



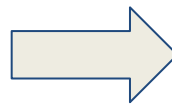
# Steering Angle Control with end-to-end Neural Networks

**Zhengyuan Yang, Jerry Yu**  
**08/30/2017**



# Objective

- *Taking an image or image sequences as input, predict steering angle end-to-end with neural networks.*
  - Input can include other side information like speed or past steering.
  - Output control can be extended to lane change, speed control and etc.
- *Nvidia first propose this idea*
  - Focus on the Nvidia drive PX computing platform.
  - Collect large amount of data under different situations. (3000 miles)
- **SAIC IC**
  - Work on general PC platform.
  - Start with parking lot scene and ideal road conditions.





# Objective

Nvidia's demo on Oct. 2016.





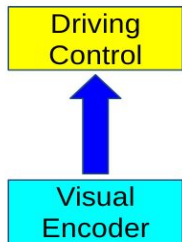
# Technical Approach

## ❖ Background: Major Approaches



**Behaviour reflex CNN\*:** Directly predict steering angle from input images.

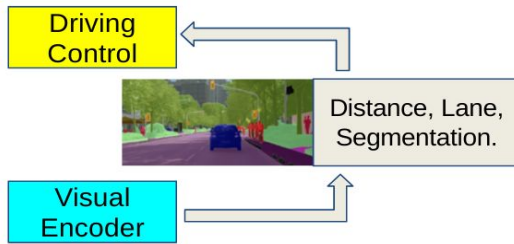
- Low model complexity. Robust with enough training data. Good generalization ability.
- Poor performance in complicated environment.



Behaviour reflex CNN

**Mediated Perception:** Learn surrounding scene first. And then control with rule-based methods.

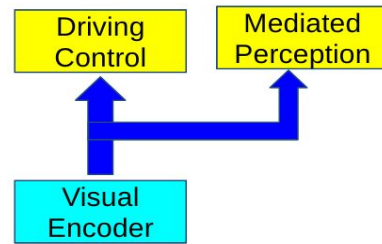
- Better vehicle smoothness.
- One specific model works in limited scenarios. Control rules designing is difficult.



Mediated Perception

**Privileged Training:** Learn the surroundings and generating driving control commands simultaneously.

- Better performance in complicated environments.
- Large amount of training data required.



Privileged Training

\* CNN: Convolutional Neural Networks



# Technical Approach

## ❖ Background: *Important Previous Works*



### *Nvidia's PilotNet:*

Nvidia use a single frame based CNN. The major contributions are the **engineering tricks** to make the model work on real car test.

### ● *Visual Attention Net:*

Jinkyu's work focus on **explaining the end-to-end system**. They introduce the **visual attention**.

### ● *Segmentation Net:*

Xu's work introduce a **side task** to improve the prediction performance. They try to predict the steering command and segmentation of the scene simultaneously.

Approach	<i>Behaviour reflex CNN</i>	<i>Behaviour reflex CNN</i>	<i>Privileged Training</i>
Model	Single frame based CNN.	LSTM for temporal, attention model on top.	LSTM for temporal. Segmentation as side taks. Discrete angle output.
Evaluation	No numbers. But work on <b>real car</b> .	Public Udacity, HCE and Comma.ai data.	Berkeley DeepDrive Video Dataset.

End to End Learning for Self-Driving Cars, *arXiv 2016*, Nvidia

Interpretable Learning for Self-Driving Cars by Visualizing Causal Attention, **Jinkyu Kim**, John Canny. *arXiv*, 2017. UCB

**H. Xu**, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. *arXiv*, 2016. UCB



# Technical Approach

## ❖ Our Method: *Work Focus*

<b>Approach</b>	Behaviour reflex CNN	Behaviour reflex CNN	Privileged Training
<b>Model</b>	Single frame based CNN.	LSTM for temporal, attention model on top.	LSTM for temporal. Segmentation as side taks. Discrete angle output.
<b>Evaluation</b>	No numbers. But work on <b>real car.</b>	Public Udacity, HCE and Comma.ai data.	Berkeley DeepDrive Video Dataset.

Our task is most similar to Nvidia's work.

We want the model to work on car with reasonable amount of data.

Therefore our work focus on:

1. Choose proper **visual encoder**, based on the data size and testing scene.
2. Adopt **engineering trick** for the model to work on car test, instead of only on recorded videos or simulators.
3. Go beyond single frame based approach. Include **temporal information** to improve accuracy and smoothness. Either on input images or output predictions.

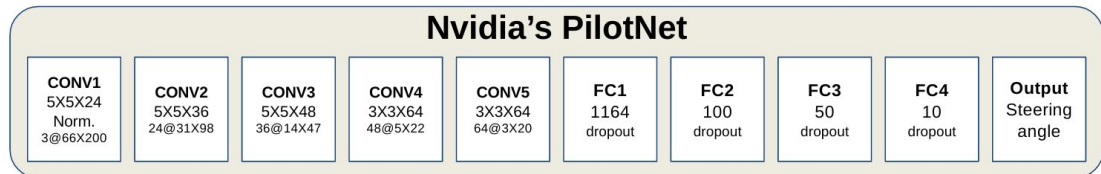
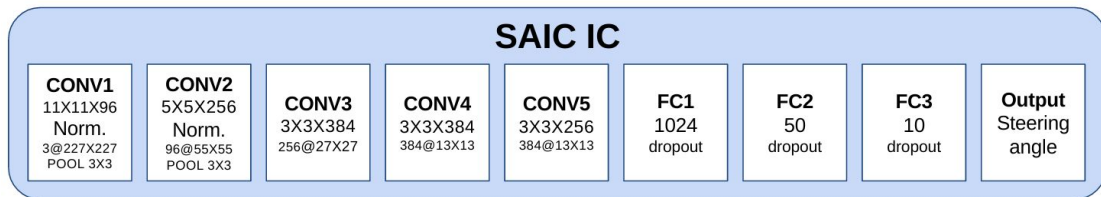
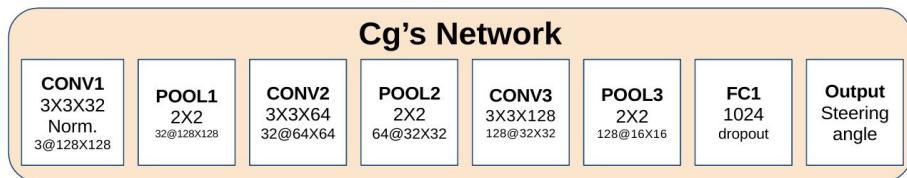


# Technical Approach

## ❖ Our Method: Visual Encoder

- **Convolutional Neural Networks (CNNs)** are used as the visual encoder.
- We start with **Nvidia's PilotNet** and **Chris Gundling's model** (short for cg) designed for an open challenge. Our final model is inspired by the structure of **AlexNet**.

- Based on experiments, our model is best balanced between model complexity and prediction accuracy.
- Popular large-size networks like VGG or ResNet tend to overfitting on the few hour's data we got.





# Technical Approach

## ❖ Our Method: *Engineering Tricks*

- Though some model might work well on recorded videos or even simple simulators, they fail to control a real car due to factors like error accumulating and noises in real world.
- One main work is to **introduce side cameras for error case generation**. It helps to prevent error accumulation, which will cause the car to slowly drift away.



- Besides side cameras, we also include color space change (RGB to HSV) to improve the robustness under lighting change, random rotation to simulate camera vibration.





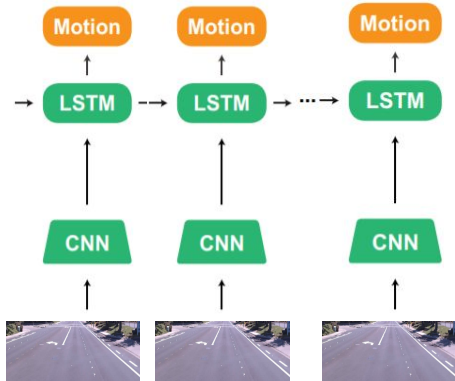
# Technical Approach

## ❖ Our Method: Temporal Model

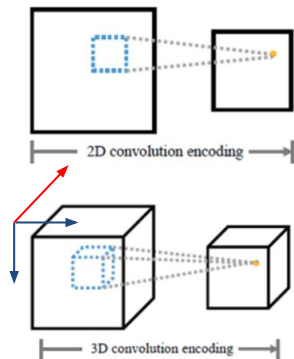
- Intuitively, prediction generated with both current and past information should be more smooth and accurate comparing to the single frame based prediction.

### Image sequence input

- Recurrent neural network.
- 3D convolution.



Recurrent neural network. (LSTM)



3D convolution

Red is the time dimension

### Output smoothing

- Single exponential smoothing with threshold.

$$\begin{pmatrix} \hat{\theta}_t \\ \hat{v}_t \end{pmatrix} = \alpha_s \begin{pmatrix} \theta_t \\ v_t \end{pmatrix} + (1 - \alpha_s) \begin{pmatrix} \hat{\theta}_{t-1} \\ \hat{v}_{t-1} \end{pmatrix}$$

And if change in prediction is even larger than the physical limit, this prediction will be considered as noise and won't be used to update past steerings.



# Technical Approach

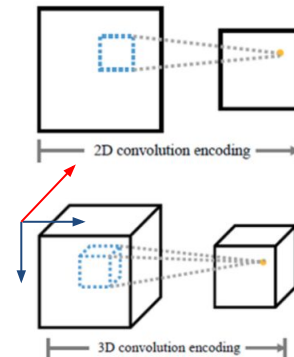
## ❖ Our Method: *Temporal Model*

→ *Recurrent neural network.*

We adopt the structure with three LSTM layers. According to experiment, a short memory length of 4 frames\* works best.

→ 3D convolution.

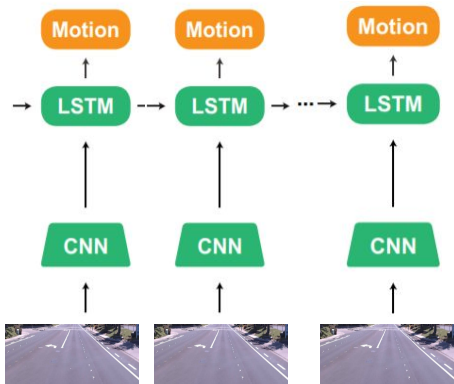
For 3D convolution, we use the temporal length of 10 frames.



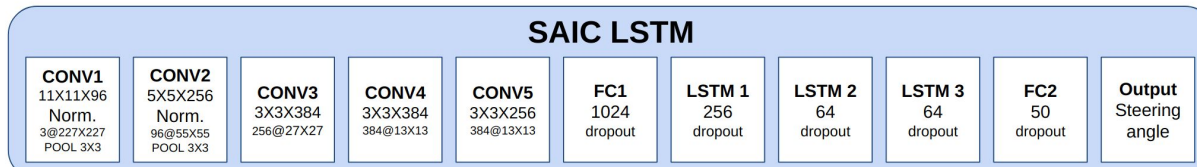
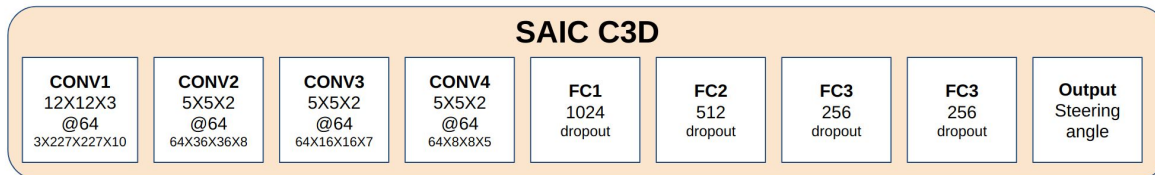
*3D convolution*

Red is the time dimension

\*Frame rate at 10 frames per second.



*Recurrent neural network. (LSTM)*








# Performance Evaluation

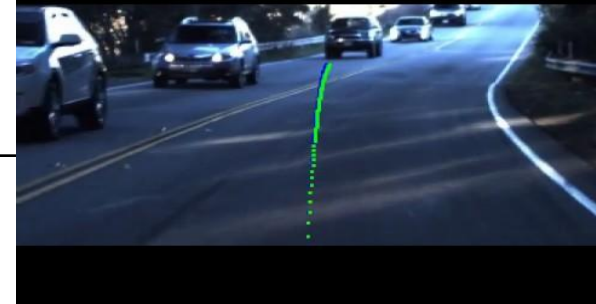
## ❖ Experiment Settings

- We've tested our methods under three dataset or scenarios.

Dataset	Udacity	SAIC Parking Lot	SAIC Road Test
Scenarios	Curvy Road. Avg. Two lane.	Open space (parking lot). Dots to formulate the road.	Area near company. Day and Night.
Data Size	~20 min	~20 min	~2 hours
Test Type	Recorded Video.	Recorded Video & Car Controlling	Recorded Video & Car Controlling
Example Frames			

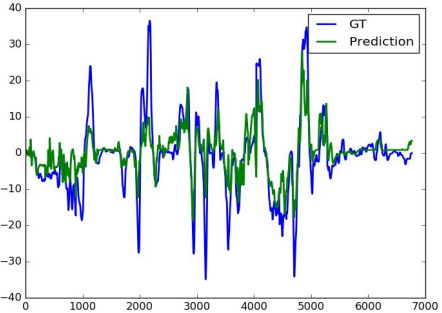
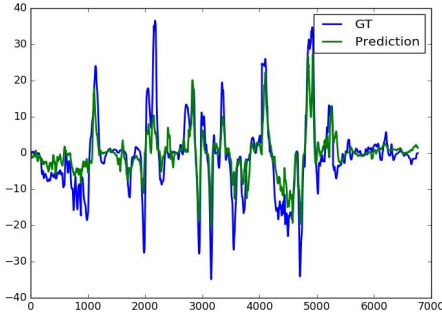
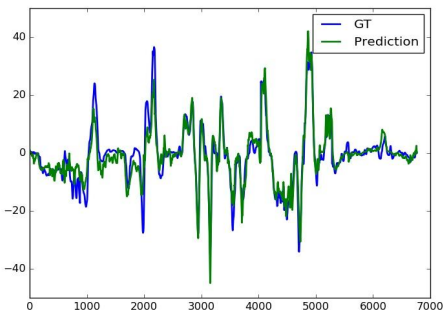


# Performance Evaluation



## ❖ Public Udacity Dataset

- Udacity provides an open dataset with several road conditions.
- In order to evaluate performance on each road type, we crop a segment from each long video pieces.

Model	Nvidia's PilotNet***	Cg's Network	SAIC IC
MAE*	4.260	4.175	2.844
Error Plots**			

\* Mean absolute error (MAE) in degree. The lower the better.

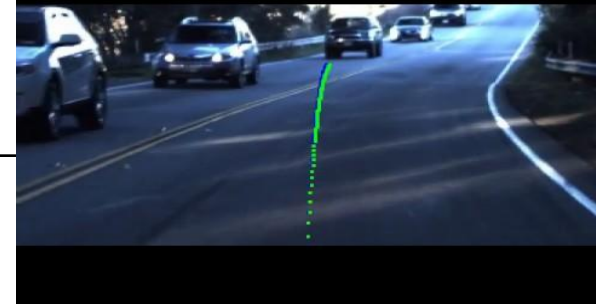
\*\* Blue: Ground Truth. Green: Prediction. *X: time Y: Steering angle*

\*\*\* Performance based on re-implementation. Preprocessing might be different.



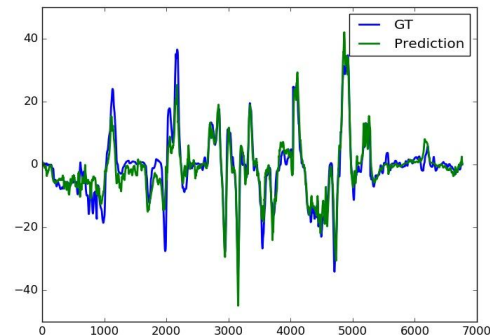
# Performance Evaluation

## ❖ Public Udacity Dataset



- Based on experiments, our model is the best on the udacity validation set.
- The major contribution is made by the design of convolutional layers. The **larger kernel size** fit the size of the potential features, and make feature extraction more accurate. Besides, the increased **number of filters** (kernels) also enables more features to be recorded.
- Our preprocessing techniques, including **synthetic view**, **data balance** and **data augmentation** also makes the results better.

- **As mentioned, a model works on recorded video didn't means it works on car controlling.**
  - **Drifting, real conditions.**
  - **Different though correct ways.**

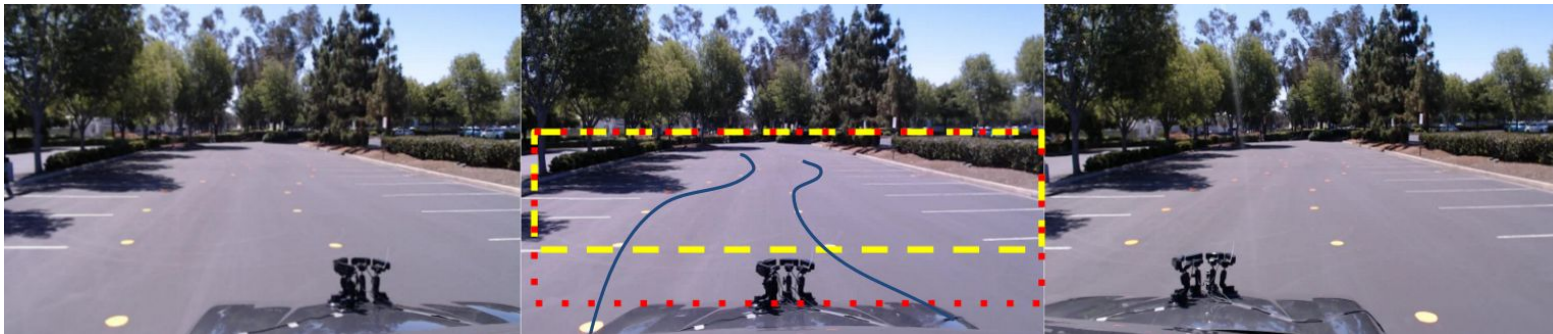




# Performance Evaluation

## ❖ Parking Lot Scene: *Experiment Setting*

- For safety concerns, real car tests are started in parking lots with artificially created routes.
- This route following task is more challenging than the general lane following task, as the markers are less clear and parking lots are open spaces.





# Performance Evaluation

## ❖ Parking Lot Scene: *Performance Evaluation*



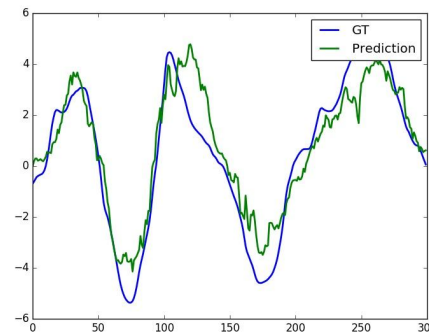
- We test the model on both recorded video and real car.

### ➤ First Road Test Performance Summary

- Manage to stay in lane under ideal condition.
- Prove the ability to recover from drifting to the side of the road.
- Limited performance when lighting condition changes violently.
- Poor control smoothness.

### ➤ Video Simulation Performance

- Based on the feedback from first road test, model is improved and will be used for second road test.
- Our model prove good accuracy and robustness under different lighting conditions.







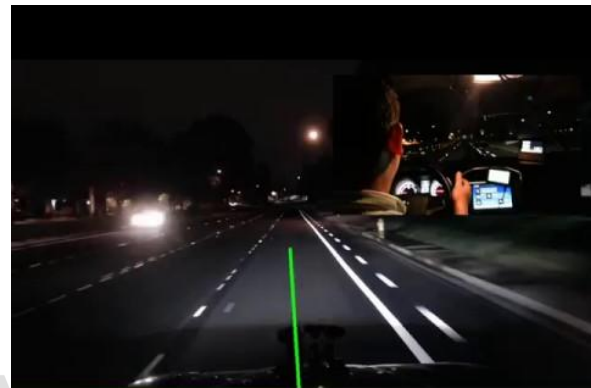
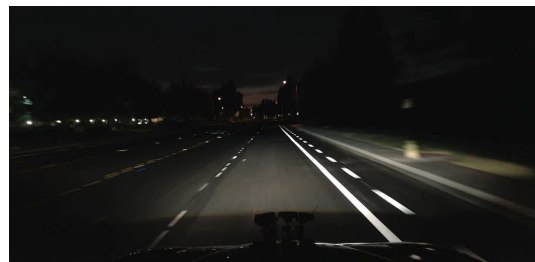
# Performance Evaluation

## ❖ Road Test

- Paralleling to the second parking lot test, we start to collect data and build model on roads in north San Jose area.
- We've got more than **five hour's** data.  
Recorded during different times of the day and night.

## ➤ Performance Summary

- To begin with, we build separate models for day and night's data.
- Since it is safer to test during the night and the task is more challenging, we start with the night scenarios.
- Here is a demo.







# Next Step

## ❖ Long Term & My Understanding

### My Understanding towards end-to-end control

- I think the end-to-end control can be a good add-on function, like cruise.
- Potential products could be:
  - ◆ Lane following on highway or local roads with less traffic.
  - ◆ Autonomous driving in traffic jam. E.g. 880 on Monday morning.
- It is hard to build a fully functional self-driving car only with this.

### Long Term Brainstorms (in research perspectives)

- Multitasking is a must. We need to get speed prediction or speed control command to make the system useful.
- Attention model or scene segmentation could be added. It helps to explain how the model works. Since it is hard to let the drivers trust the autonomous driving system, even if the system is robust enough.



**Thank you**

---

Thank you.

**SAIC USA INC.**