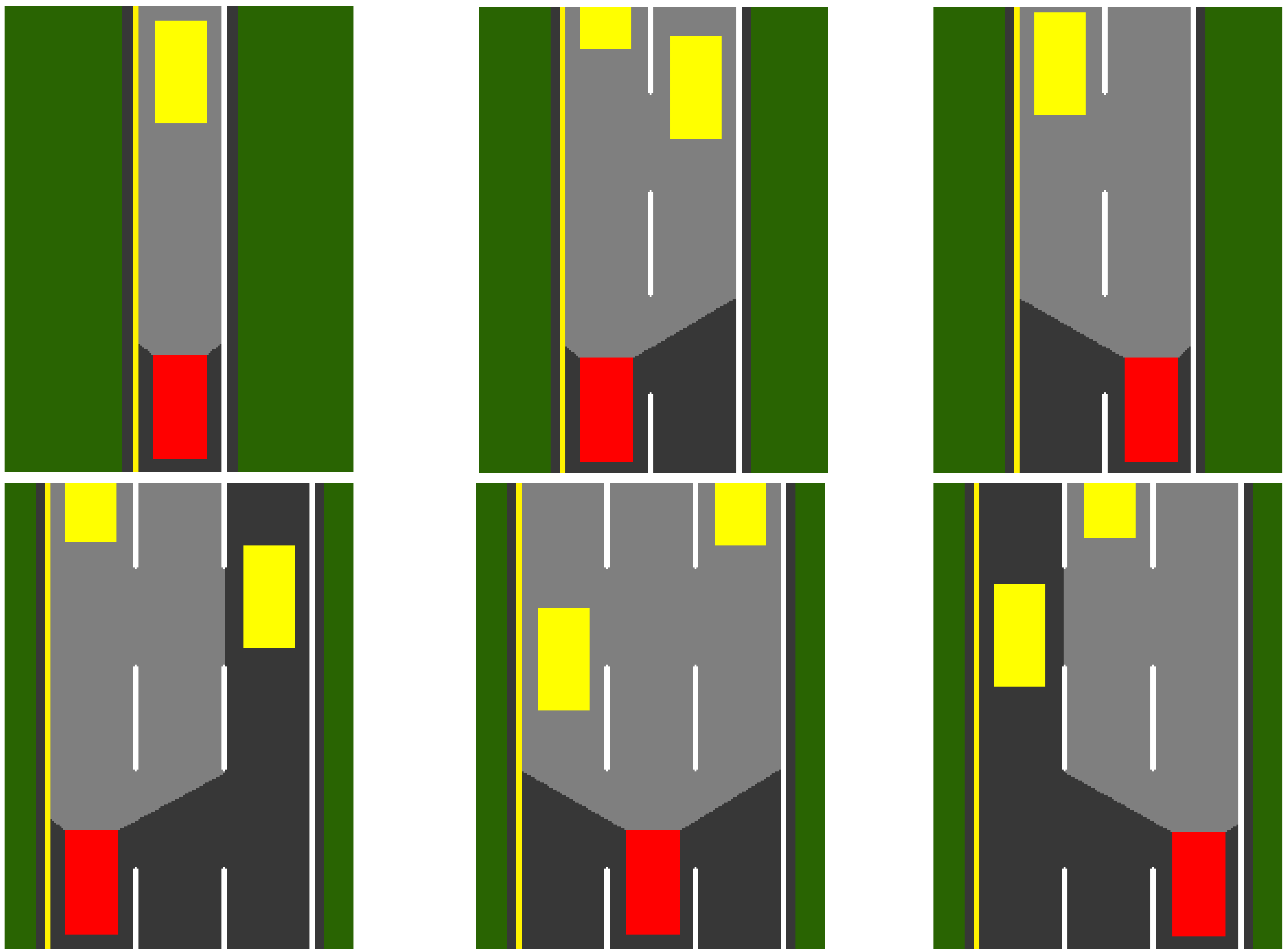
- Illustration of key parameters for car pose and localization



# The cases we are dealing with

- Monitoring current lane and left & right neighboring lanes

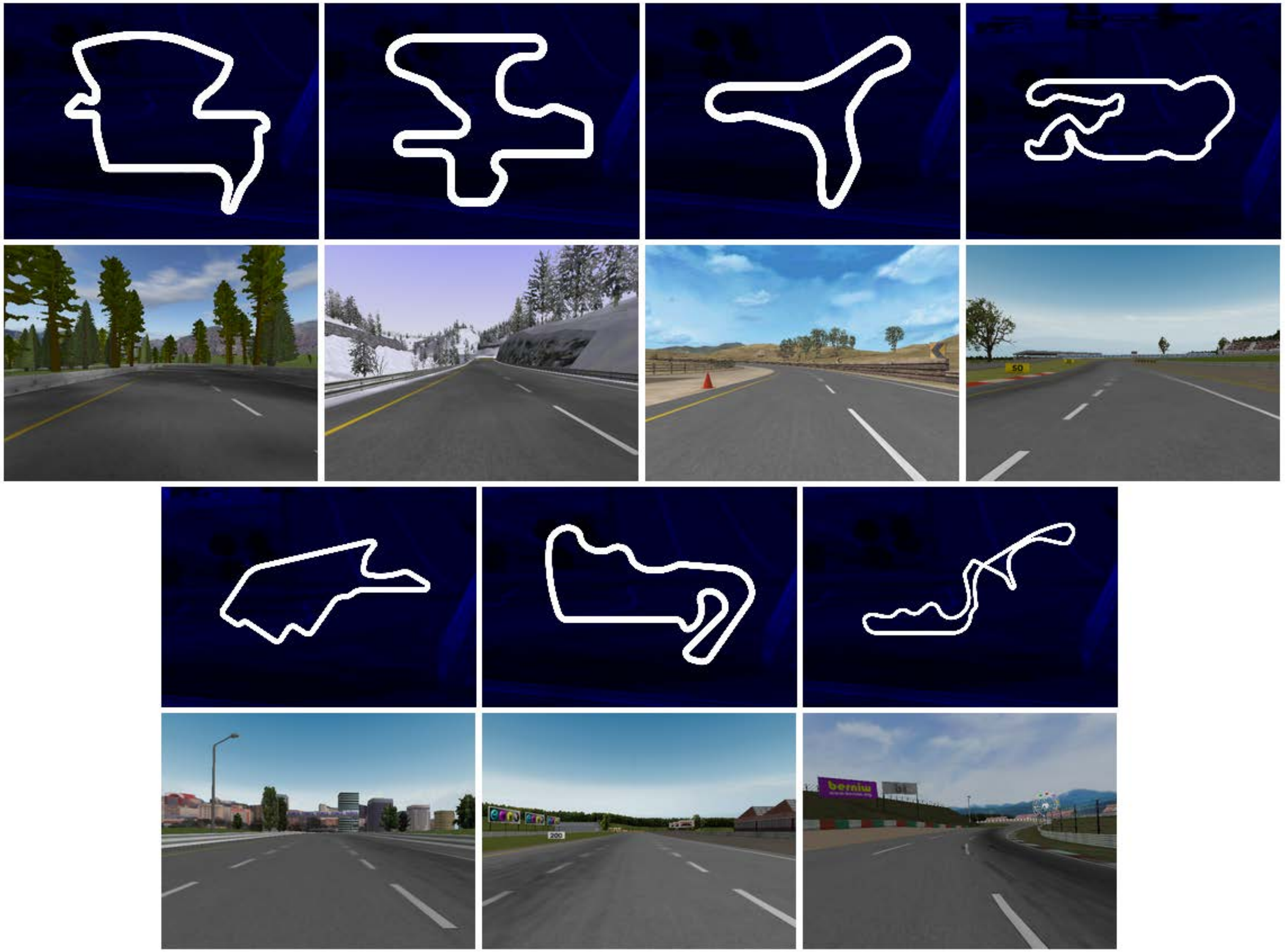


# In our experiment

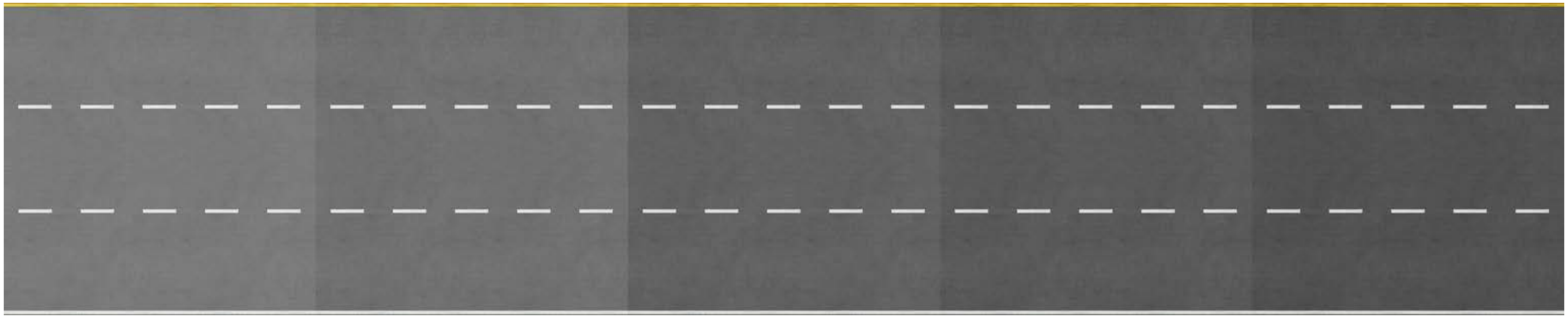
- Let the deep learning CNN drive in a racing game -- TORCS

# How to train the system?

# Seven Training Tracks



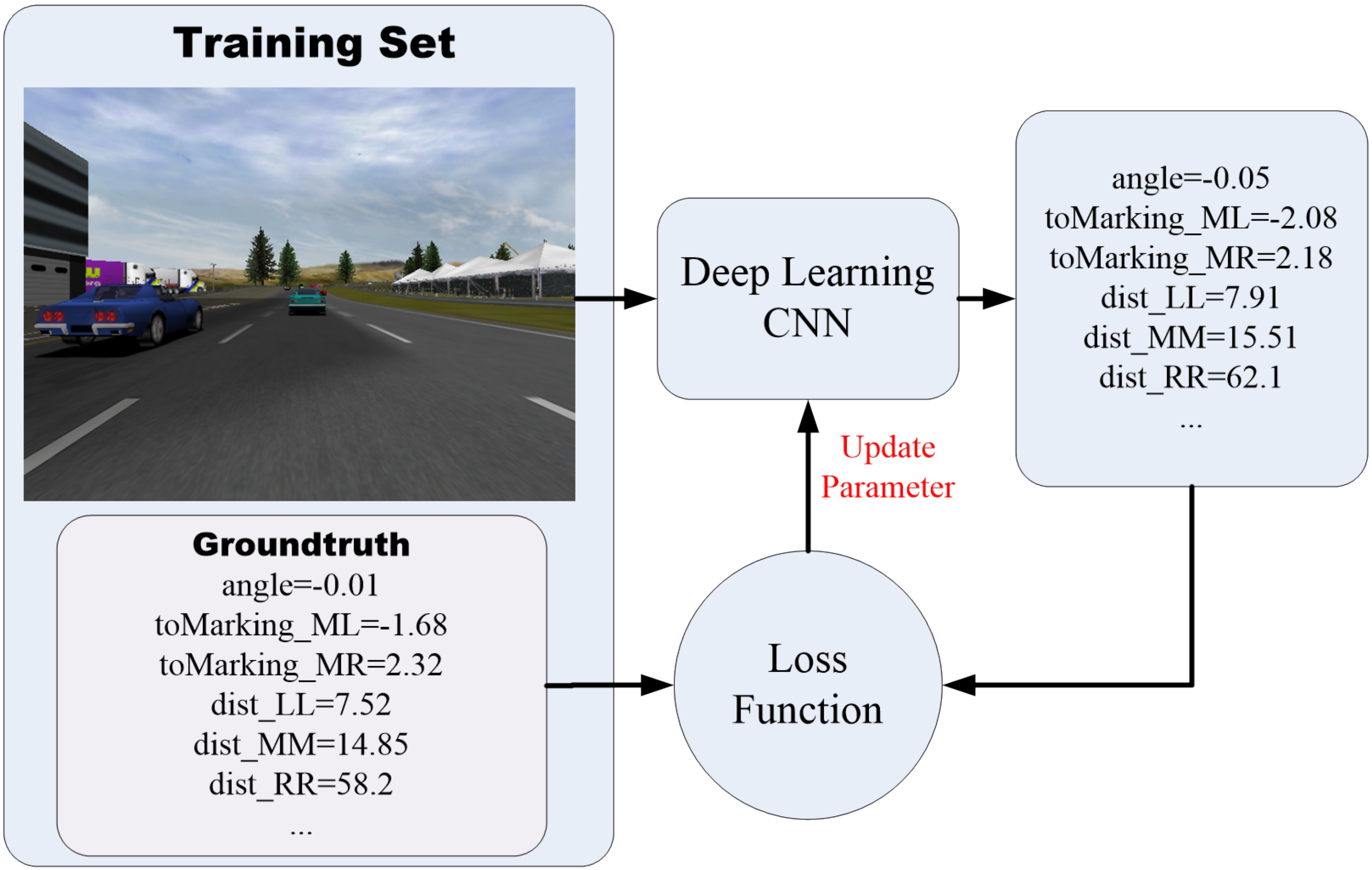
# Multiple Asphalt Darkness Level for the Training Tracks



# 22 Training Cars

# CNN during training phase

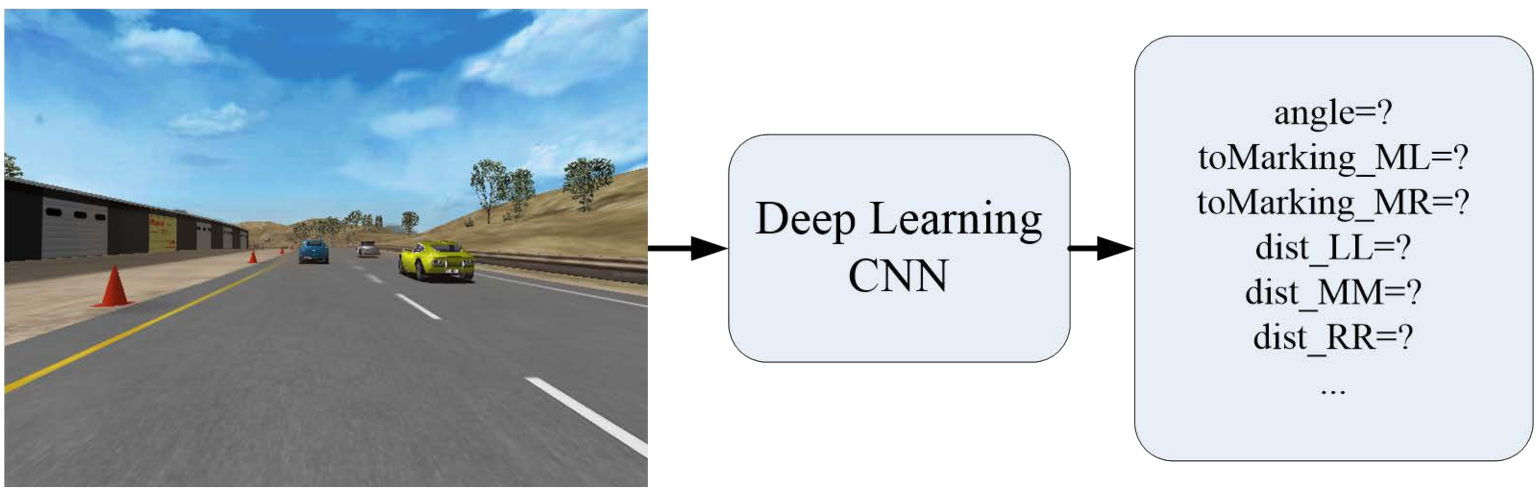
- Supervised learning, ground truth extracted from the game engine



# How to run the system?

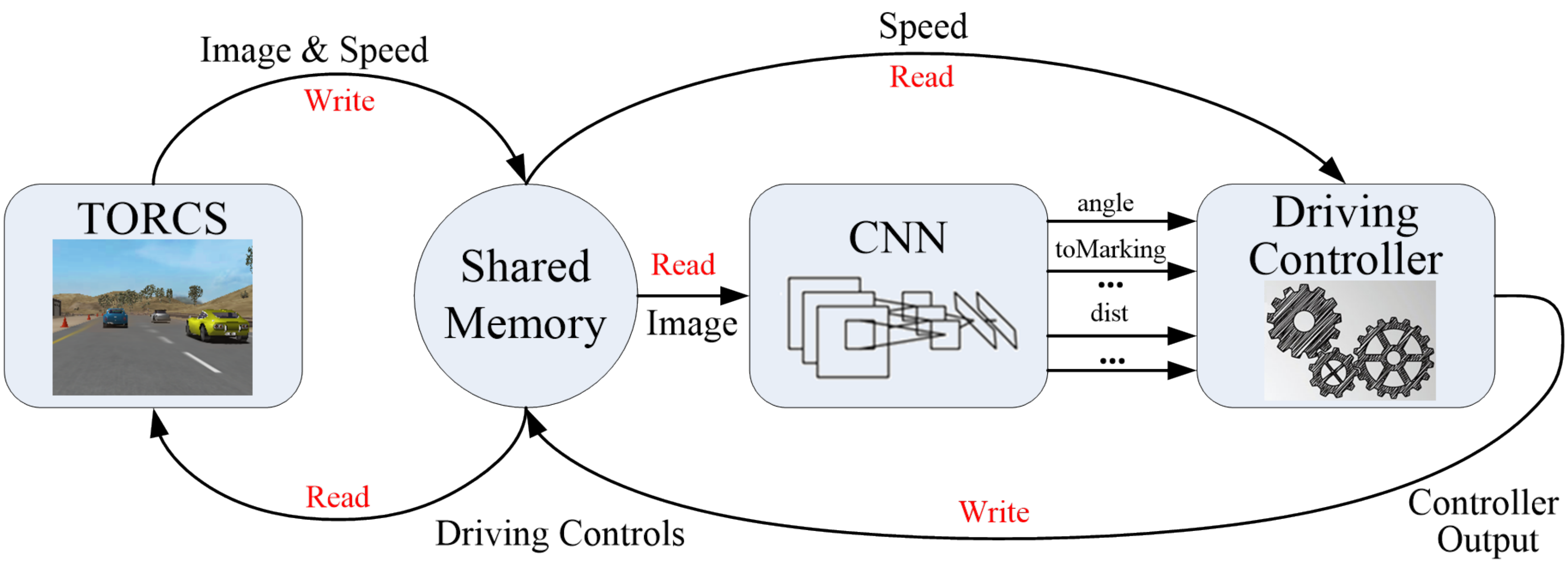
# CNN during testing phase

- Unseen track with unseen cars

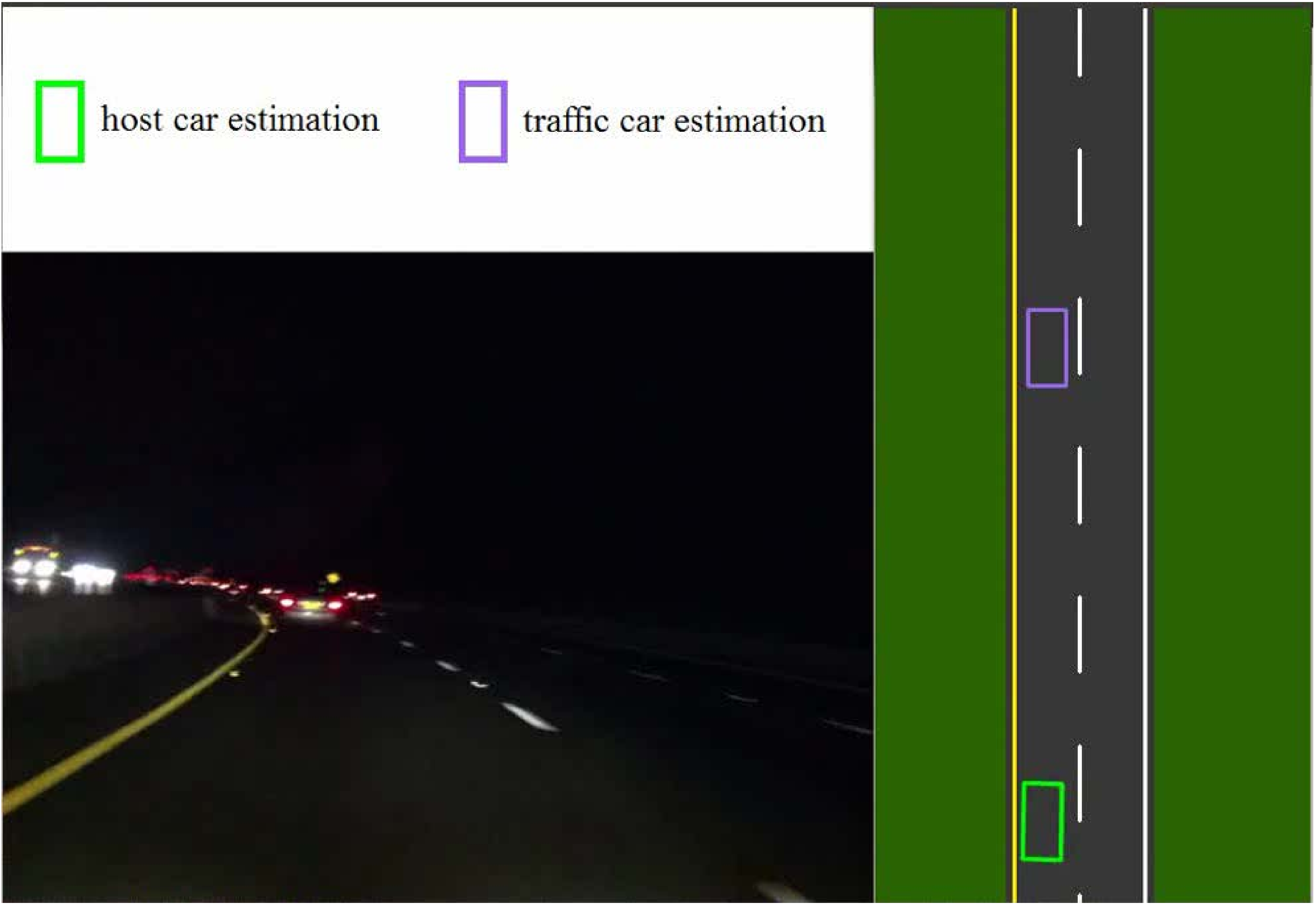


# How does the successfully trained system drive a car?

- The system runs at 10Hz, real time



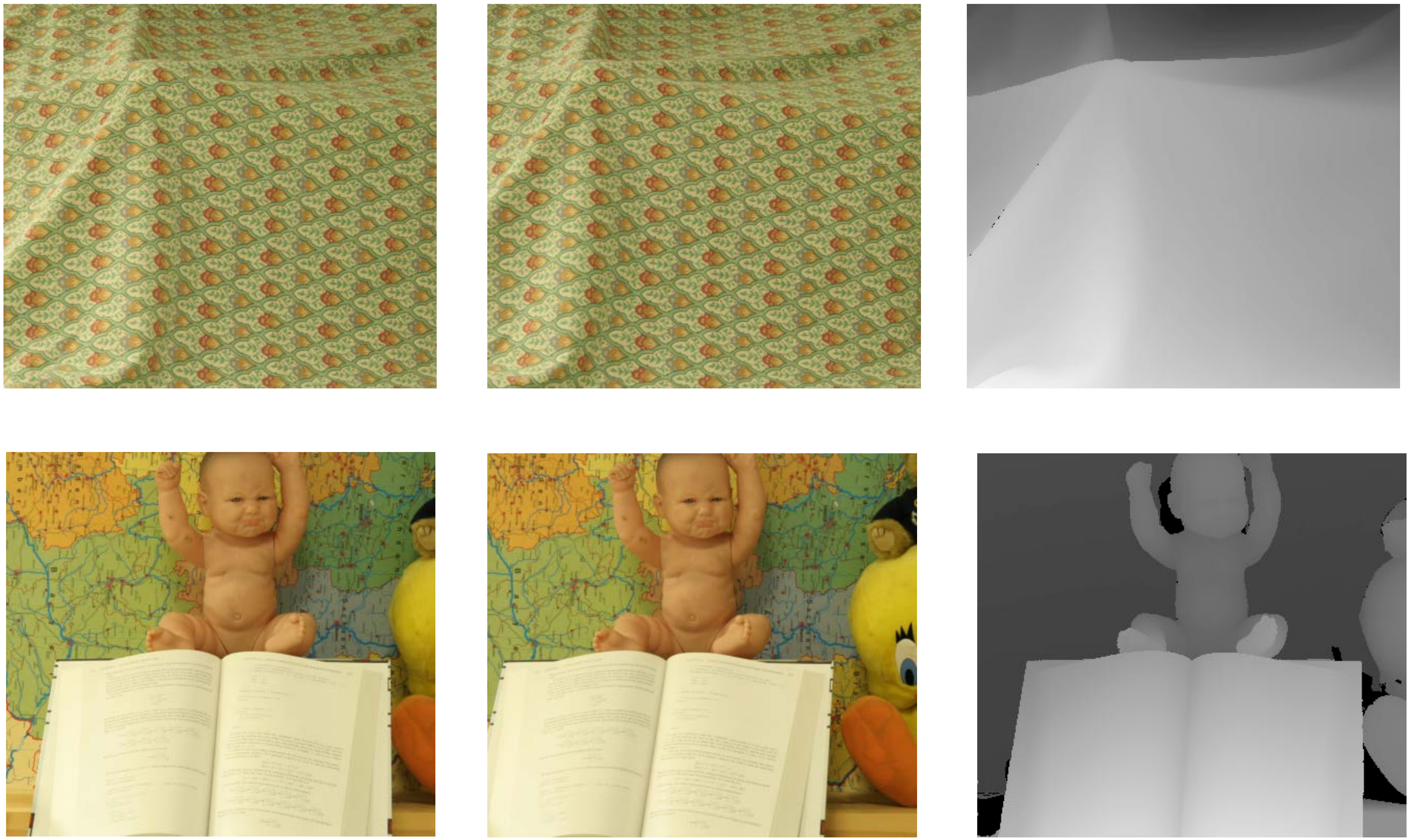
# A Challenging Test:
# Perception during Night Driving


host car estimation
traffic car estimation

# Next step: incorporating temporal information

# Other Interesting Stuffs

# A Taste of Stereo Vision

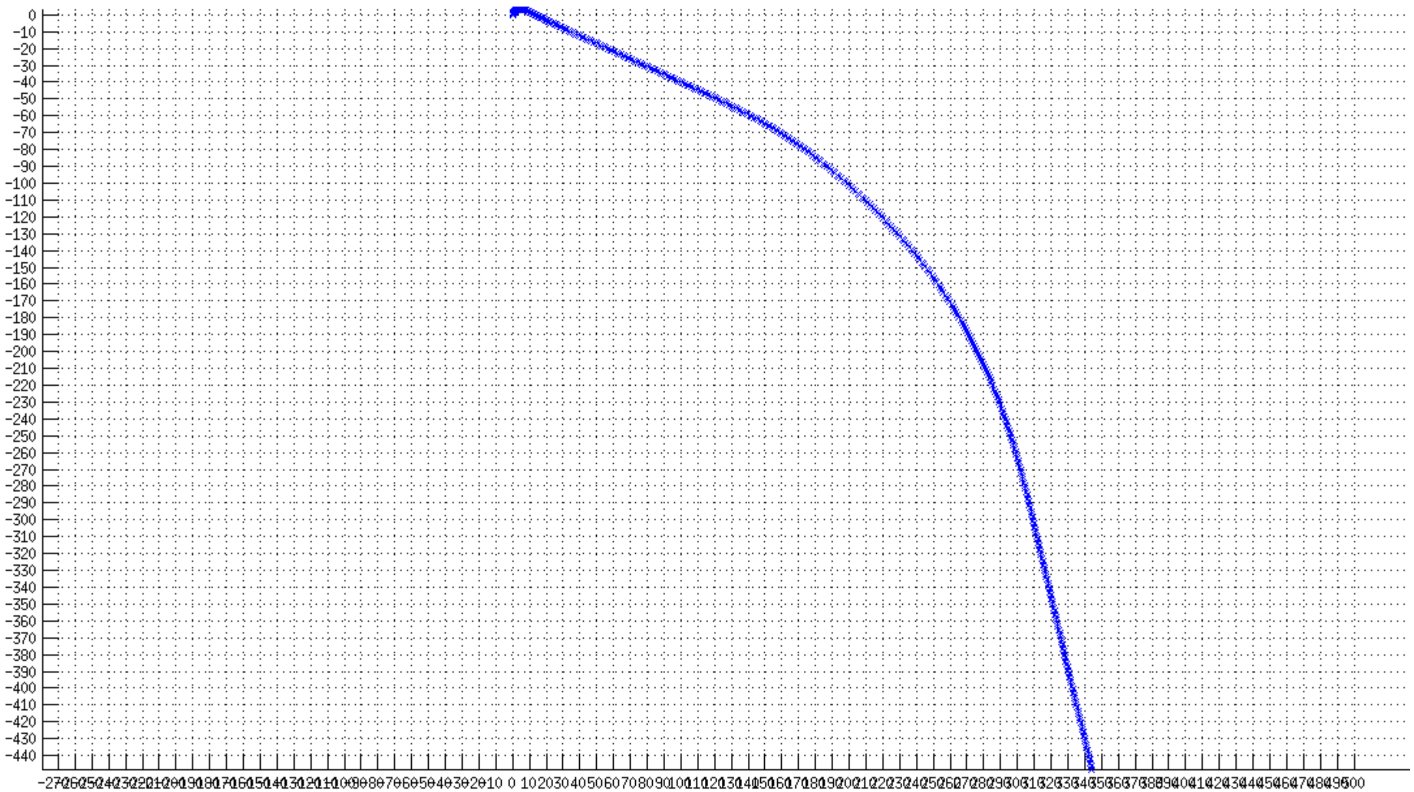


# 3D Scene Reconstruction

- Stereo images
- Color
- 1382*512
- 10 FPS

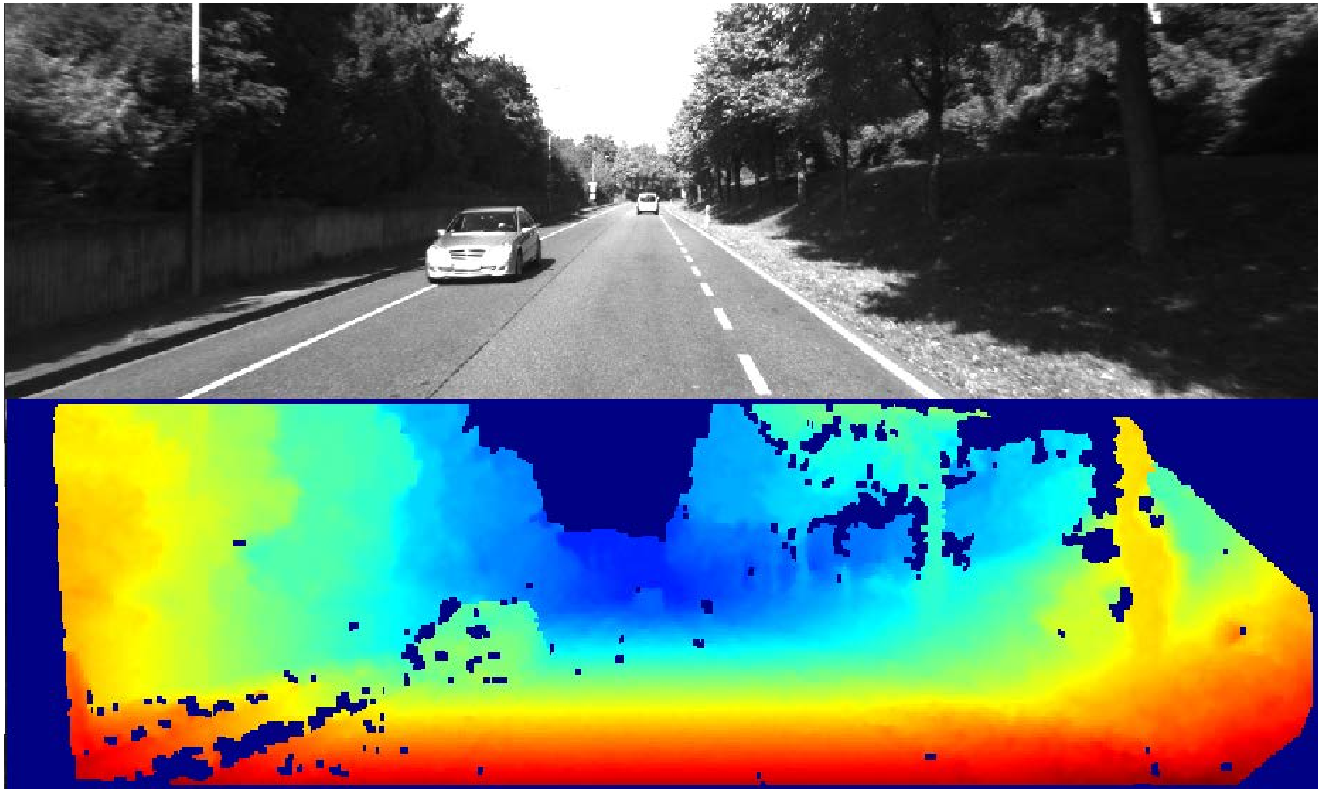
# Visual Odometry

- Visual odometry computes the trajectory of the vehicle only based on image sequences



# Depth Map

- Disparity map is computed from grayscale stereo image pairs
- Depth map can be derived from disparity map and camera model



# Other Demos for Structure-from-Motion

- https://www.youtube.com/watch?v=i7ierVkXYa8
- https://www.youtube.com/watch?v=vpTEobpYoTg

# Other Demos for Structure-from-Motion

# Other Demos for Structure-from-Motion

# Q & A