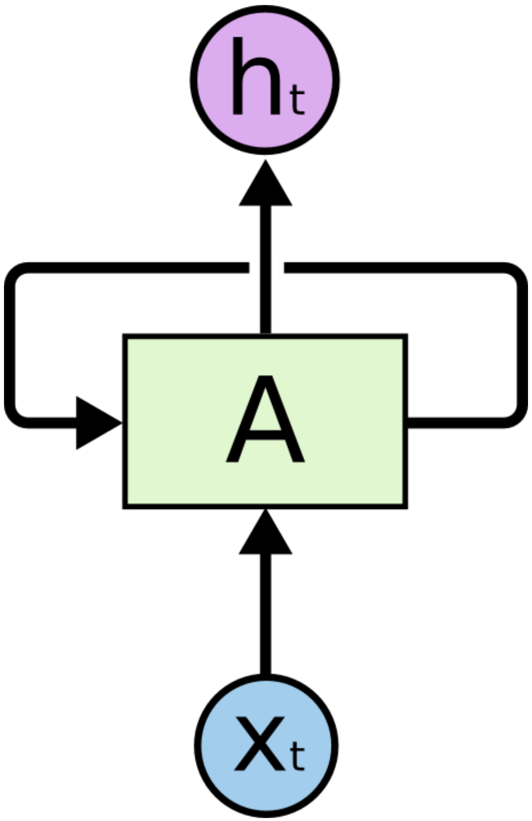
RECURRENT NEURAL NETWORKS

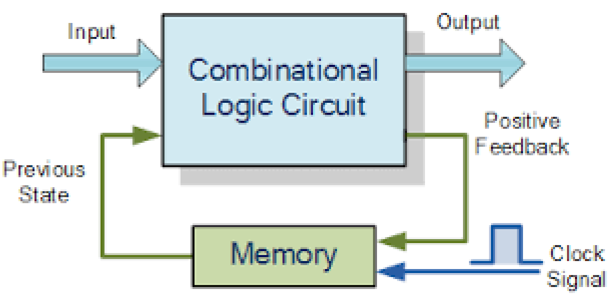
Methods					Tasks					
					ADAS					
					Self Driving					
					Localization	Perception	Planning/ Control	Driver state	Vehicle Diagnosis	Smart factory
Methods	Traditional	Non-machine Learning			GPS, SLAM		Optimal control			
Methods	Traditional	Machine-Learning based method	Supervised	SVM MLP		Pedestrian detection (HOG+SVM)				
Methods	Deep-Learning based	Machine-Learning based method	Supervised	CNN		Detection/ Segmentation/Classification	End-to-end Learning			
Methods	Deep-Learning based	Machine-Learning based method	Supervised	RNN (LSTM)		Dry/wet road classification	End-to-end Learning	Behavior Prediction/ Driver identification		*
Methods	Deep-Learning based	Machine-Learning based method	Supervised	DNN					*	*
Methods	Deep-Learning based	Machine-Learning based method	Reinforcement				*			
Methods	Deep-Learning based	Machine-Learning based method	Unsupervised							*

RNN INTRODUCTION

Recurrent Neural Network

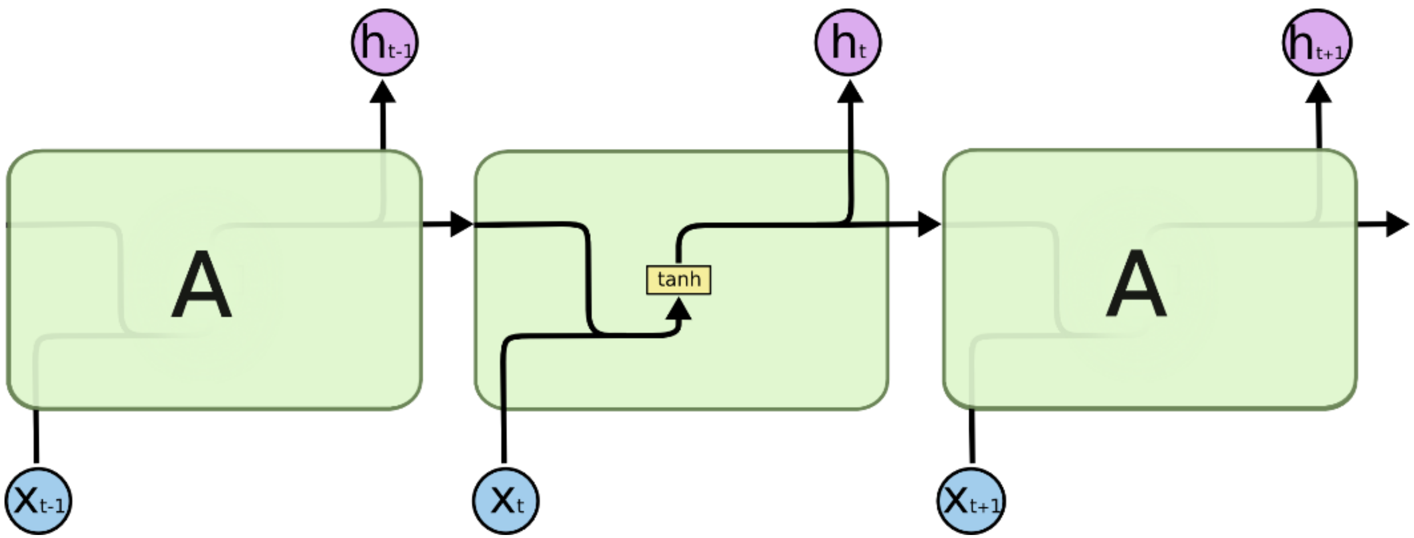


(참고)



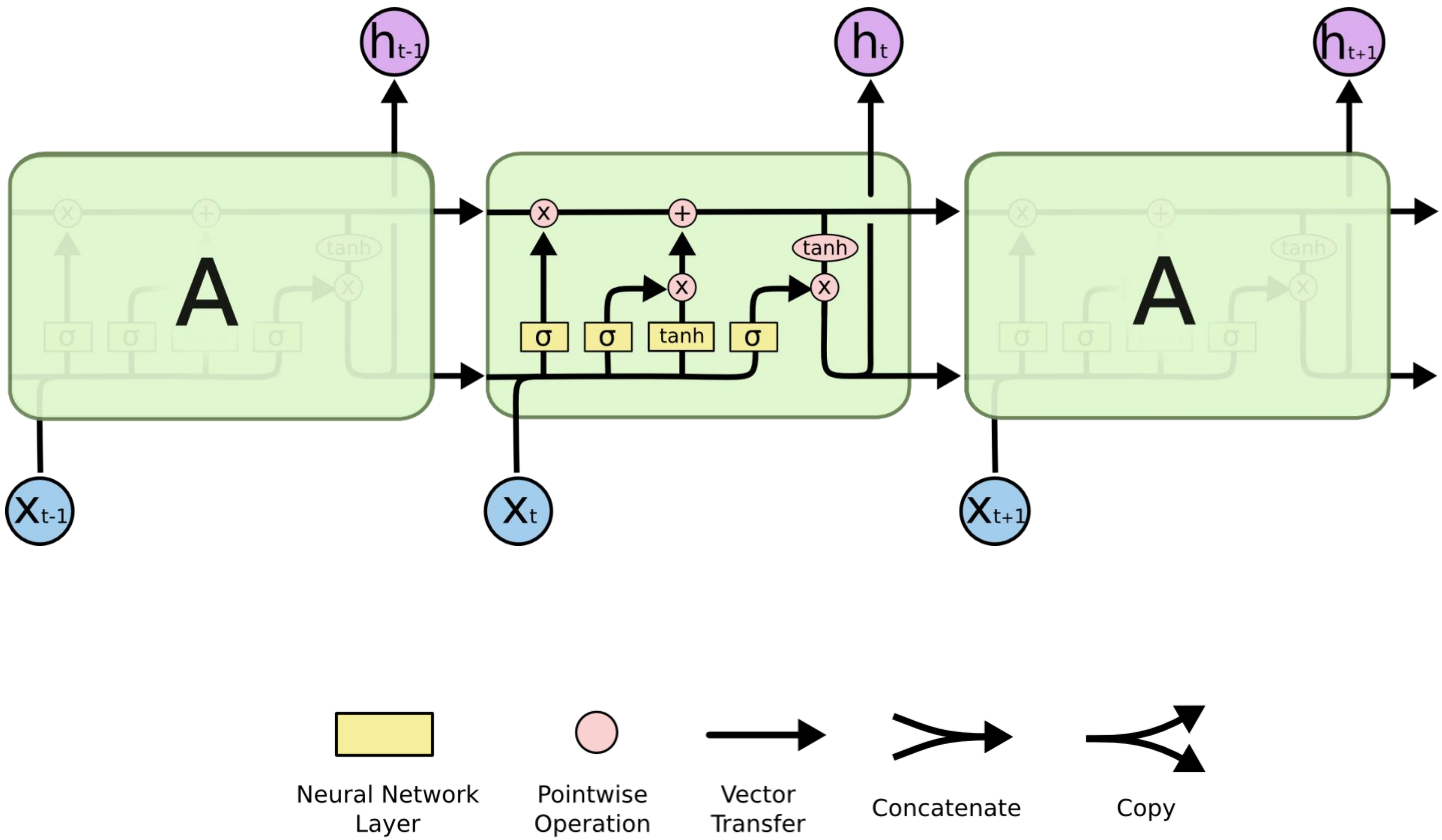
colah.github.io/posts/2015-08-Understanding-LSTMs/

Repeating module in RNN



ⓘ colah.github.io/posts/2015-08-Understanding-LSTMs/

Repeating module in LSTM



colah.github.io/posts/2015-08-Understanding-LSTMs/

Recurrent Neural Network

We can process a sequence of vectors **x** by applying a recurrence formula at every time step:

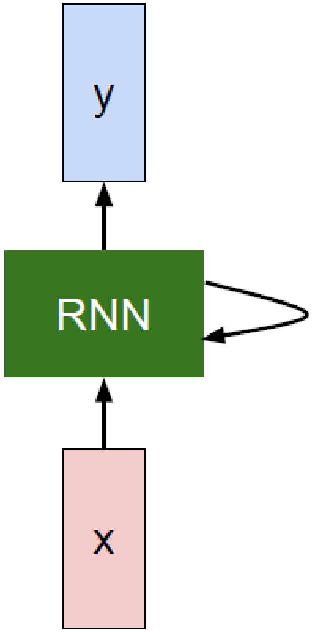
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

some function with parameters W

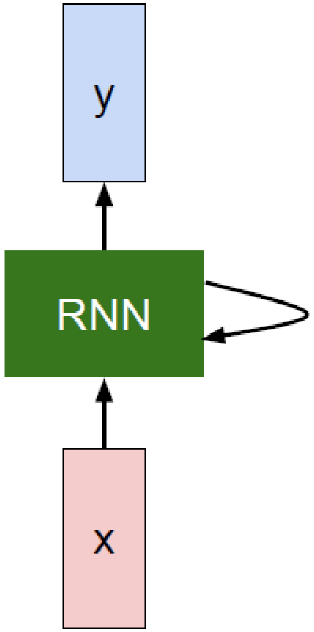


Recurrent Neural Network

We can process a sequence of vectors **x** by applying a recurrence formula at every time step:

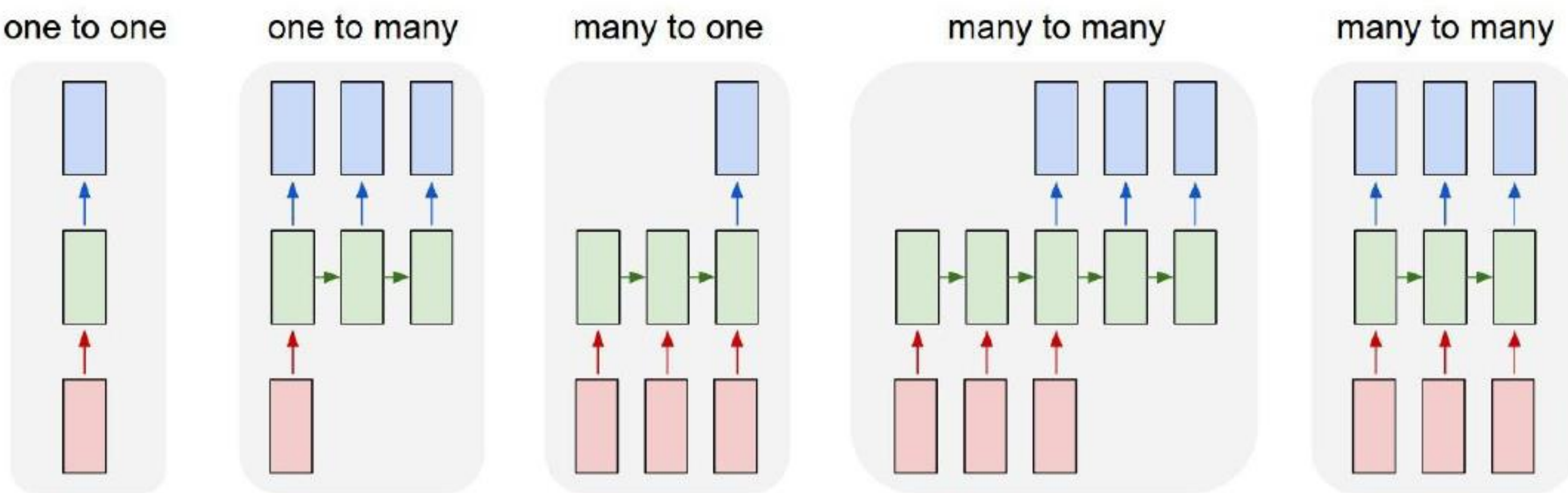
$$h_t = f_W(h_{t-1}, x_t)$$

Notice: the same function and the same set of parameters are used at every time step.



FLEXIBILITY OF RNN

RNN offers a lot of flexibility



Vanilla N.N.

Image Captioning

Sentiment Classification

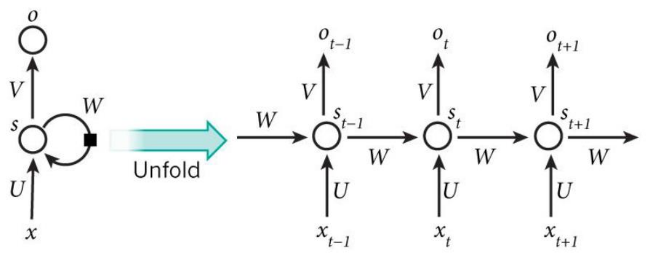
Machine Translation

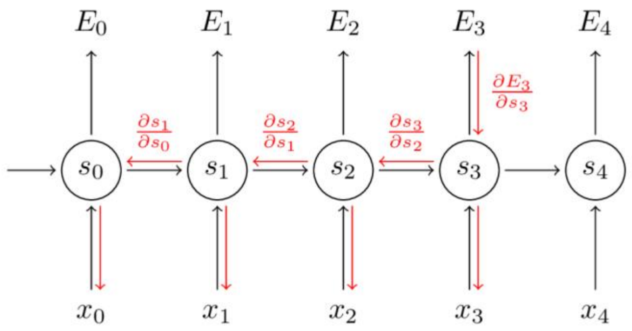
Video Classification on Frame Level

TRAINING

Training

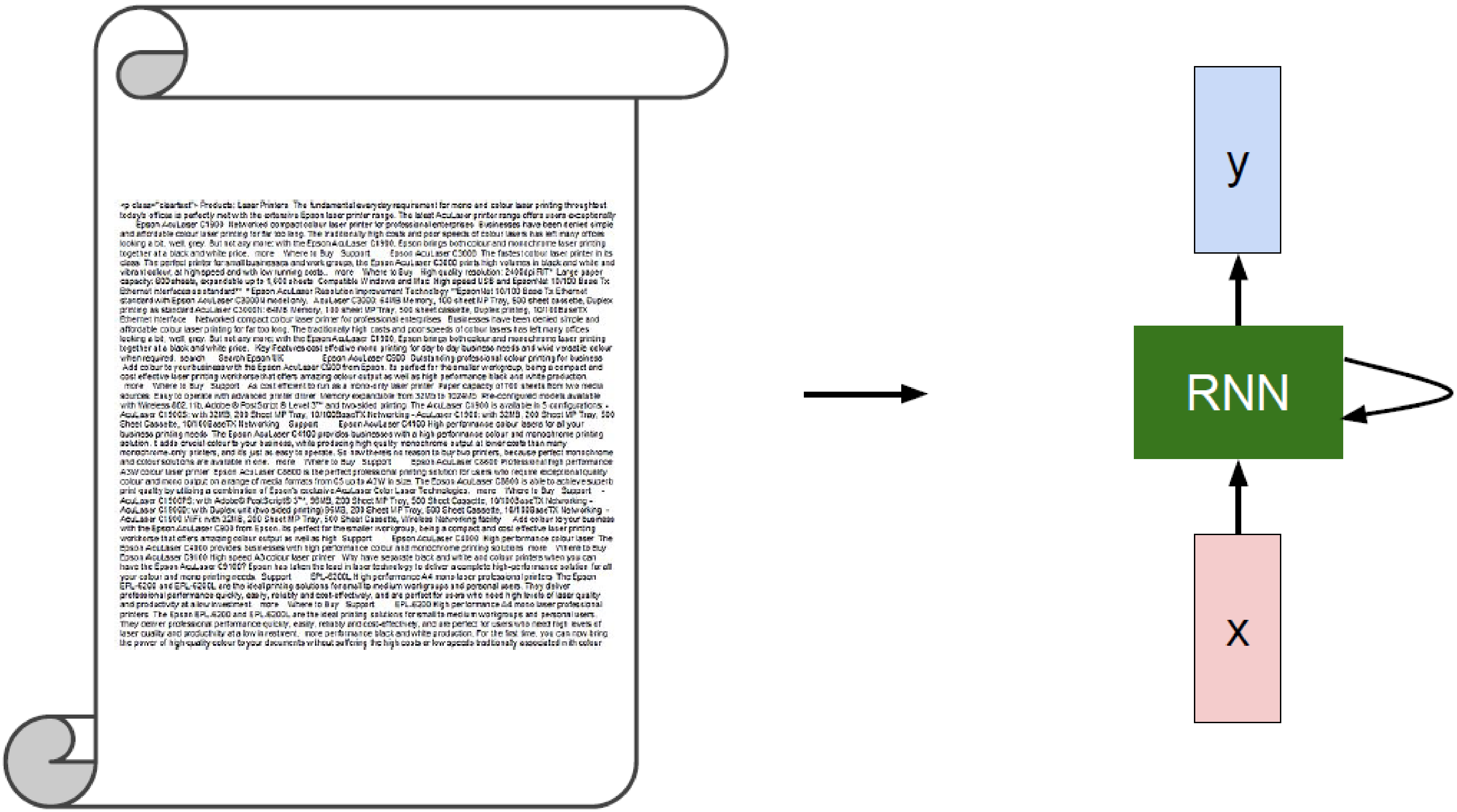
- Backpropagation through time (BPTT)
  - Unfold and apply SGD





EXAMPLE/VARIANTS

Training



Results

at first:

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

Generated C code

```c
static void do_command(struct seq_file *m, void *v)
{
  int column = 32 << (cmd[2] & 0x80);
  if (state)
    cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
  else
    seq = 1;
  for (i = 0; i < 16; i++) {
    if (k & (1 << 1))
      pipe = (in_use & UMXTHREAD_UNCCA) +
        ((count & 0x00000000fffffff8) & 0x000000f) << 8;
    if (count == 0)
      sub(pid, ppc_md.kexec_handle, 0x20000000);
    pipe_set_bytes(i, 0);
  }
  /* Free our user pages pointer to place camera if all dash */
  subsystem_info = &of_changes[PAGE_SIZE];
  rek_controls(offset, idx, &soffset);
  /* Now we want to deliberately put it to device */
  control_check_polarity(&context, val, 0);
  for (i = 0; i < COUNTER; i++)
    seq_puts(s, "policy ");
}
```

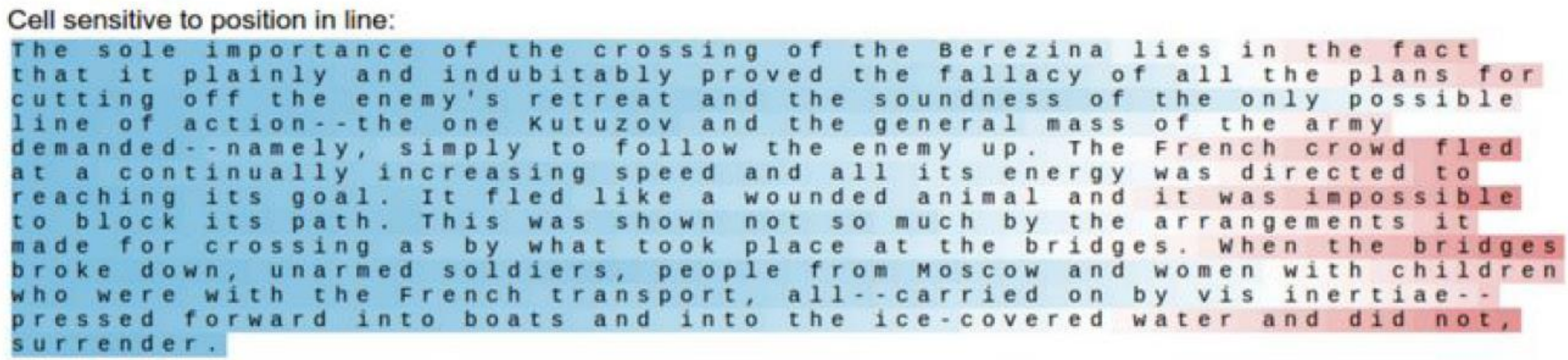
Searching for interpretable cells

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

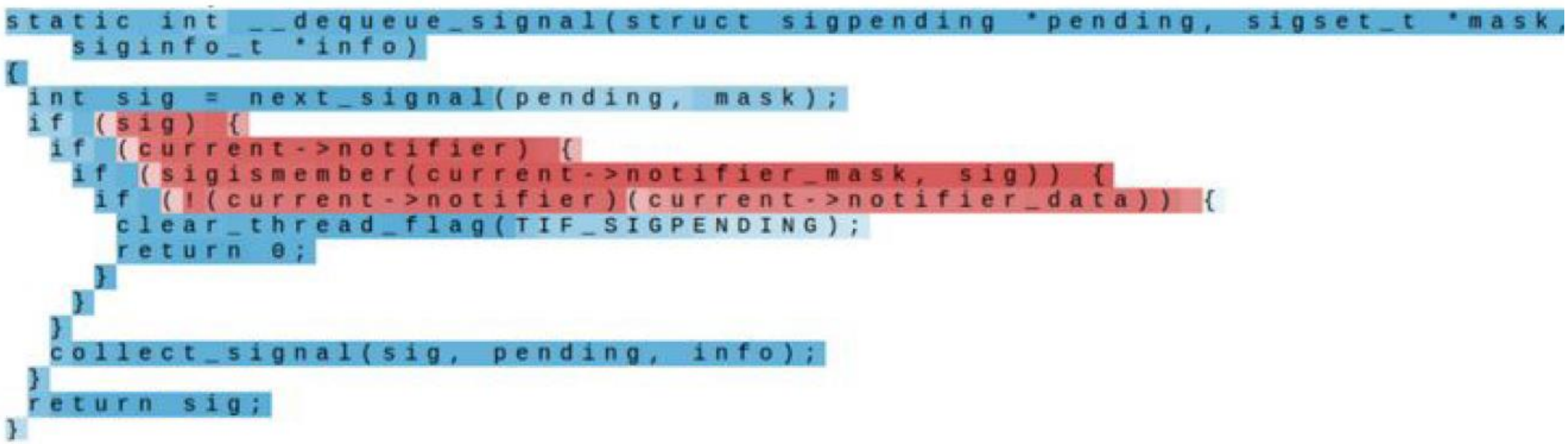
quote detection cell

Searching for interpretable cells



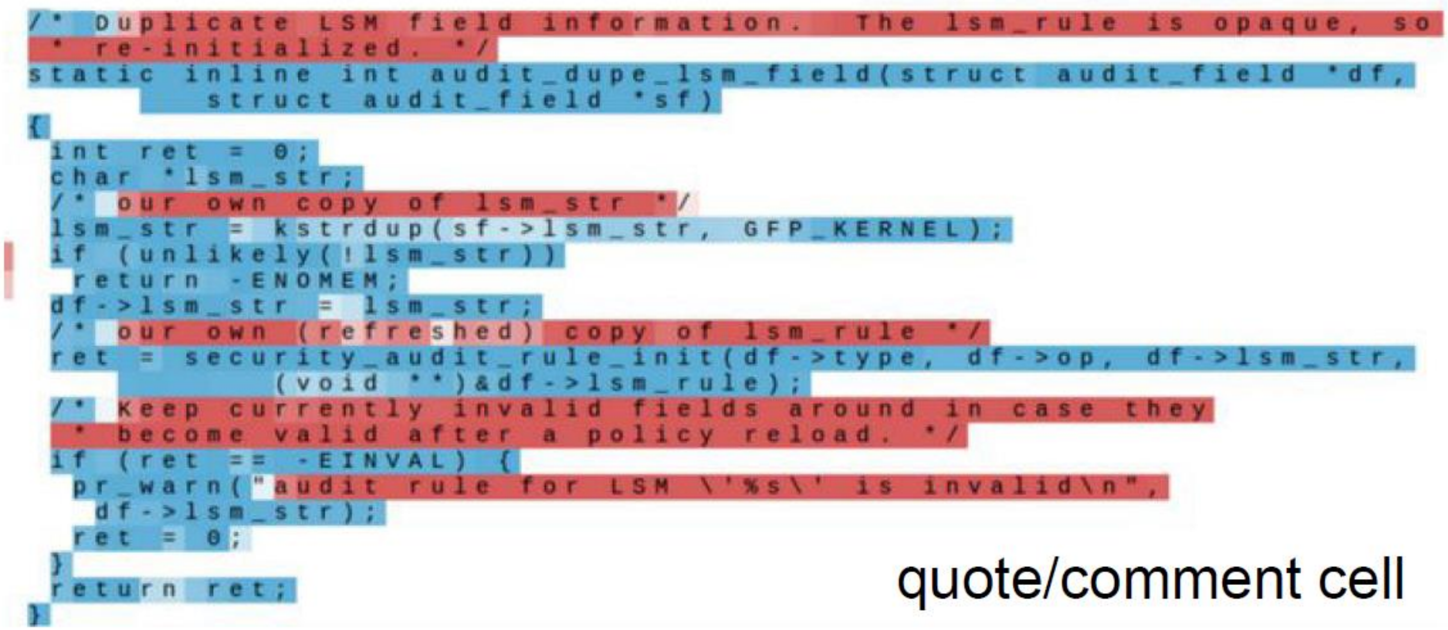
line length tracking cell

Searching for interpretable cells



if statement cell

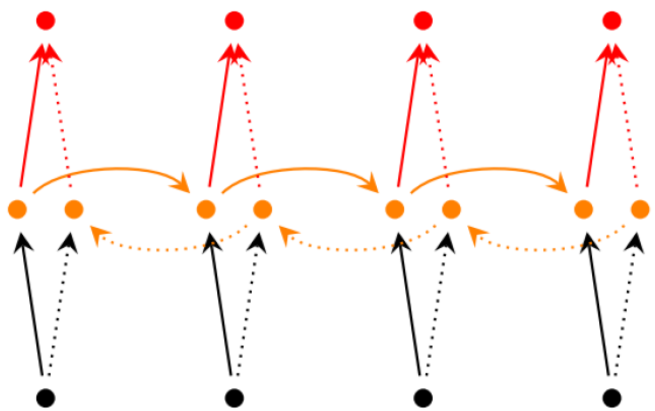
Searching for interpretable cells

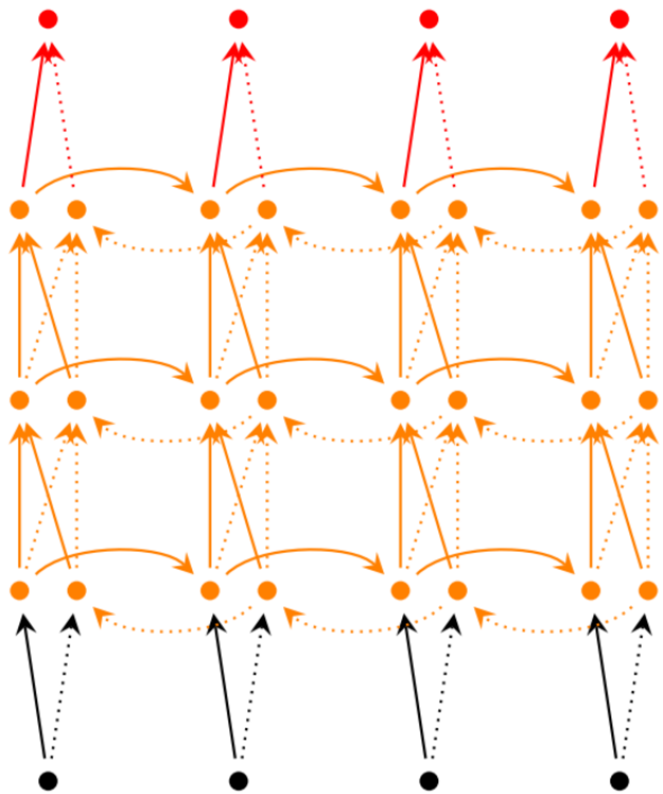


quote/comment cell

Variants

- Bidirectional
- Bidirectional Deep





예제코드

- LSTM

APPLICATIONS

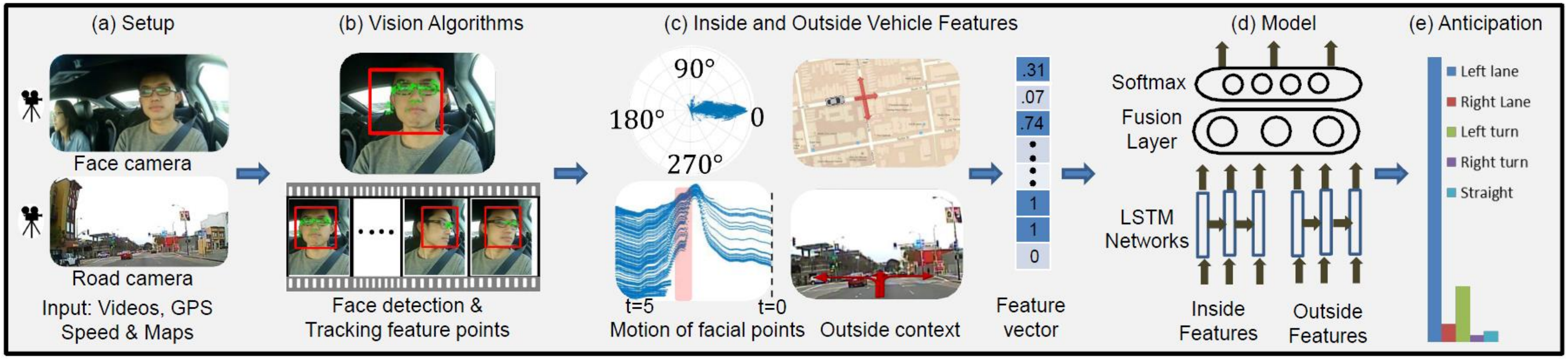
CAR THAT KNOWS BEFORE YOU DO VIA SENSORY–FUSION DEEP LEARNING ARCHITECTURE

ICCV 2015, Cornell Univ., Stanford Univ., Brain Of Things Inc

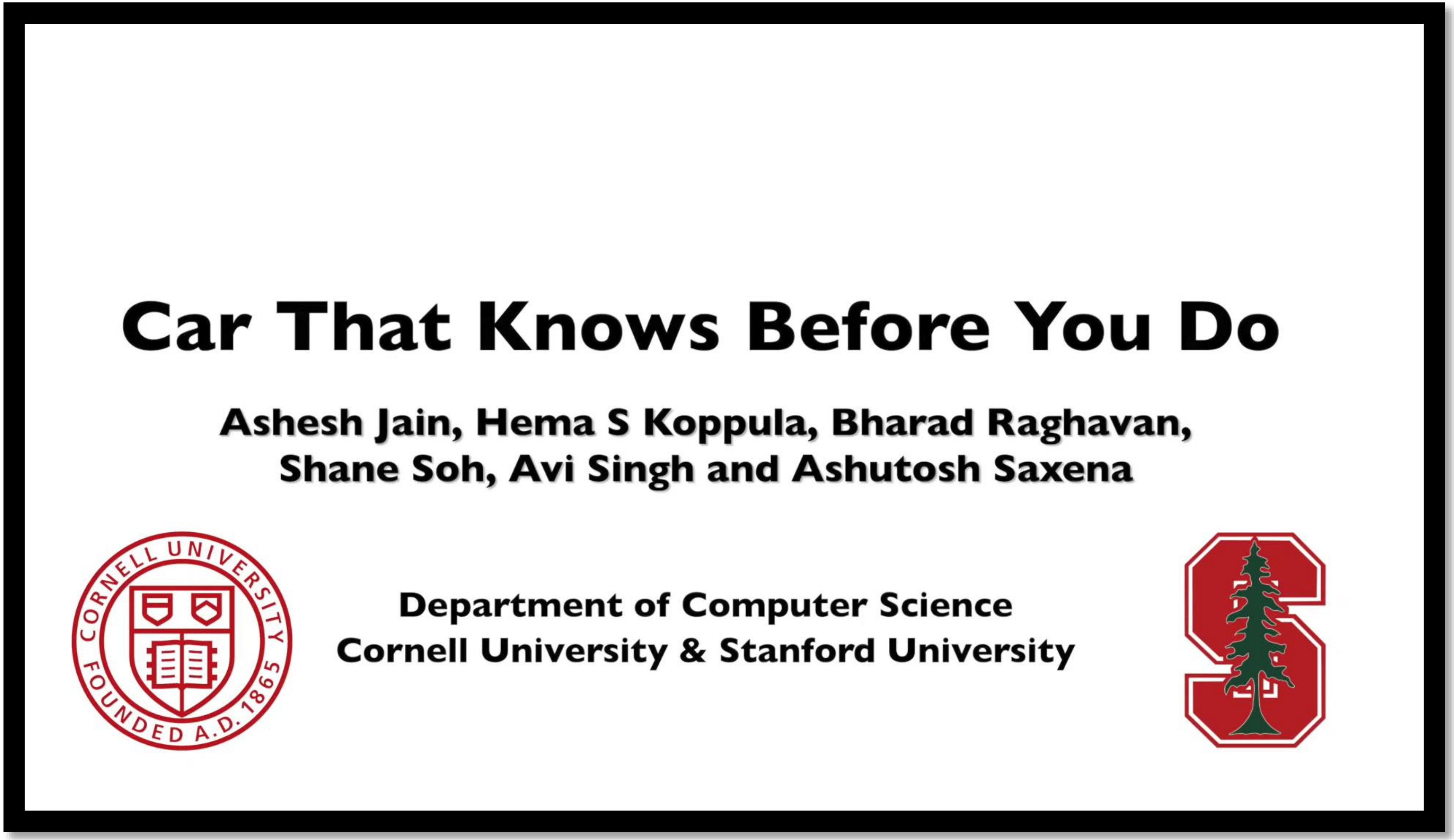
					Tasks					
					ADAS					
					Self Driving					
					Localization	Perception	Planning/ Control	Driver state	Vehicle Diagnosis	Smart factory
Methods	Traditional	Non-machine Learning			GPS, SLAM		Optimal control			
Methods	Traditional	Machine-Learning based method	Supervised	SVM MLP		Pedestrian detection (HOG+SVM)				
Methods	Deep-Learning based	Machine-Learning based method	Supervised	CNN		Detection/ Segmentation/Classification	End-to-end Learning			
Methods	Deep-Learning based	Machine-Learning based method	Supervised	RNN (LSTM)		Dry/wet road classification	End-to-end Learning	Behavior Prediction/ Driver identification		*
Methods	Deep-Learning based	Machine-Learning based method	Supervised	DNN					*	*
Methods	Deep-Learning based	Machine-Learning based method	Reinforcement				*			
Methods	Deep-Learning based	Machine-Learning based method	Unsupervised							*

Overview

- An approach for anticipating driving maneuvers, several seconds in advance: lane change, keeping straight, turn, ...
- Generic sensory-fusion RNN-LSTM architecture for anticipation in robotics applications

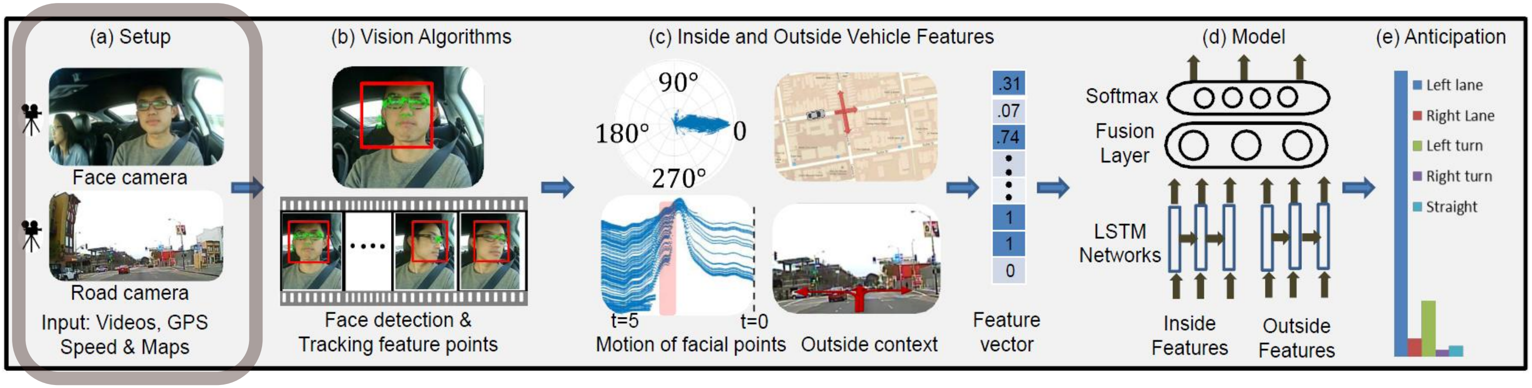


Demo Video



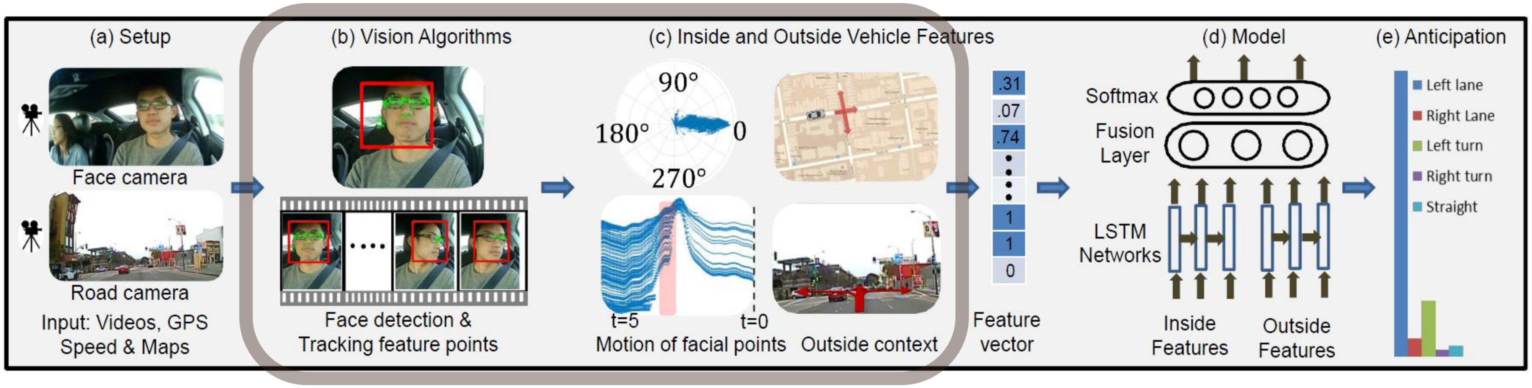
https://youtu.be/O5I1hBwkwJc

Setup



- Driver-facing camera inside the vehicle
- Camera facing the road
- Speed logger of the car
- Global Positioning System (GPS)

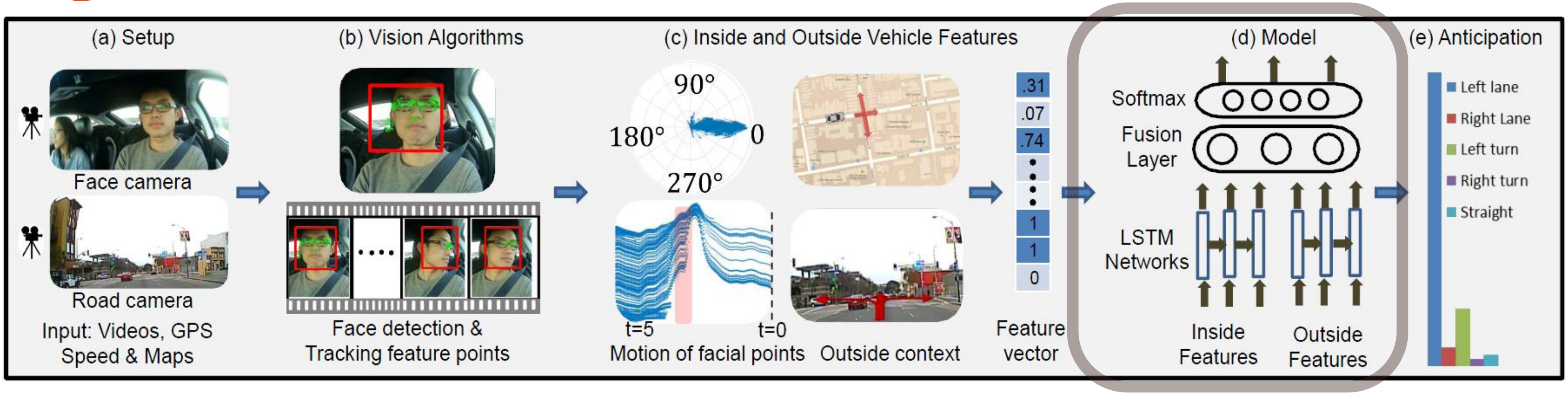
Features



1. Face detection and tracking:
   - Driver's face: Viola Jones face detector
   - Point extract: Shi-Tomasi corner detector
   - Facial points tracker: KLT(Kanade-Lucas-Tomasi)
2. Head motion features ($\phi_{face} \in \mathbb{R}^9$):
   - histogram features are used
   - matches facial points and
     create histograms of corresponding horizontal motions
3. 3D head pose and facial landmark features($\phi_{face} \in \mathbb{R}^{12}$): CLNF tracker model

- Aggregate ϕ_{face} for every 20 frames

Network Architecture – at a glance



LSTM units is used for training

- \- consider accumulated information from the past:

  noted as event $\{(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T), \mathbf{y}\}$ $\left(\begin{array}{c}\mathbf{x}_t\text{: } \textit{observation at time } \mathrm{t} \\ \mathbf{y}\text{:} \textit{representation of the event}\end{array}\right)$

  \- $\mathrm{y_t}$ is updated as $\mathbf{y}_t = \mathrm{softmax}(\mathbf{W}_y \mathbf{h}_t + \mathbf{b}_y)$,

  with the representation $\mathbf{h}_t = f(\mathbf{W}\mathbf{x}_t + \mathbf{H}\mathbf{h}_{t-1} + \mathbf{b})$

Data set

- Fully natural driving
- 1180 miles of natural freeway, city driving
- Collected across two states
- Ten drivers, different kinds of driving maneuvers
- 2 months to take
- About 17GB
- 700 events of annotation
  - 274 lane change
  - 131 turns
  - 295 straight

Result

- With sensory fusion:
  - Precision of 84.5%
  - Recall of 77.1%
  - Anticipates maneuvers 3.5 seconds on average
- With incorporating the driver's 3D head-pose:
  - Precision of 90.5%
  - Recall of 87.4%
  - Anticipates maneuvers 3.16 seconds on average

Method	Pr (%)	Re (%)	Time-to-maneuver (s)
F-RNN-EL	84.5 $\pm$ 1.0	77.1 $\pm$ 1.3	3.58
F-RNN-EL w/ 3D head-pose	**90.5** $\pm$ 1.0	**87.4** $\pm$ 0.5	3.16

CHARACTERIZING DRIVING STYLES WITH DEEP LEARNING

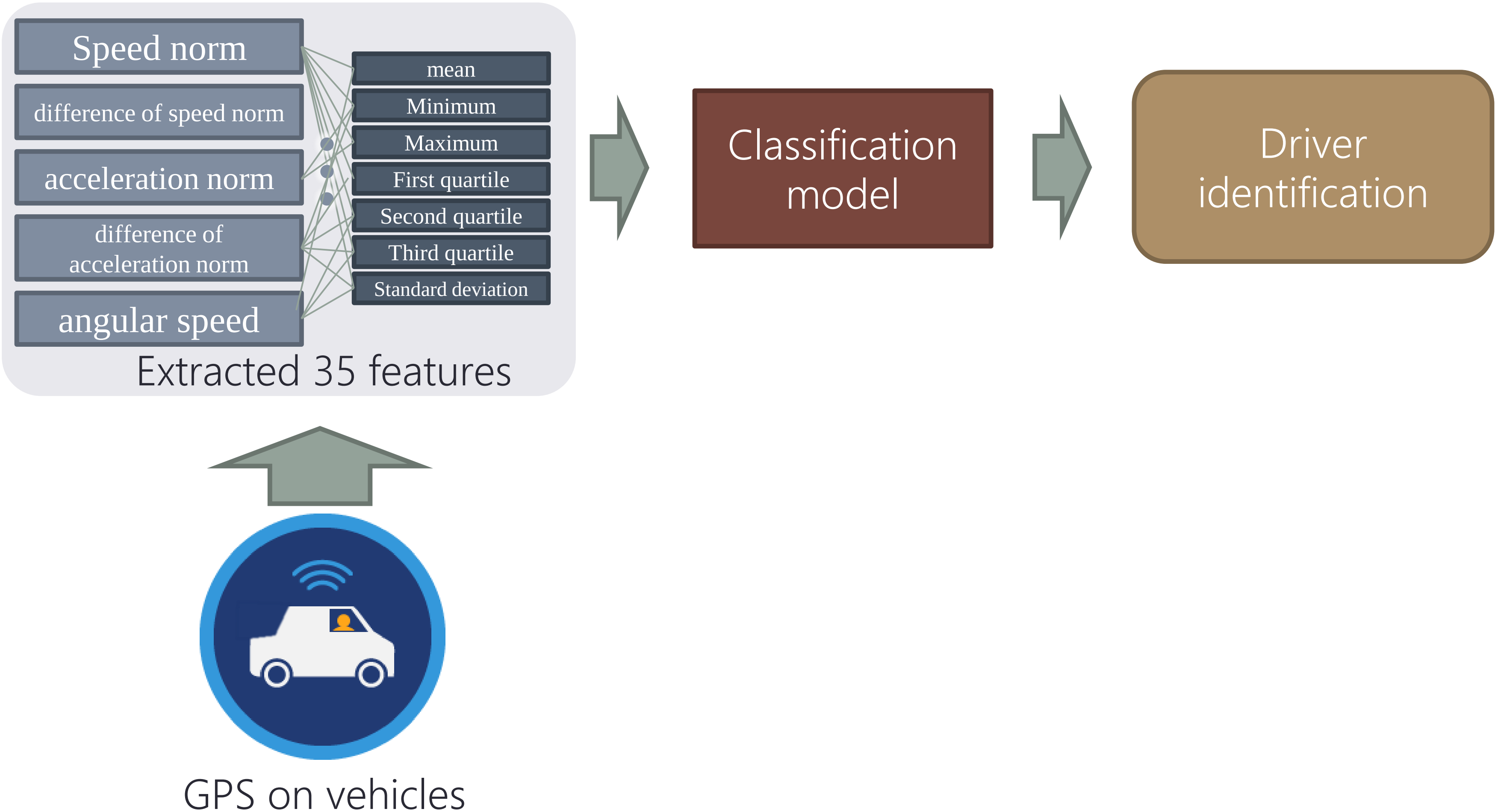
W. Dong et al.

ArXiv 2016

IBM Research China, Nanjing Univ., Univ. of Waterloo

					Tasks					
					ADAS					
					Self Driving					
					Localization	Perception	Planning/ Control	Driver state	Vehicle Diagnosis	Smart factory
Methods	Traditional	Non-machine Learning			GPS, SLAM		Optimal control			
		Machine-Learning based method	Supervised	SVM MLP		Pedestrian detection (HOG+SVM)				
	Deep-Learning based			CNN		Detection/ Segmentation/Classification	End-to-end Learning			
				RNN (LSTM)		Dry/wet road classification	End-to-end Learning	Behavior Prediction/ Driver identification		*
				DNN					*	*
			Reinforcement				*			
			Unsupervised							*

Overview


Speed norm
difference of speed norm
acceleration norm
difference of acceleration norm
angular speed
mean
Minimum
Maximum
First quartile
Second quartile
Third quartile
Standard deviation
Extracted 35 features
Classification model
Driver identification
GPS on vehicles

Data

- **목표**: GPS**데이터를 이용하여 사용자** ID **분류 수행**
- Input sensor
  - GPS trajectory as a sequence (x, y, t)
- Dataset and feature map
  - Training/Test data
    - Data from 50/1,000 drivers (original data: Kaggle 2015 competition on Driver Telematics Analysis)
  - Data transformation
    - Feature map (35=5 x 7)
      - Basic features (5)
        :Speed norm; difference of speed norm; acceleration norm; difference of acceleration norm; angular speed
      - Statistical features (7)
        : mean; min; max; ¼; ½; ¾; standard deviation of each basic feature

Machine Learning

- Machine Learning Models
  - CNN
    - Six layers model (3 conv. + 3 FC)
  - RNN
    - RNN model
  - Gradient boosting decision tree (GBDT)
    - For reference

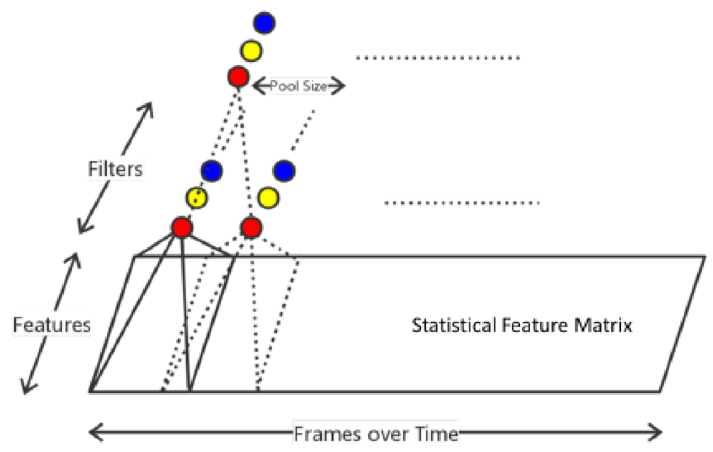


Figure 3: 1-D convolution and pooling in CNN

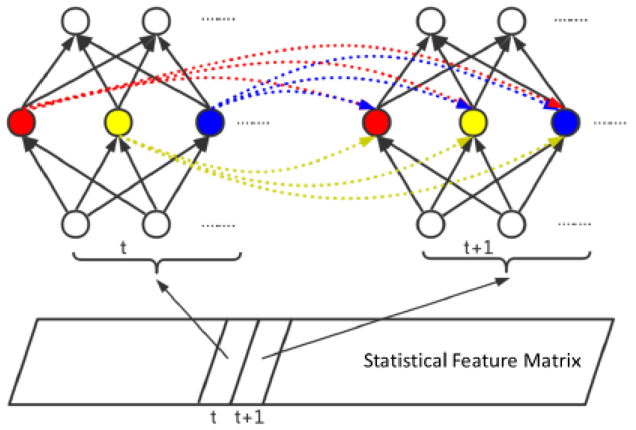


Figure 4: RNN training on the input statistical feature matrix

Experimental Results

- Experiments
  - 80% for training, 20% for testing
  - Small scale test for 50 drivers data
    - Include 35,000 segments from 8,000 trips
  - large scale test for 1,000 drivers data

Table 1: Results on 50 drivers' data

Method	Seg (%)	Trip (%)	Trip Top-5 (%)
NoPoolCNN	16.9	28.3	56.7
CNN	21.6	34.9	63.7
PretrainIRNN	28.2	44.6	70.4
IRNN	34.7	49.7	76.9
StackedIRNN	**34.8**	**52.3**	**77.4**
GBDT	18.3	29.1	55.9
TripGBDT	-	51.2	74.3

Table 2: Results on 1000 drivers' data

Method	Seg (%)	Trip (%)	Trip Top-5 (%)
CNN	23.4	26.7	46.7
StackedIRNN	**27.5**	**40.5**	**60.4**
TripGBDT	-	9.2	15.8

DETECTING ROAD SURFACE WETNESS

Methods					Tasks					
					ADAS					
					Self Driving					
					Localization	Perception	Planning/ Control	Driver state	Vehicle Diagnosis	Smart factory
Methods	Traditional	Non-machine Learning			GPS, SLAM		Optimal control			
Methods	Traditional	Machine-Learning based method	Supervised	SVM MLP		Pedestrian detection (HOG+SVM)				
Methods	Deep-Learning based	Machine-Learning based method	Supervised	CNN		Detection/ Segmentation/Classification	End-to-end Learning			
Methods	Deep-Learning based	Machine-Learning based method	Supervised	RNN (LSTM)		Dry/wet road classification	End-to-end Learning	Behavior Prediction/ Driver identification		*
Methods	Deep-Learning based	Machine-Learning based method	Supervised	DNN					*	*
Methods	Deep-Learning based	Machine-Learning based method	Reinforcement				*			
Methods	Deep-Learning based	Machine-Learning based method	Unsupervised							*

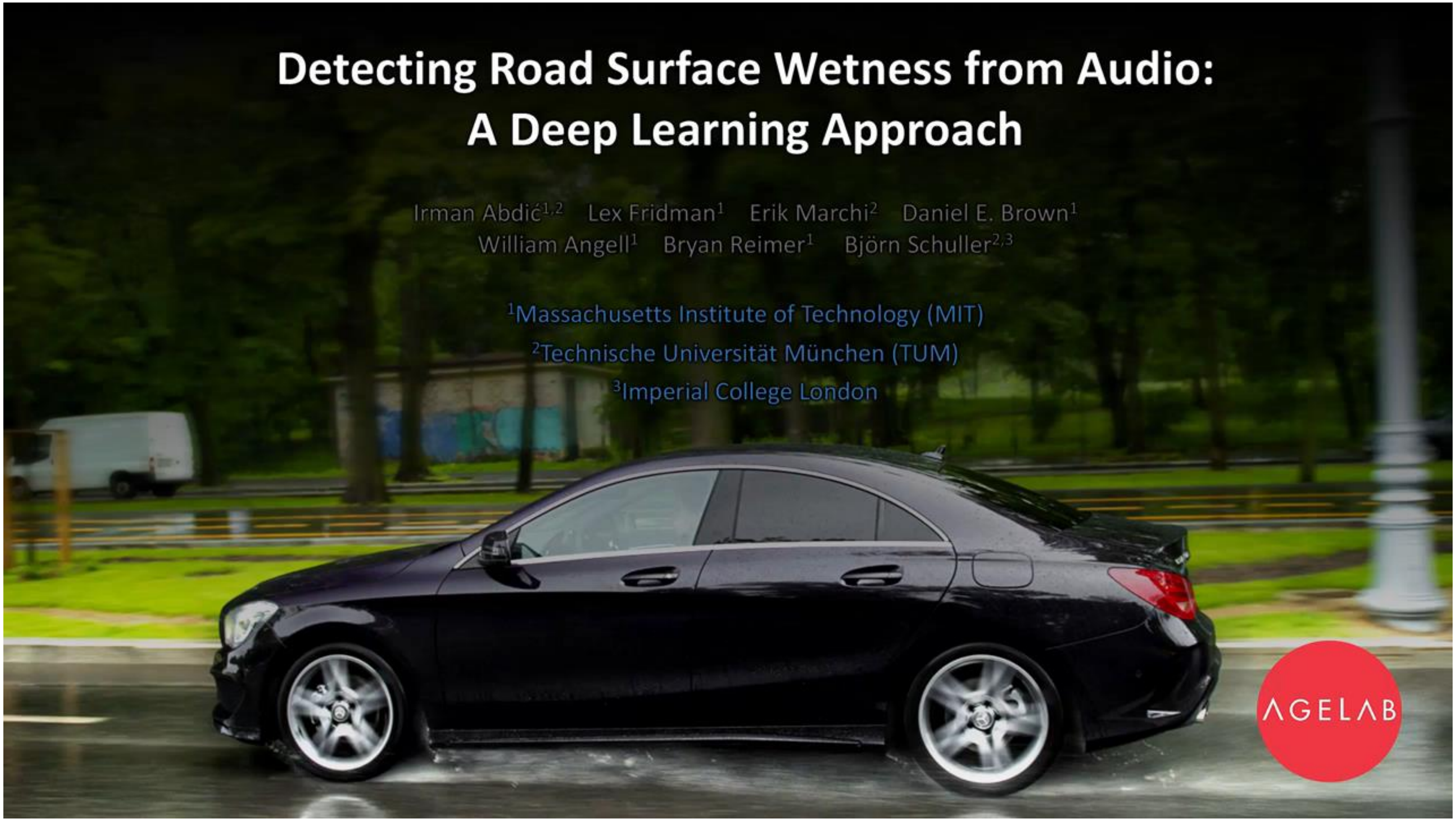

[CH:1] 2017-07-10 12:32:57
13.84V
BRAVO v20141229

Dataset/Machine Learning

- Shotgun microphone behind the rear tire
- Spectogram + LSTM

- 관련 기사
  - http://www.ibtimes.co.uk/heres-how-self-driving-cars-can-detect-dangerous-roads-using-sound-ai-1532407

Detecting Road Surface wetness



END-TO-END DRIVING WITH RNNS

<table>
<tr><td colspan="5" rowspan="4"></td><th colspan="6">Tasks</th></tr>
<tr><th colspan="5">ADAS</th><th></th></tr>
<tr><th colspan="3">Self Driving</th><th></th><th></th><th></th></tr>
<tr><th>Localization</th><th>Perception</th><th>Planning/ Control</th><th>Driver state</th><th>Vehicle Diagnosis</th><th>Smart factory</th></tr>
<tr><td rowspan="7">Methods</td><td rowspan="2">Traditional</td><td colspan="3">Non-machine Learning</td><td>GPS, SLAM</td><td></td><td>Optimal control</td><td></td><td></td><td></td></tr>
<tr><td rowspan="6">Machine-Learning based method</td><td rowspan="4">Supervised</td><td>SVM
MLP</td><td></td><td>Pedestrian detection (HOG+SVM)</td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="5">Deep-Learning based</td><td>CNN</td><td></td><td>Detection/ Segmentation/Classification</td><td>End-to-end Learning</td><td></td><td></td><td></td></tr>
<tr><td>RNN (LSTM)</td><td></td><td>Dry/wet road classification</td><td>End-to-end Learning</td><td>Behavior Prediction/ Driver identification</td><td></td><td>*</td></tr>
<tr><td>DNN</td><td></td><td></td><td></td><td></td><td>*</td><td>*</td></tr>
<tr><td colspan="2">Reinforcement</td><td></td><td></td><td>*</td><td></td><td></td><td></td></tr>
<tr><td colspan="2">Unsupervised</td><td></td><td></td><td></td><td></td><td></td><td>*</td></tr>
</table>

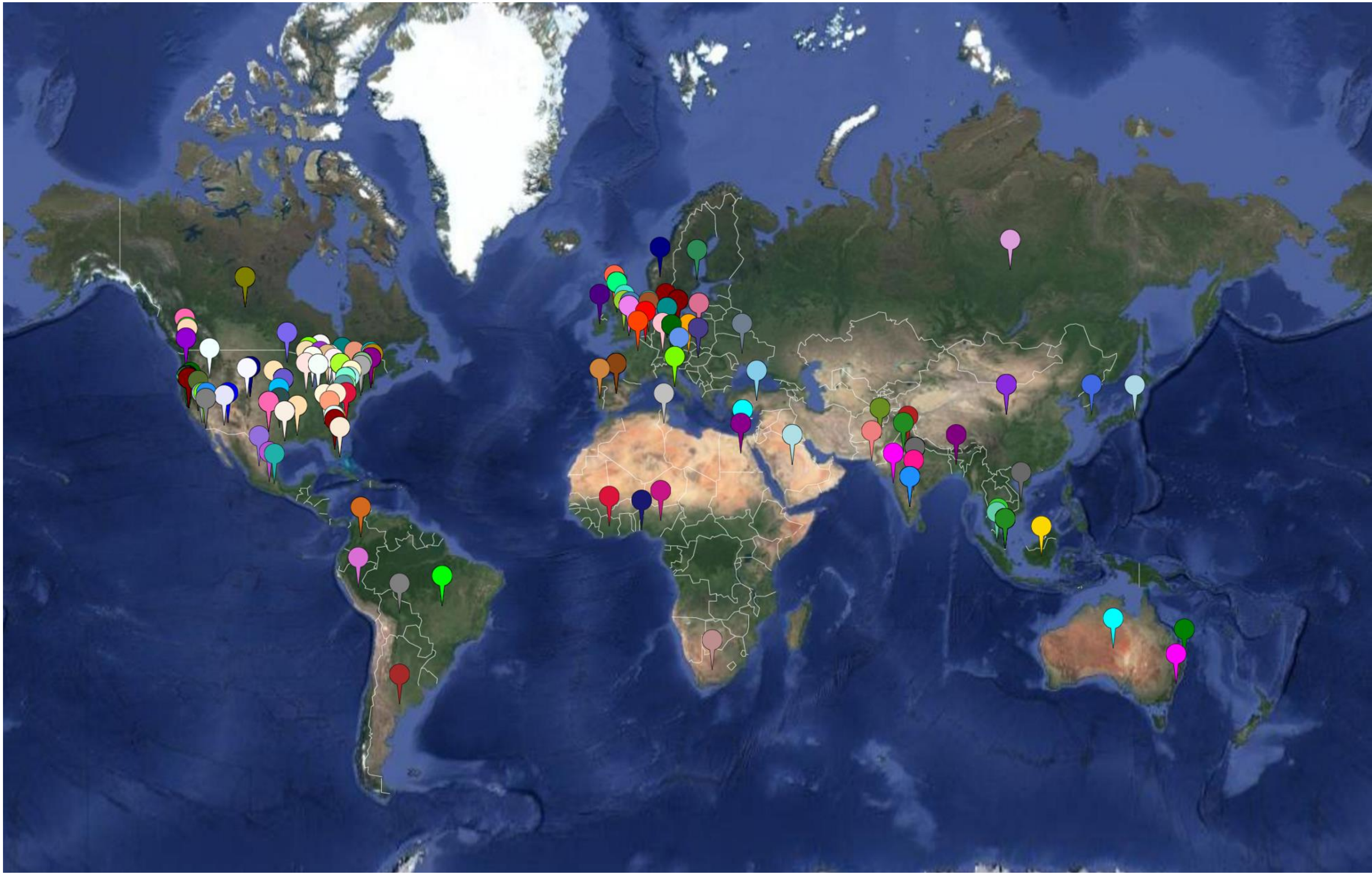
End-to-end Driving with RNNs

- 1st and 3rd place winner of the Udacity end-to-end steering competition used RNNs:
  - Sequence-to-sequence mapping from images to steering angles

- **관련 링크**
  - https://medium.com/udacity/challenge-2-using-deep-learning-to-predict-steering-angles-f42004a36ff3
  - https://medium.com/udacity/teaching-a-machine-to-steer-a-car-d73217f2492c

UDACITY

Self-Driving Car Engineer Nanodegree



1st Place Winner

- LSTM inputs: 3D convolution of image sequence
- Outputs: predicted steering angle, speed, torque
- Sequence length = 10

3rd Place Winner

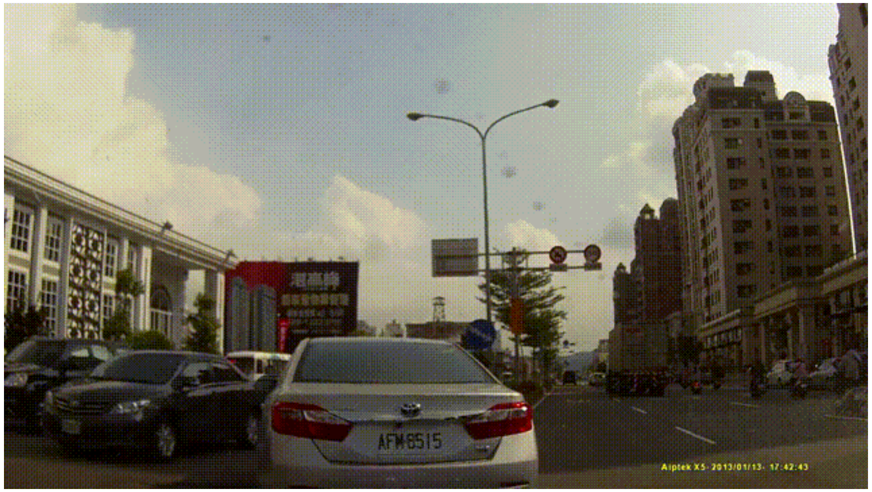
- LSTM inputs: 3000 features extracted with CNN
- Outputs: predicted steering angle
- Sequence length = 10

ANTICIPATING ACCIDENTS IN DASHCAM VIDEOS

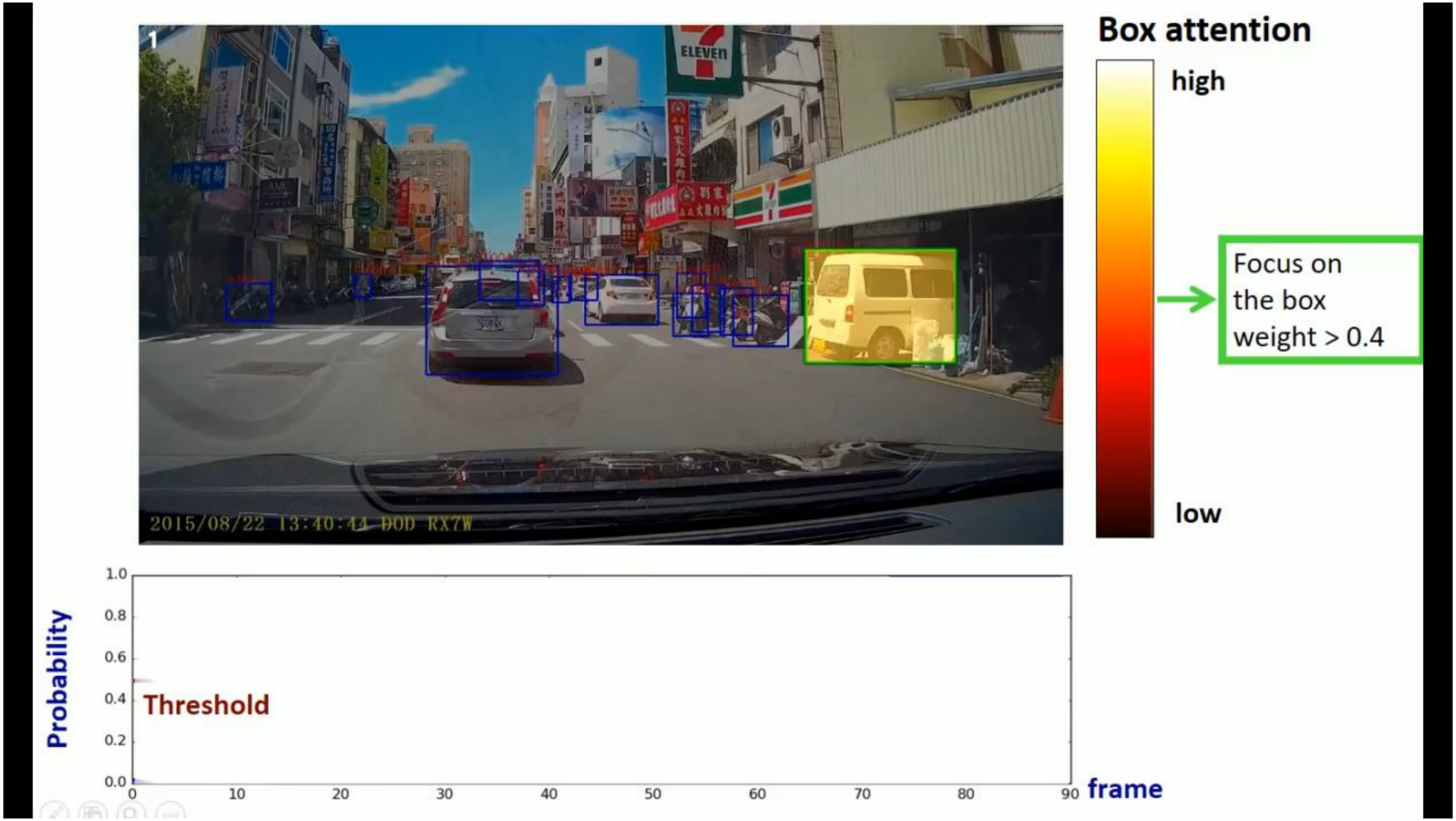
Dataset

	Positive examples	Negative examples	Total
Training set	455	829	1284
Testing set	165	301	466
Total	620	1130	1730

Positive example



Negative example



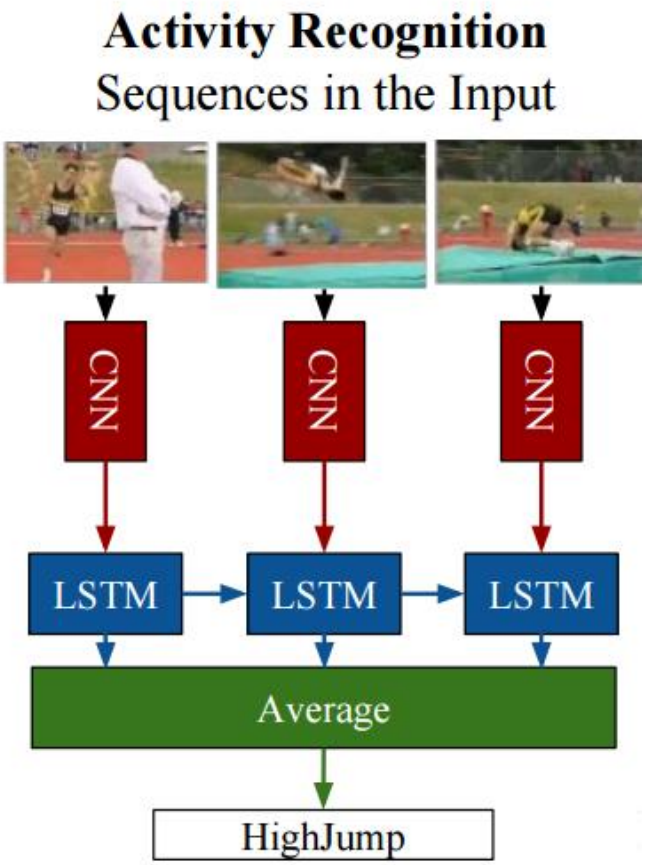
1
2015/08/22 13:40:44 DOD RX7W
Box attention
high
low
Focus on the box weight > 0.4
Probability
Threshold
frame
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
70
80
90

BACKUPS

LONG–TERM RECURRENT CONVOLUTIONAL NETWORKS FOR VISUAL RECOGNITION AND DESCRIPTION

Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, Trevor Darrell

Long-time Spatio-Temporal ConvNets

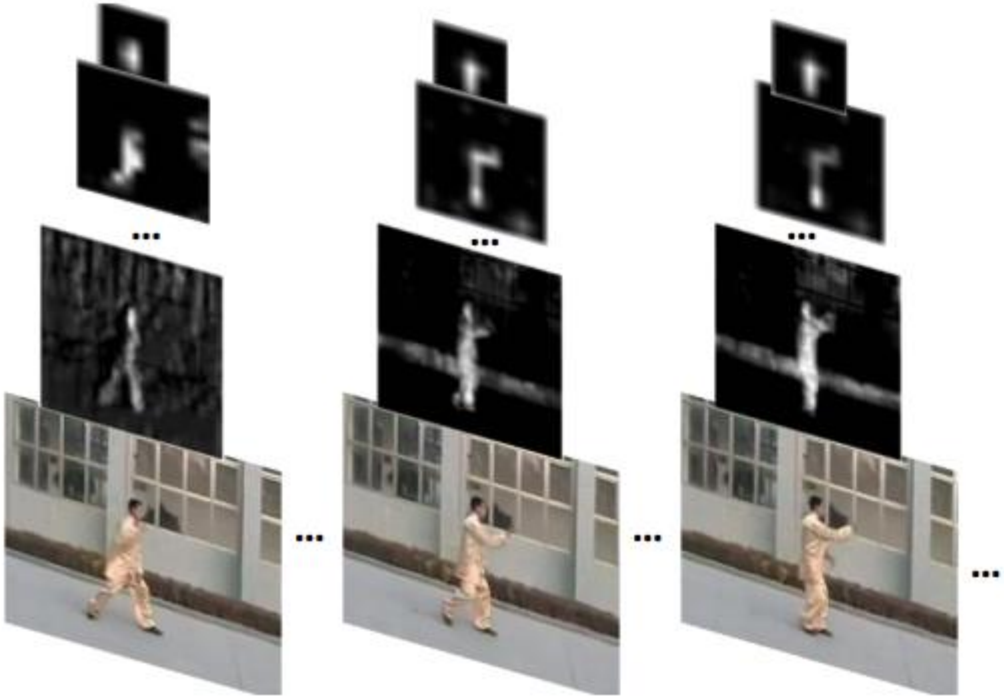


DELVING DEEPER INTO CONVOLUTIONAL NETWORKS FOR LEARNING VIDEO REPRESENTATIONS

Ballas et al., 2016

Limitations in simple RNN-ConvNet

- Visualization of convolutional maps on successive frames in video. As we go up in the CNN hierarchy, we observe that the convolutional maps are more stable over time, and thus discard variation



Stack-GRU-RCN

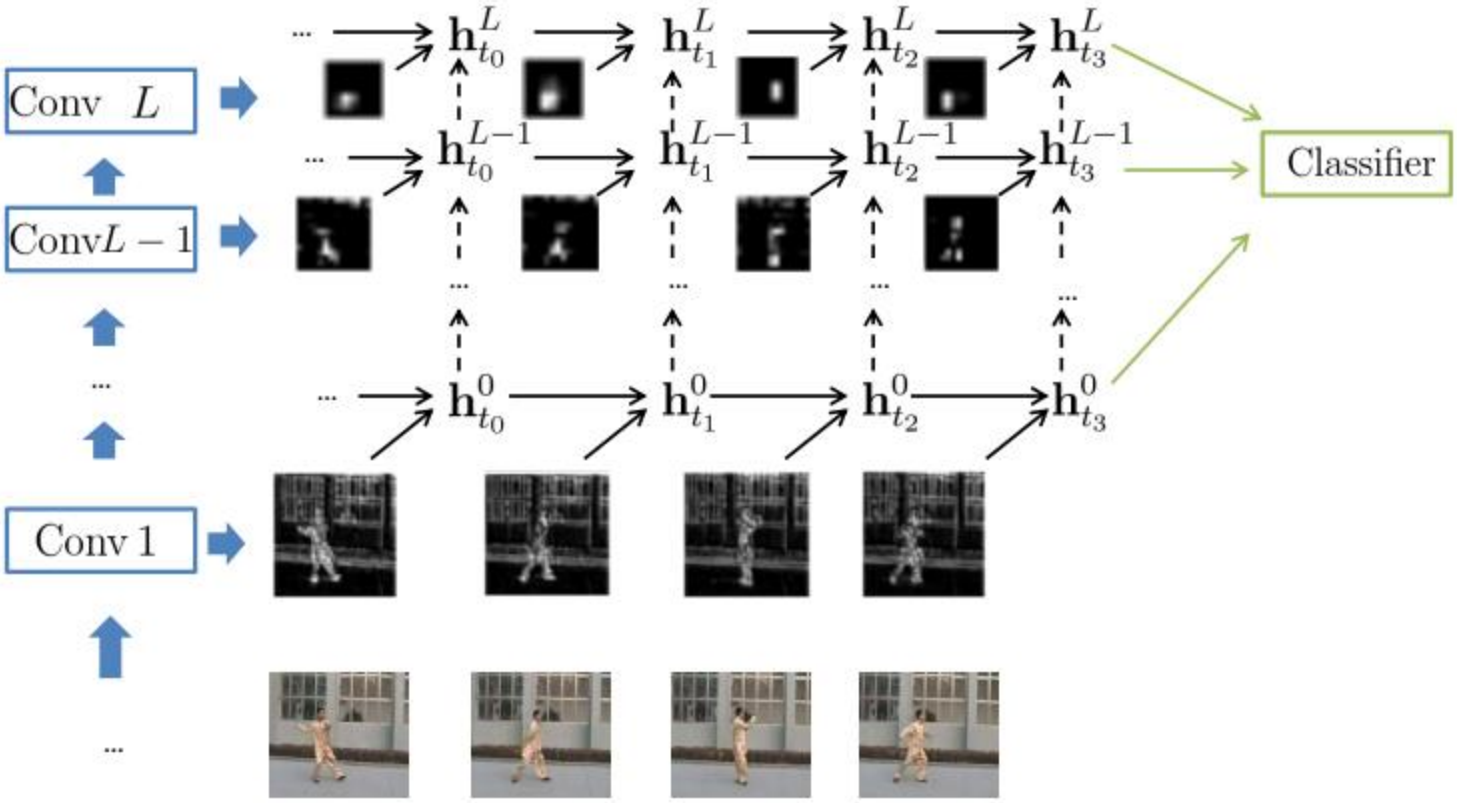


Figure 2: High-level visualization of our model. Our approach leverages convolutional maps from different layers of a pretrained-convnet. Each map is given as input to a convolutional GRU-RNN (hence GRU-RCN) at different time-step. Bottom-up connections may be optionally added between RCN layers to form Stack-GRU-RCN.

RCN (Recurrent Convolution Networks)