Deep Learning And Robotics
- An Introduction

Liang Yang

2017. 06. 07

Outline

- The Knowledge Structure of Deep Learning
- History - Now and Past
- Robotics - What can we do, and Why?
- My Research

Outline

- The Knowledge Structure of Deep Learning
- History - Now and Past
- Robotics - What can we do, and Why?
- My Research

The Knowledge Structure of Deep Learning

Texas Hold'em poker

AI

NN

Deep Learning in Medicine

Deep learning

NLP

Generative adversarial networks

AlphaGo

You may heard all these topics

Face Detection

Computer Vision

Brain Chip

Assistant robot

Intelligent Translation

Autonomous Driving

CNN

Adversarial learning

S/US learning

LSTM

Semantics

RL

Recommender system

Basic Idea And Discussion

Traditional Approach

Pre-processing → Feature detection → Choose Classifier

Normalization

De-noise

Reduce D

Artificial Feature:

CV: ORB, LBP, Fisher, Hog, SIFT

LP:MFCC word2vec

SVM

decision tree

bayes network

clustering

linear regression

No need anymore??

Deep Learning

Pre-processing → Model design → Training

Data searching

labeling

CNN, RNN, CNN+RNN

Try: Structure, loss function, other parameters

Much better performance...

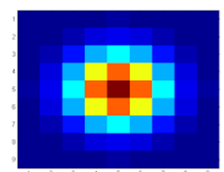
Basic Idea And Discussion

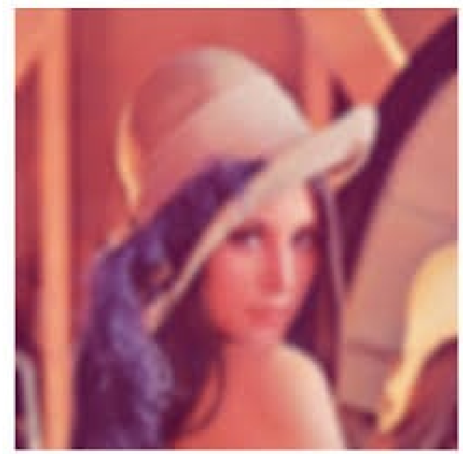
Neural Network → Convolutional Neural Network

1/256 x

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

$*$





Convolutional kernel (mostly initialized with gaussian distribution): a filter?

Basic Idea And Discussion

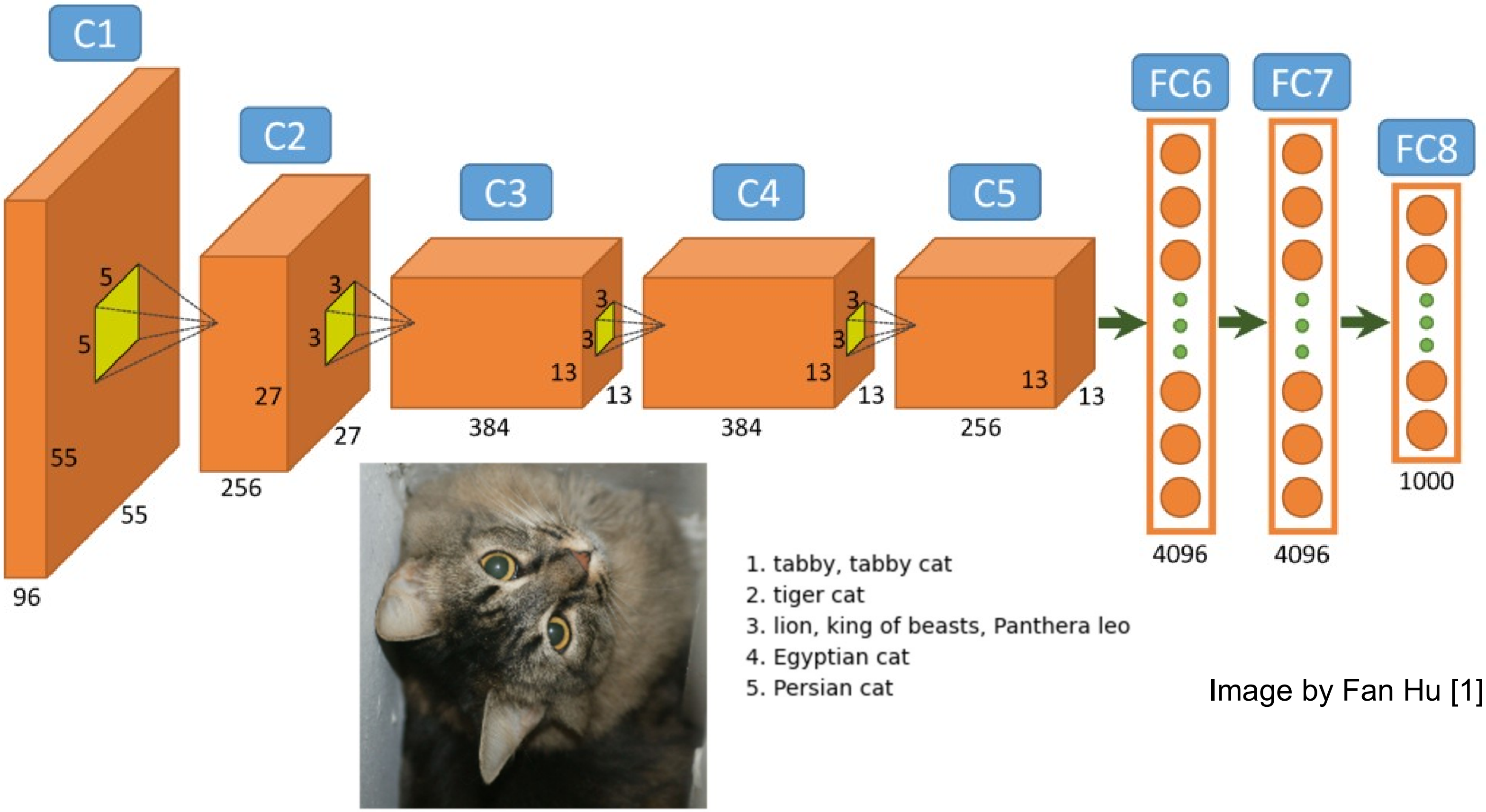


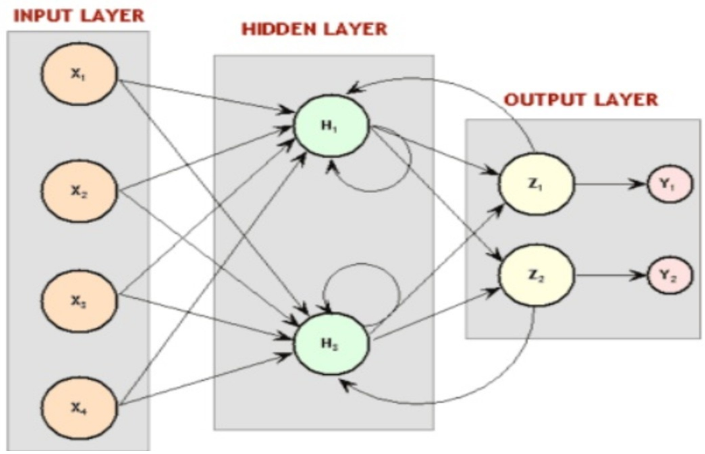
Image by Fan Hu [1]

Hu, Fan, Gui-Song Xia, Jingwen Hu, and Liangpei Zhang. "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery." Remote Sensing 7, no. 11 (2015): 14680-14707.

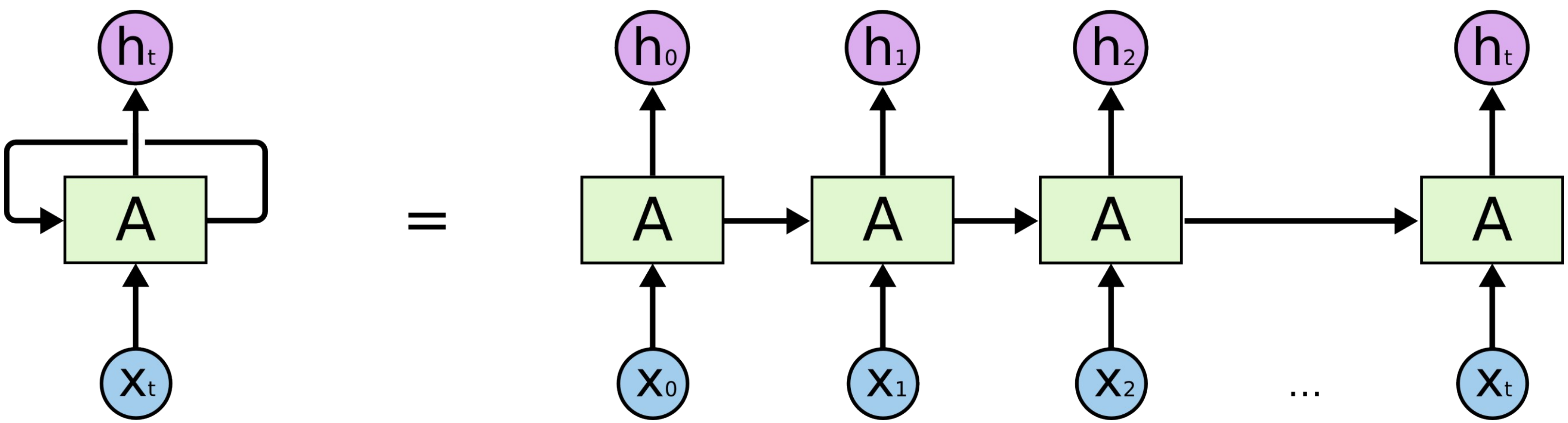
Basic Idea And Discussion

Recurrent Neural Networks (RNN):

Remenber last step information: real deep in both spacial and time domain



$$h_t = \sigma(W^{hh} h_{t-1} + W^{hx} x_t)$$

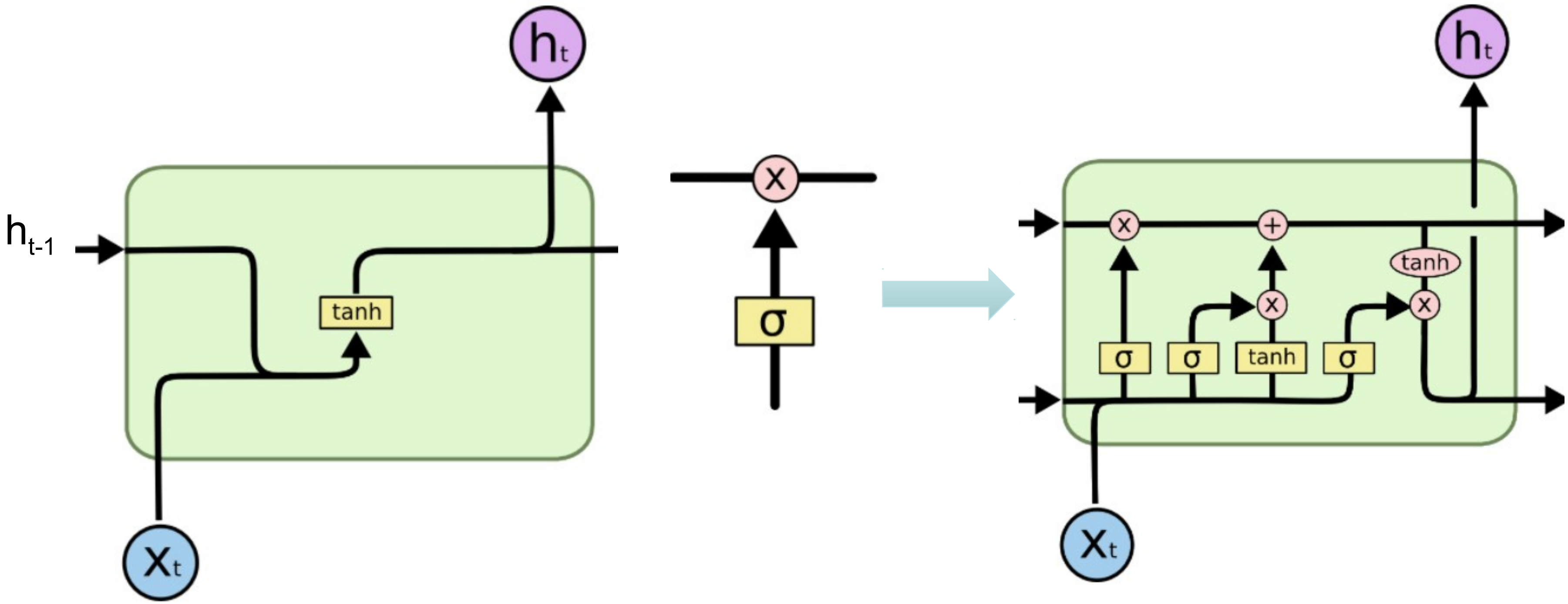


Basic Idea And Discussion

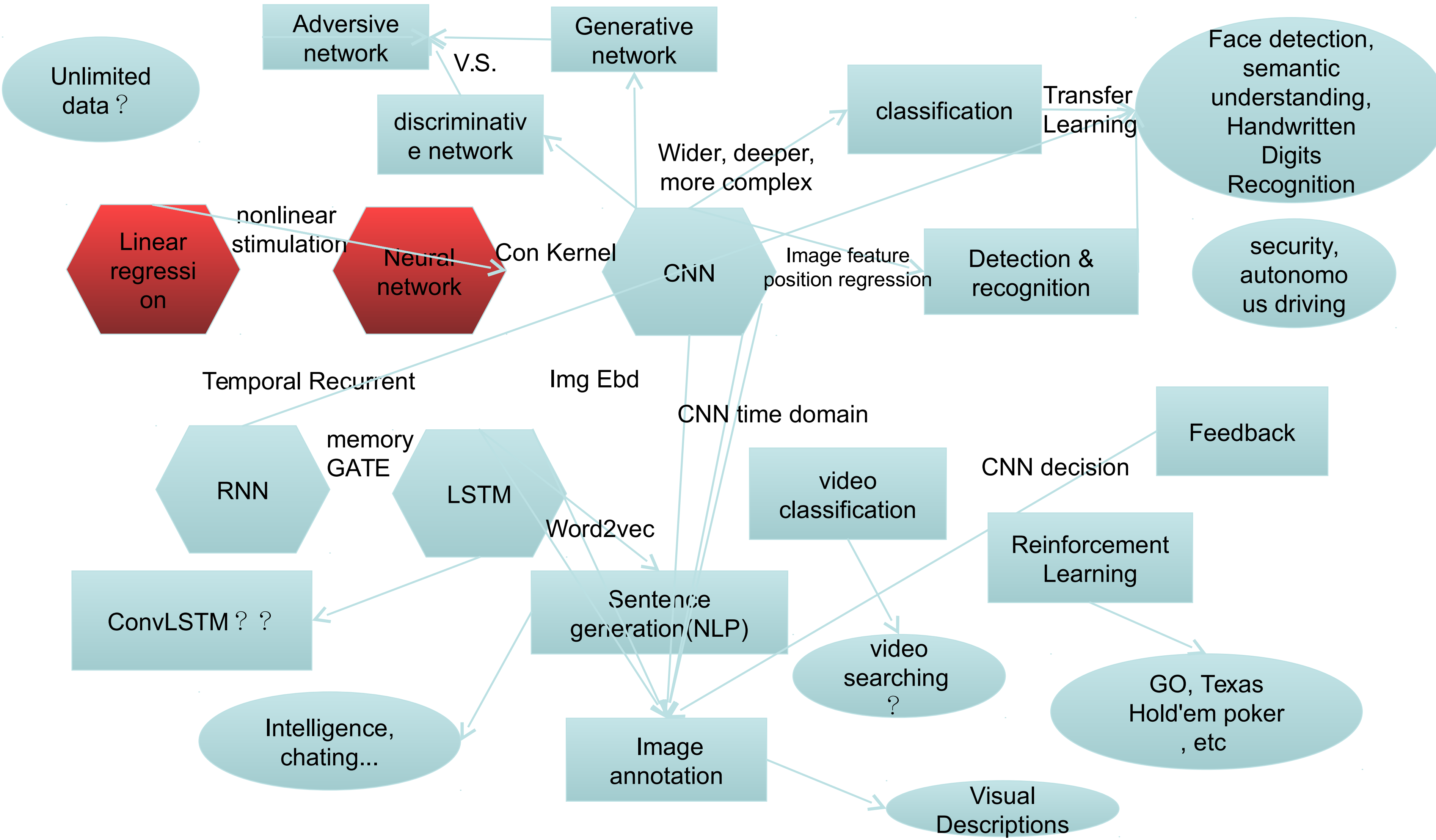
Long Short-term Memory (LSTM):

Not only one state from past, but many

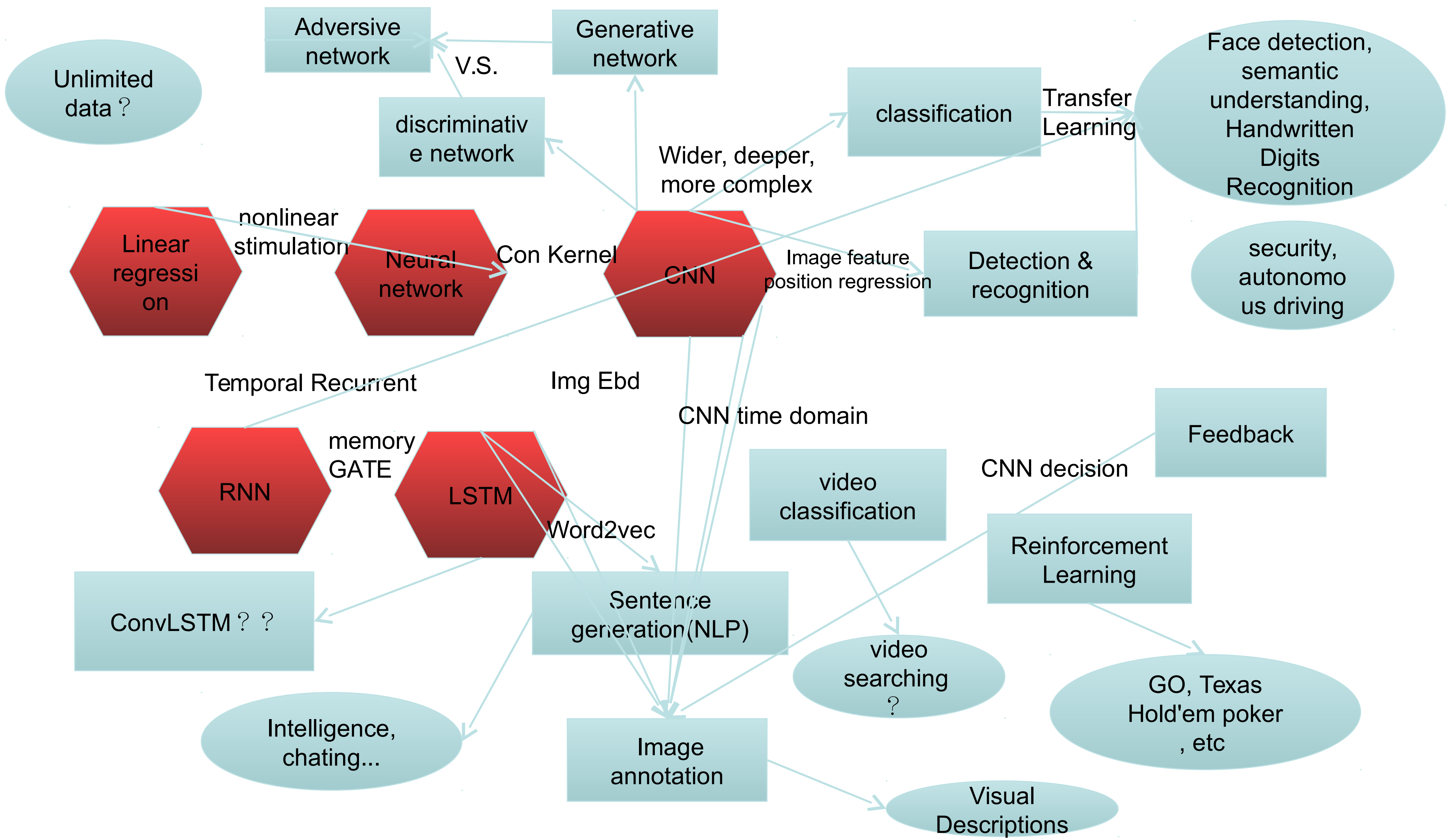
Forget gate, input gate, cell state



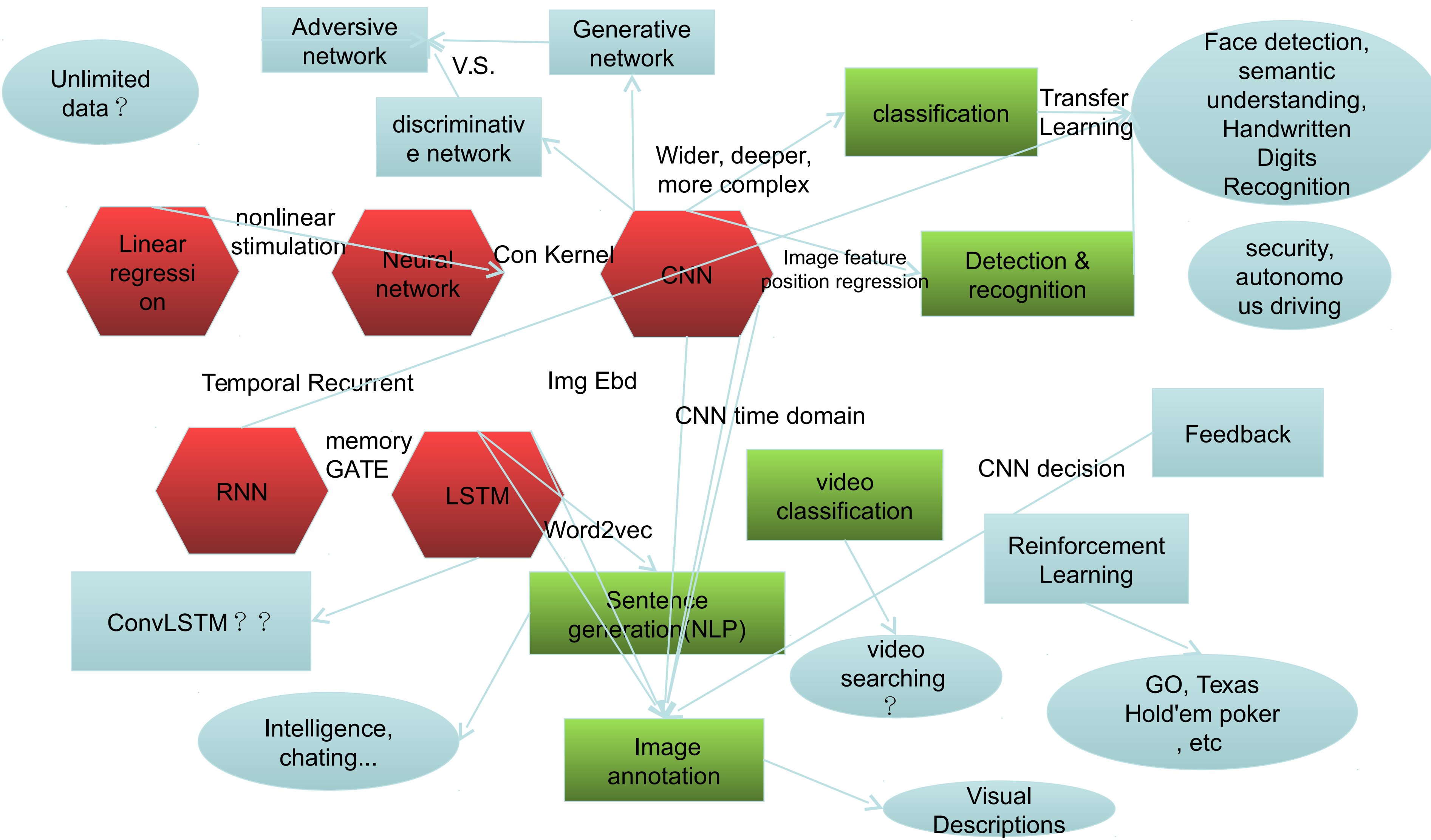
The Knowledge Structure of Deep Learning



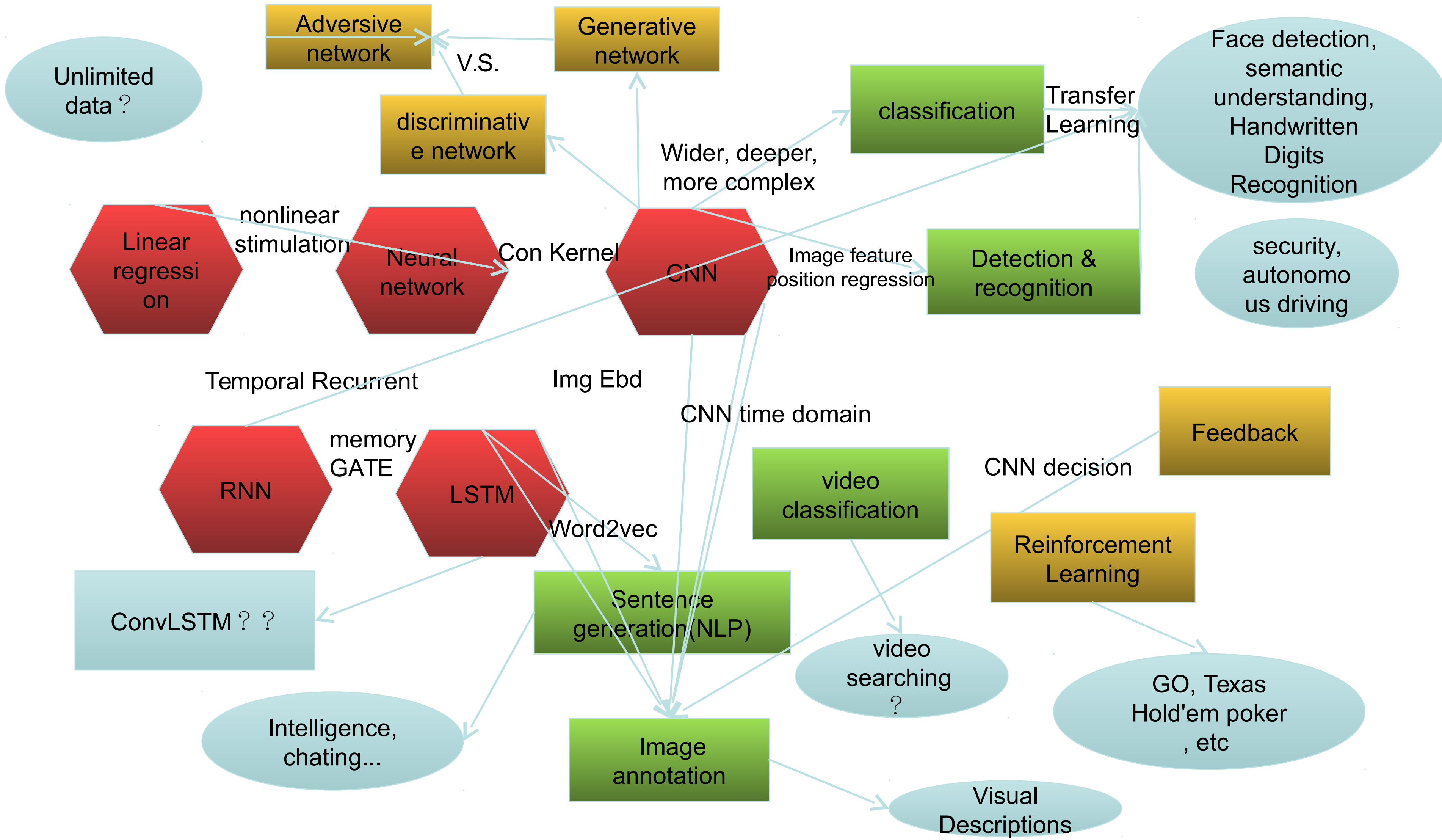
The Knowledge Structure of Deep Learning



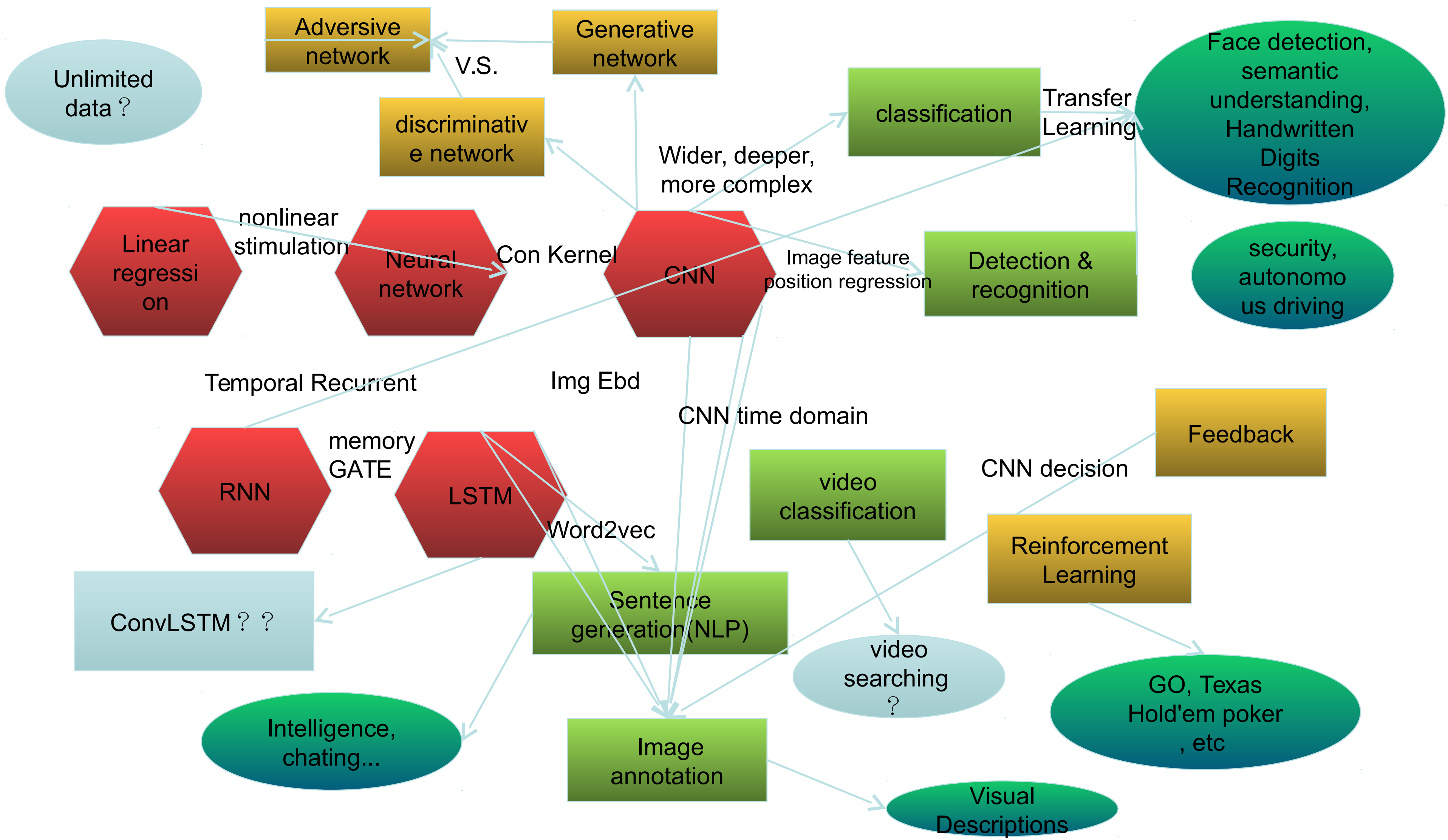
The Knowledge Structure of Deep Learning



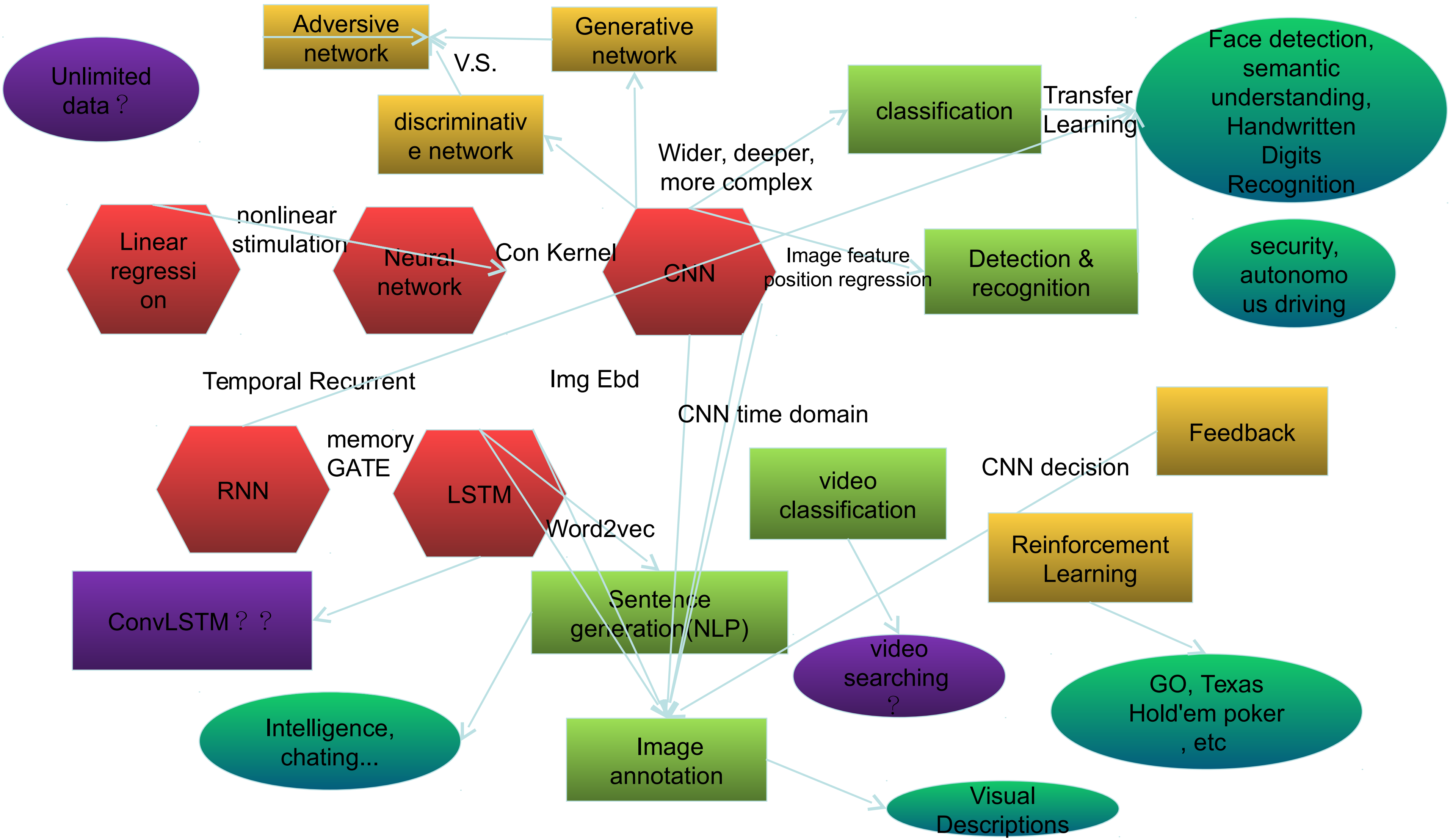
The Knowledge Structure of Deep Learning



The Knowledge Structure of Deep Learning



The Knowledge Structure of Deep Learning



Outline

- The Knowledge Structure of Deep Learning
- History - Now and Past
- Robotics - What can we do, and Why?
- My Research

History - Now and Past

- 1990s ，Yann Lecun CNN

  'Gradient-based learning applied to document recognition'
- 2009 ，ImageNet data set published

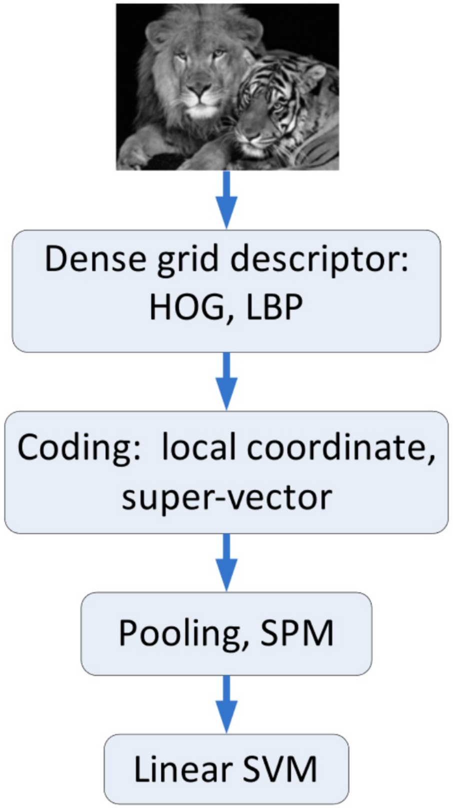
  'Imagenet: A large-scale hierarchical image database'
- 2012 ，AlexNet: Proposed to use GPU for training, first place for ImageNet classification

  'ImageNet Classification with Deep Convolutional Neural Networks'
- 2016 ，Google DeepMind:Alpha Go

  You already know, at least heard something about it...

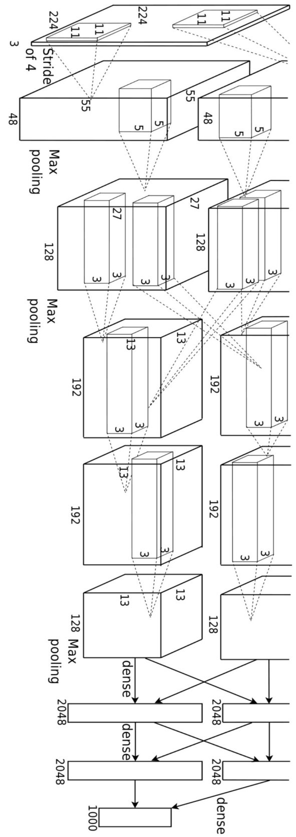
History - Now and Past

Year 2011
NEC-UIUC



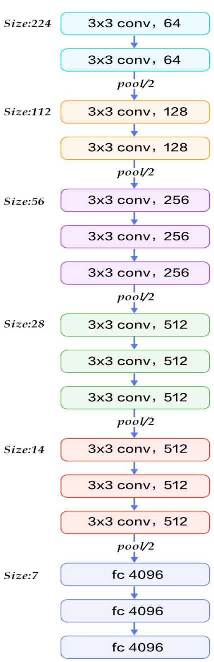
Lin CVPR 2011

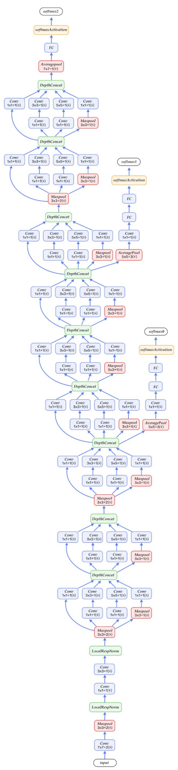
Year 2012
Super vision



krizhevsky NIPS 2012

Year 2014
VGG GoogleNet

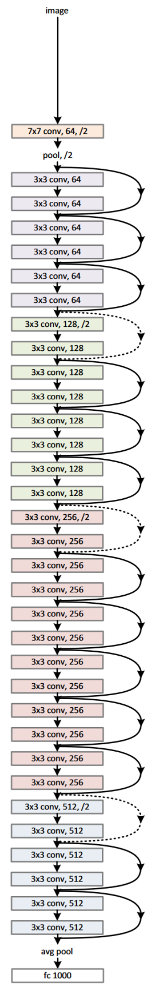




Simonyan arxiv 2014

Szegedy arxiv 2014

Year 2015
MSRA- DR



History - Now and Past

框架	语言	文档资料	CNN兼容	RNN兼容	上手难易	速度	并行支持	Keras兼容	支持团队
Theano	Python/ C++	++	++	++	+	++	+	+	蒙特利尔大学
Tensor Flow	Python	+++	+++	++	+++	++	++	+	Google
Torch	Lua, Python	+	+++	++	++	+++	++		Facebook
Caffe	C++	+	++		+	+	+		贾扬清 加州伯克利
MXNet	Python, R, Julia	++	++	+	++	++	+++	+?	李沐, Amazon
Neon	Python	+	++	+	+	++	+		Intel
CNTK	C++	+	++	+++	+	++	+		Microsoft

From [机器之心]

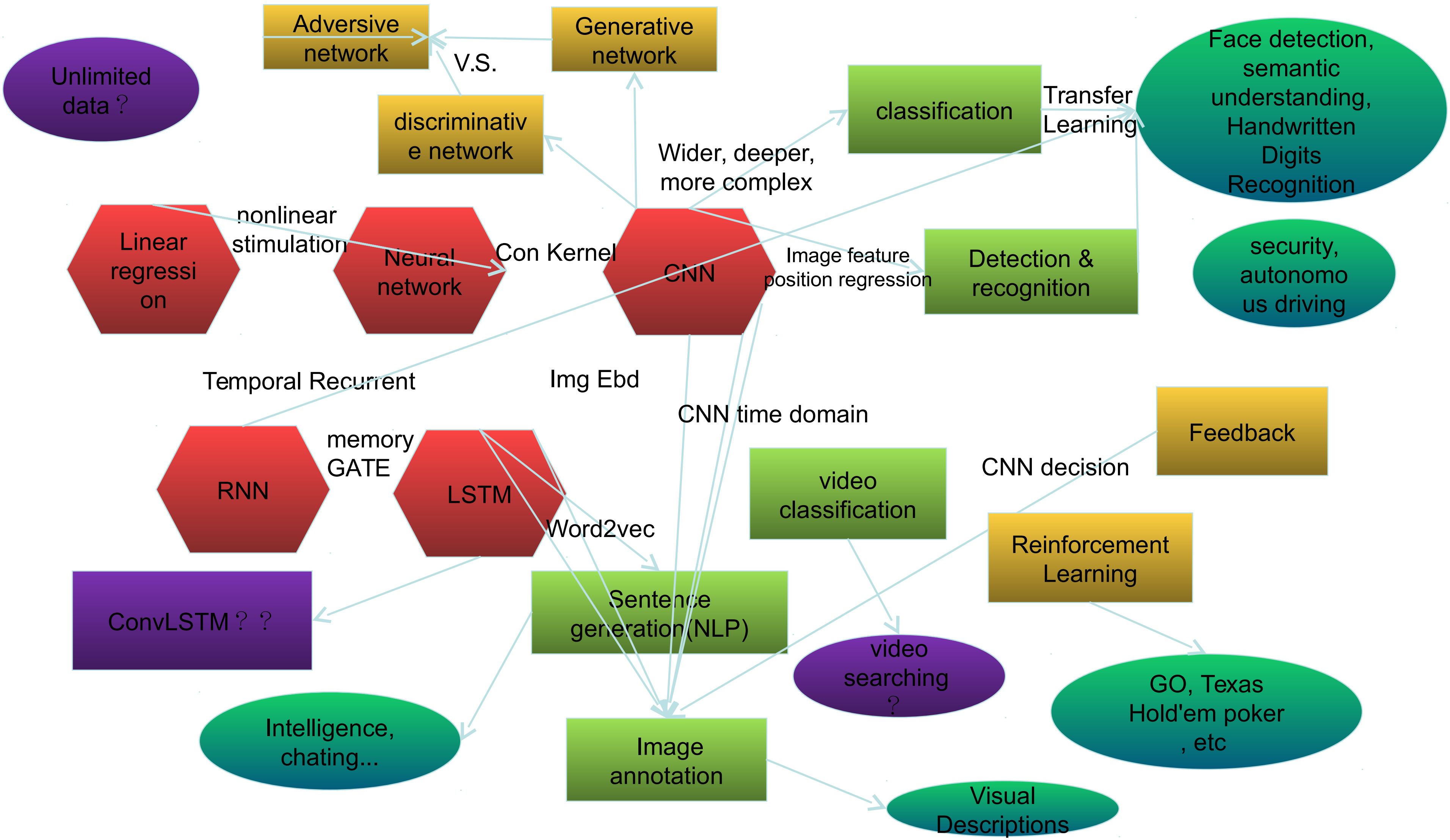
Outline

- The Knowledge Structure of Deep Learning
- History - Now and Past
- Robotics - What can we do, and Why?
- My Research

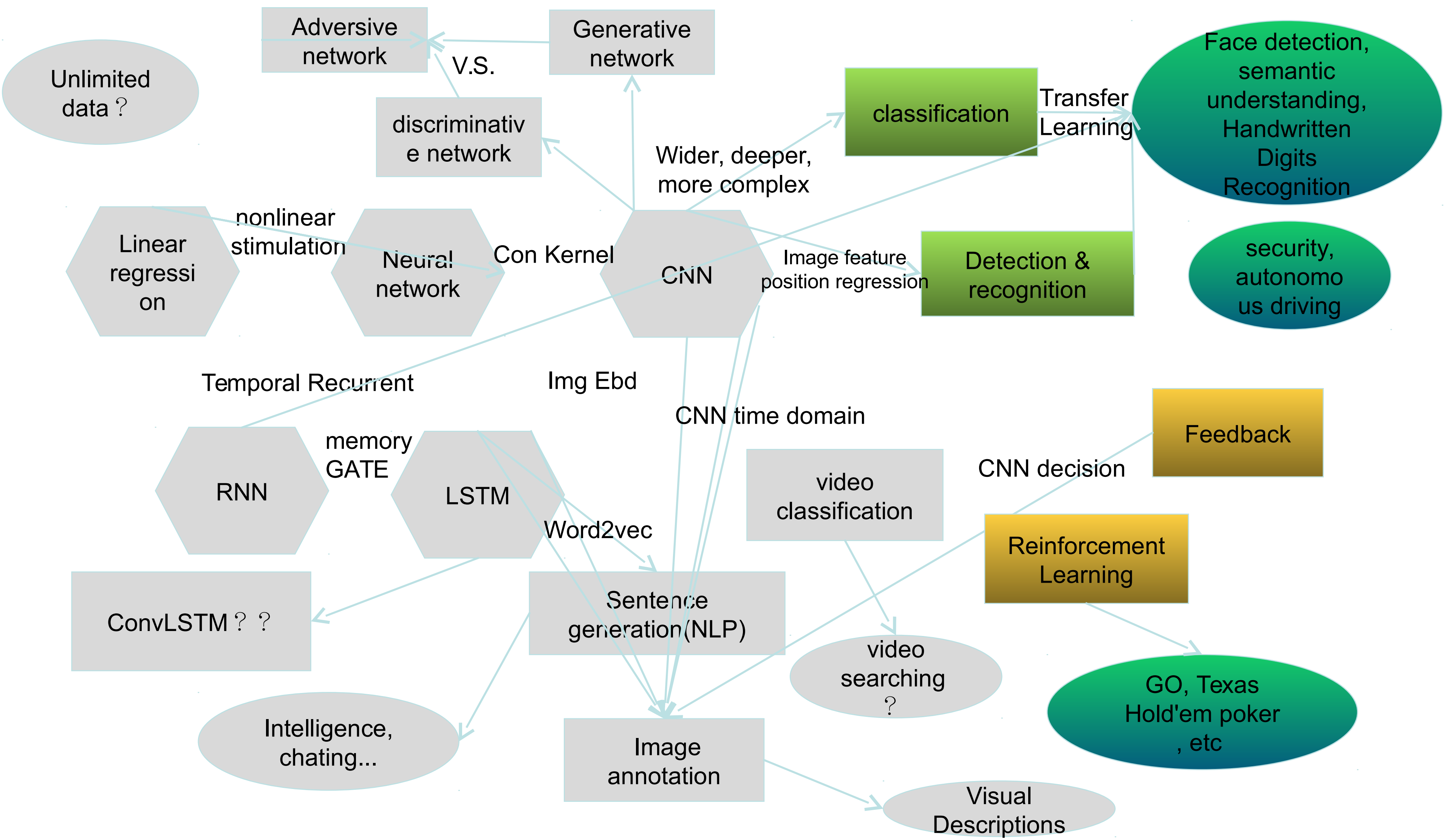
Robotics - What can we do, and Why?

- Control
- Tracking
- computer vision
- Autonomous driving
- Path planing
- Obstacle detection
- 3D modeling
- Markov tree
- Obstacle avoidance
- Decision
- Motion planning
- Navigation
- Structure optimization
- Swarm

Robotics - What can we do, and Why?



Robotics - What can we do, and Why?

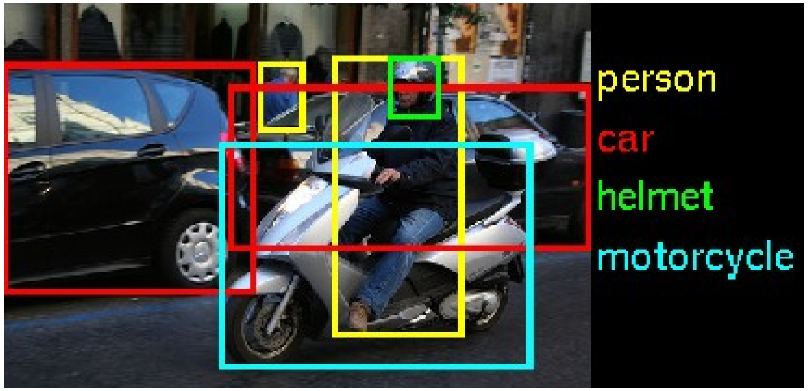


Robotics - What can we do, and Why?

Detection

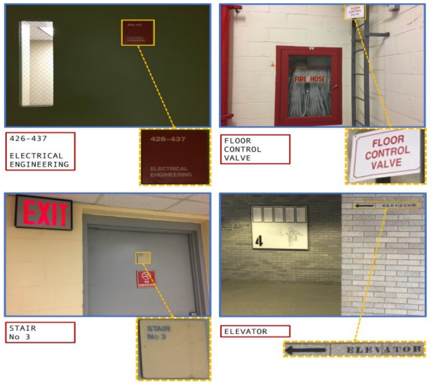
Face detection

Target detection



.
.
.

word detection



Robotics - What can we do, and Why?

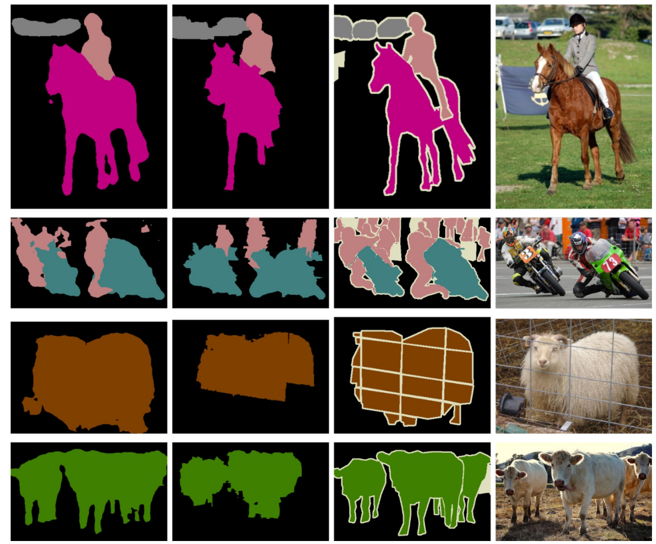
Detection and labeling

Segmentation

Pixel labeling

. . .

Obstacle detection



[1] Evan Shelhamer et. al, Fully Convolutional Networks for Semantic Segmentation
[2] Sebastian Ramos, Detecting Unexpected Obstacles for Self-Driving Cars: Fusing Deep Learning and Geometric Modeling

Robotics - What can we do, and Why?

Try new ideas

Mono depth estimation

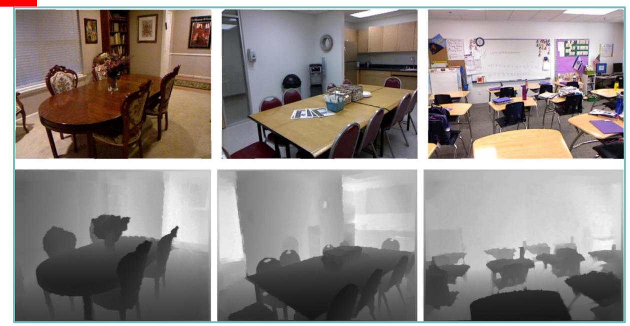
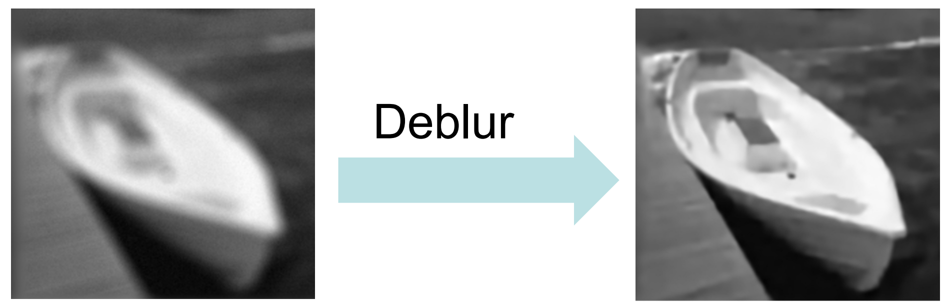
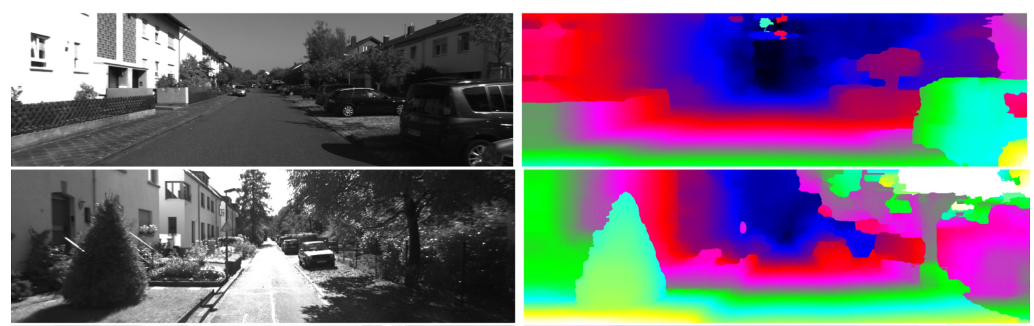


Image deblur



. . .

Stereo matching



[1] Fayao Liu, Deep Convolutional Neural Fields for Depth Estimation from a Single Image
[2] Ruomei Yan, Blind Image Blur Estimation via Deep Learning
[3] Wenjie Luo, Efficient Deep Learning for Stereo Matching

Robotics - What can we do, and Why?

Try new implementation

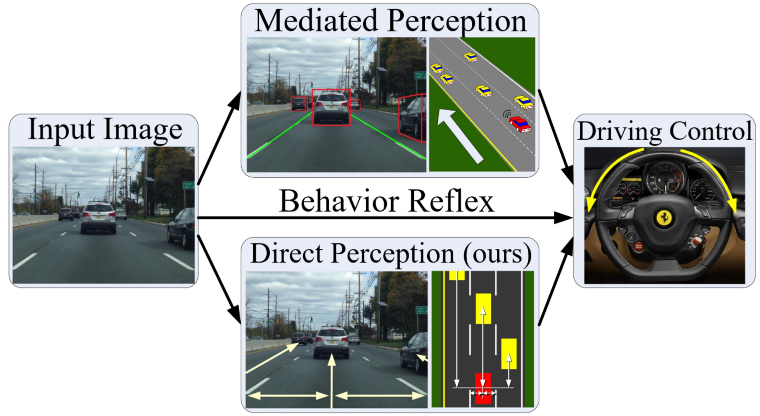
Decision

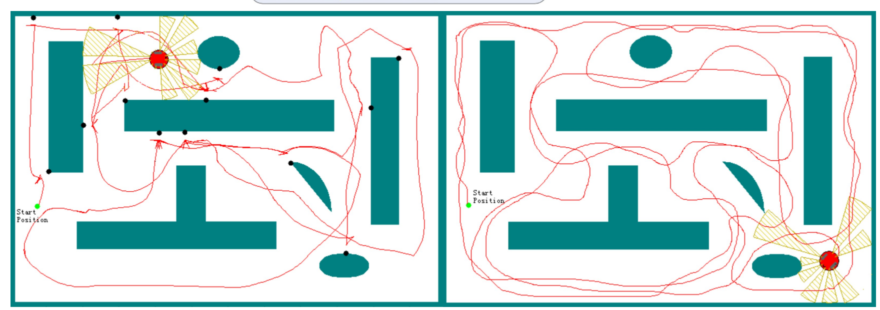
Path planning

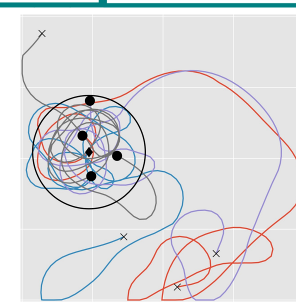
.
.
.

Obstacle avoidance

Swarm



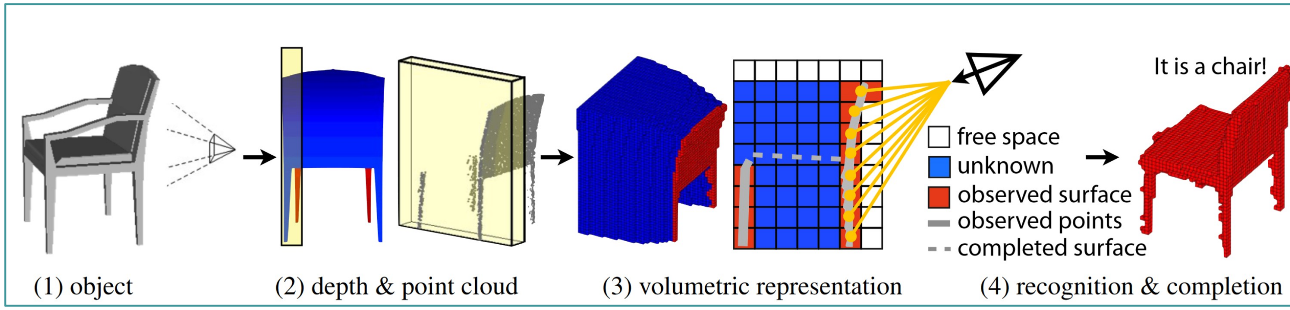


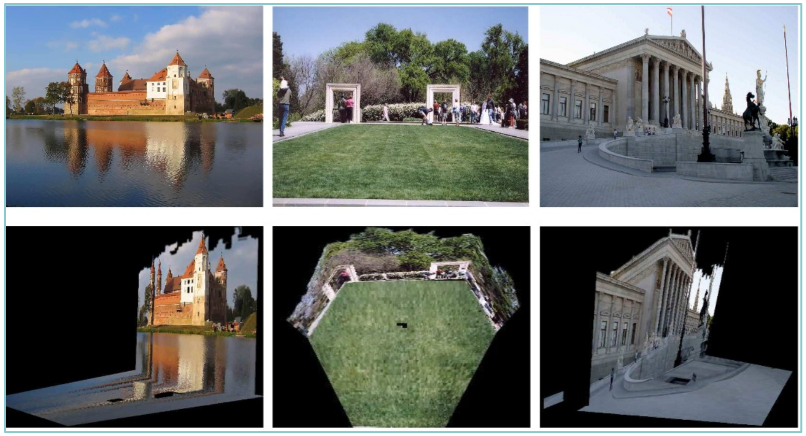


[1] Chenyi Chen, DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving
[2] Josh Beitelspacher, Applying Reinforcement Learning to Obstacle Avoidance
[3] Gerhard Neumann, Guided Deep Reinforcement Learning for Robot Swarms

Robotics - What can we do, and Why?

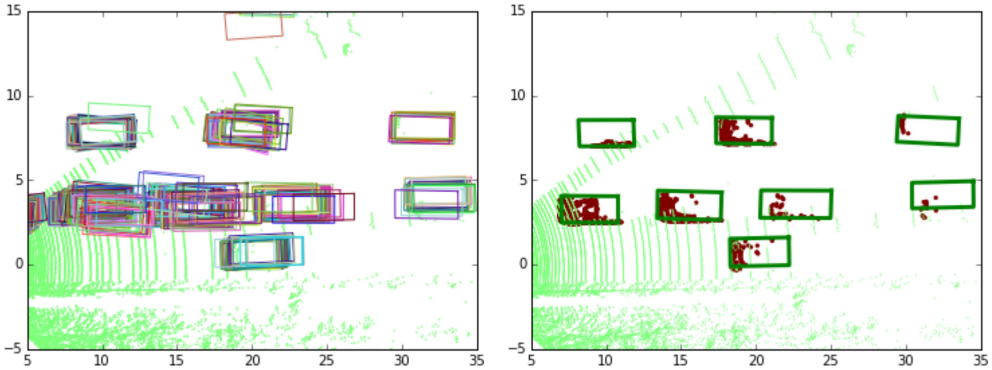
Higher dimension:





Mono reconstruction

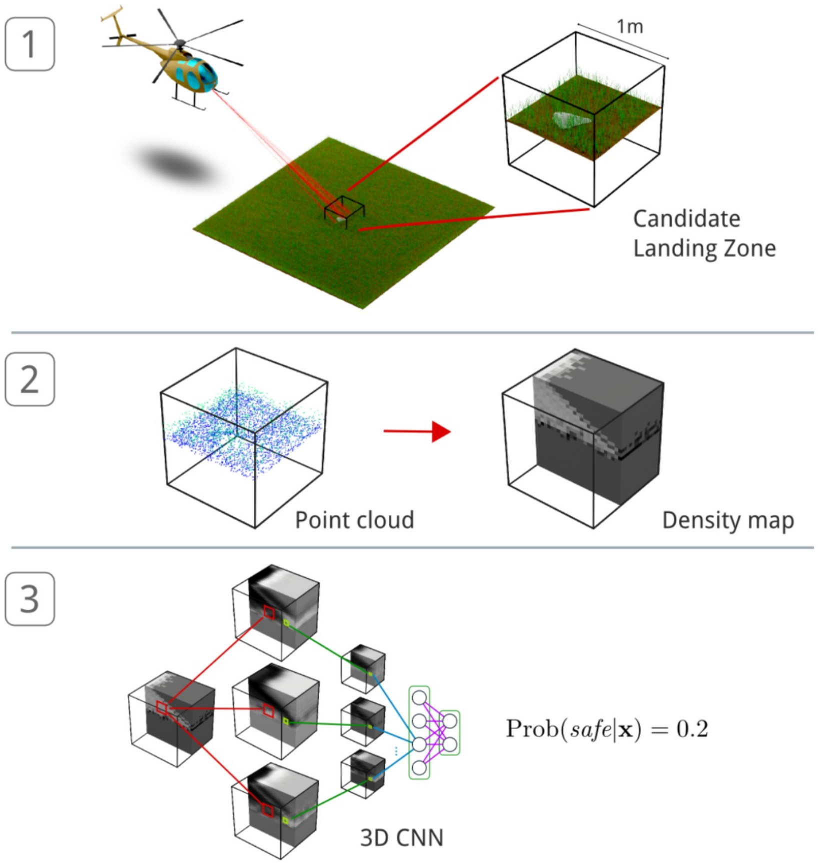
Vehicle detection based on Lidar



[1] Zhirong Wu, 3D ShapeNets: A Deep Representation for Volumetric Shapes
[2] Ashutosh Saxena, Make3D: Learning 3D Scene Structure from a Single Still Image
[3] Bo Li, Vehicle Detection from 3D Lidar Using Fully Convolutional Network

Robotics - What can we do, and Why?

Higher dimension:



[1] Zhirong Wu, 3D ShapeNets: A Deep Representation for Volumetric Shapes
[2] Ashutosh Saxena, Make3D: Learning 3D Scene Structure from a Single Still Image
[3] Bo Li, Vehicle Detection from 3D Lidar Using Fully Convolutional Network
[4] Daniel Maturana, 3D Convolutional Neural Networks for Landing Zone Detection from LiDAR

Robotics - What can we do, and Why?

Control

Tracking

computer vision

Autonomous driving

Path planing

Obstacle detection

3D modeling

Markov tree

Obstacle avoidance

Decision

Motion planning

Navigation

Structure optimization

Swarm

Outline

- The Knowledge Structure of Deep Learning
- History - Now and Past
- Robotics - What can we do, and Why?
- My Research

My Research

Concrete Structure Inspection Using Stereo-Vision UAV and Deep Network Algorithm

Liang Yang, Bing LI, Wei LI, Jizhong Xiao

My Research

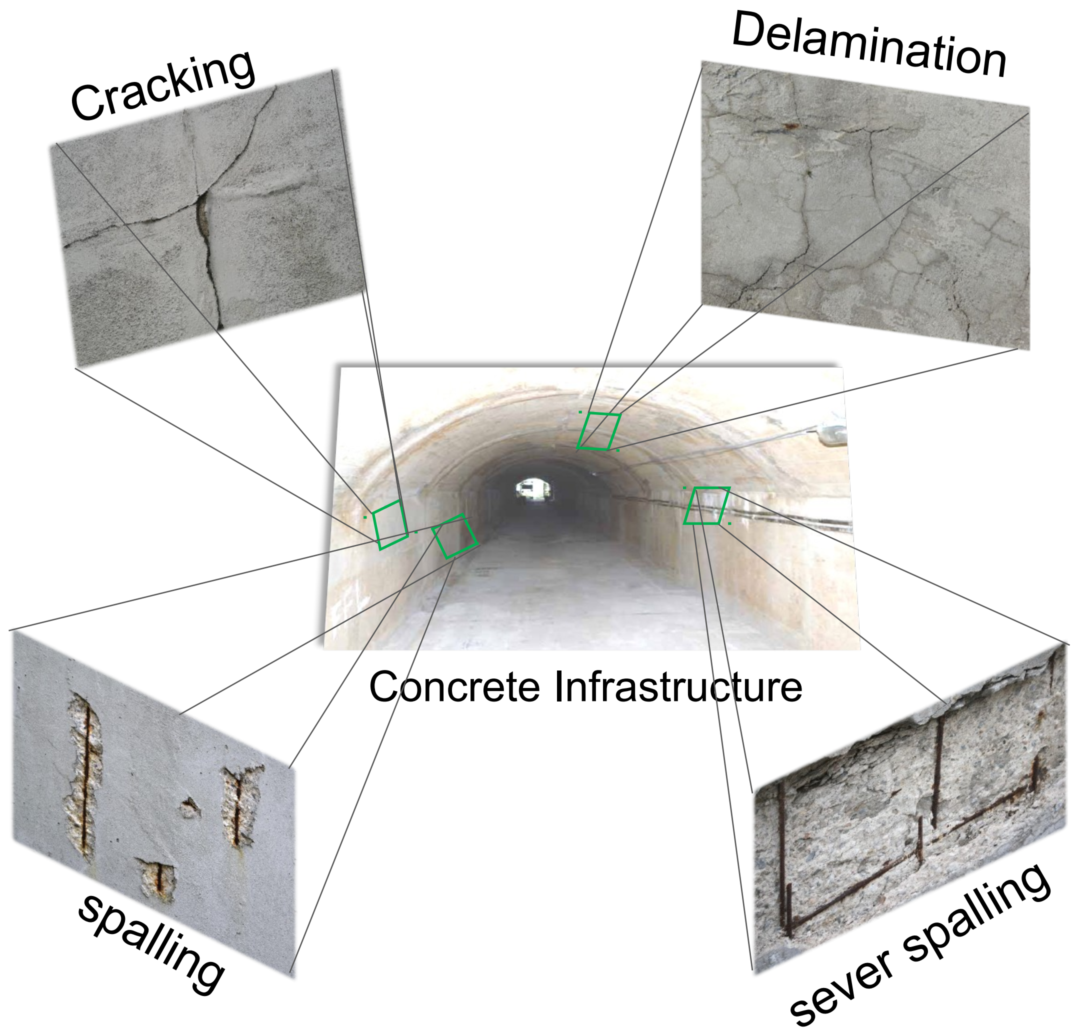
- Problem Statement
- Proposed System Framework
- Localization And Navigation
- Deep Based 3D Registration
- Experiments And Conclusion

Problem Statement

Health Inspection of Concrete Infrastructure :

- Inspection and Registeration:
  - ▫ Detect Region Of Interest (ROI)
  - ▫ Obtain corresponding 3D pose of ROI
  - ▫ Register to 3D model

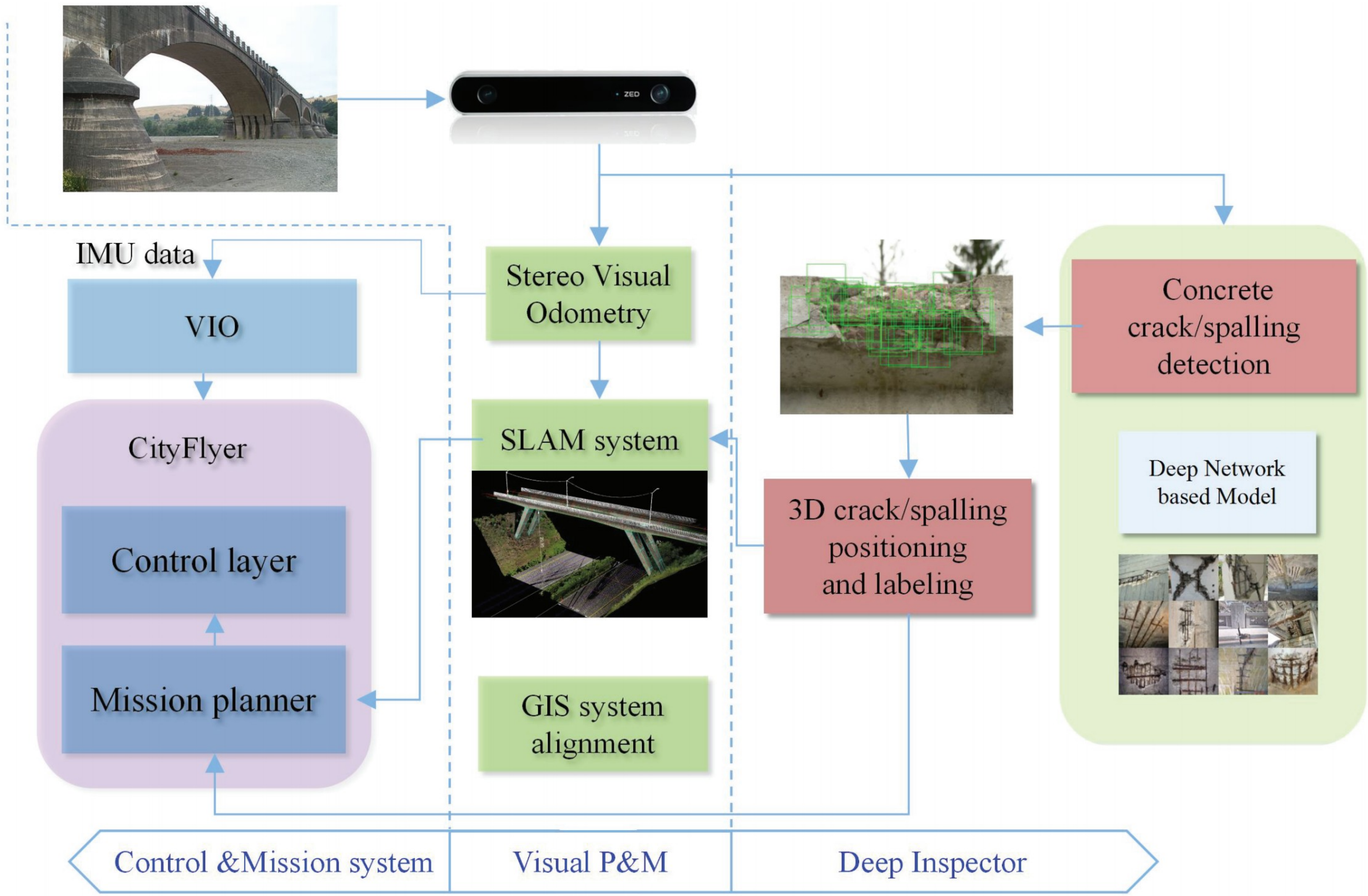
- Issues*:
  - ☐ GPS denied environment positioning
  - ☐ High performance detection



My Research

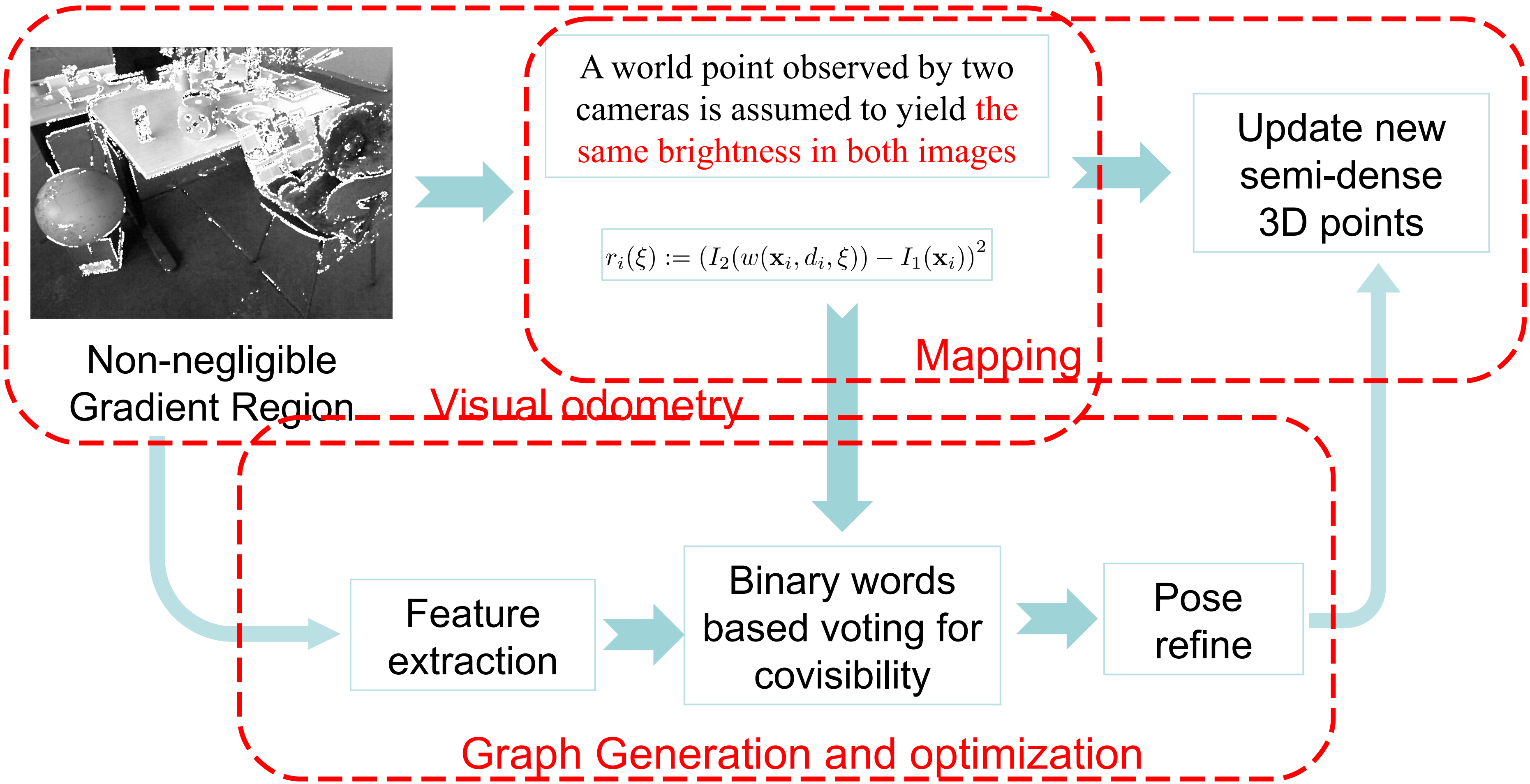
- Problem Statement
- Proposed System Framework
- Localization And Navigation
- Deep Based 3D Registration
- Experiments And Conclusion

Proposed System Framework



Proposed Sub-system —— 1

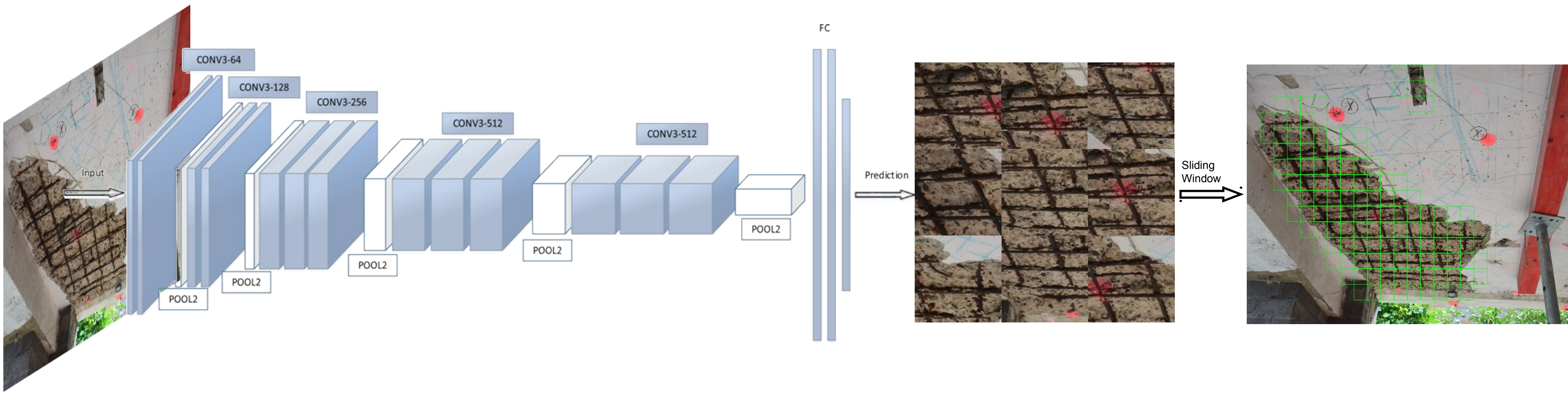
Semi-dense SLAM System:



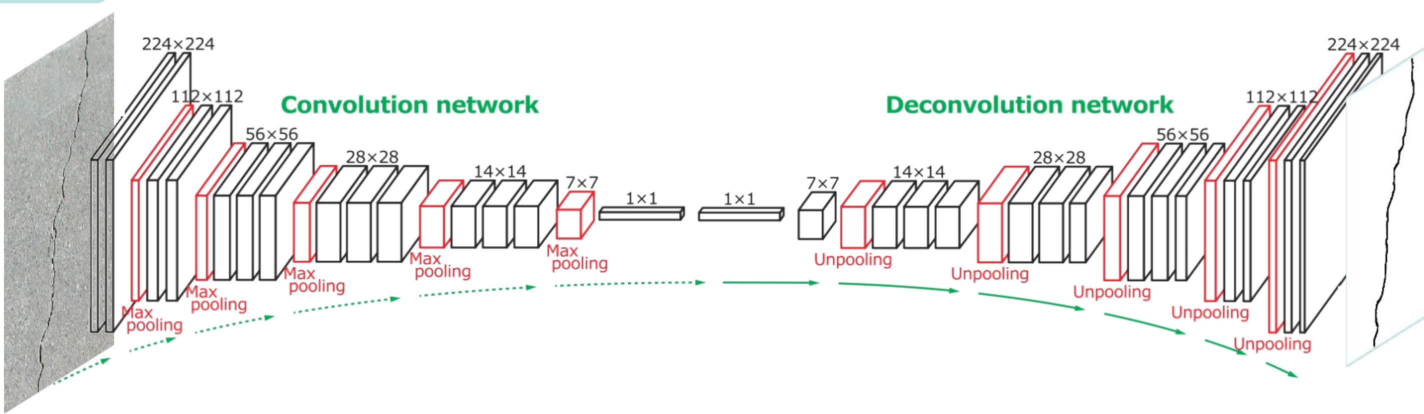
Proposed Sub-system —— 2

Detection And
Label

Spalling



Cracking



My Research

- Problem Statement
- Proposed System Framework
- Localization And Navigation
- Deep Based 3D Registration
- Experiments And Conclusion

Localization And Navigation

Proposed Semi-dense SLAM System Improved Parts:

Semi-dense Direct Approach for Pose Estimation

$$Gradient = \sqrt{g(x)^2 + g(y)^2}$$

$$r_i(\xi) := \left(I_2(w(\mathbf{x}_i, d_i, \xi)) - I_1(\mathbf{x}_i)\right)^2$$

Inverse depth

static stereo depth update

Depth Belief Update

$$\boldsymbol{u}' = \pi\left(\mathbf{T}_{\boldsymbol{\xi}}\pi^{-1}(\boldsymbol{u}, d)\right)$$

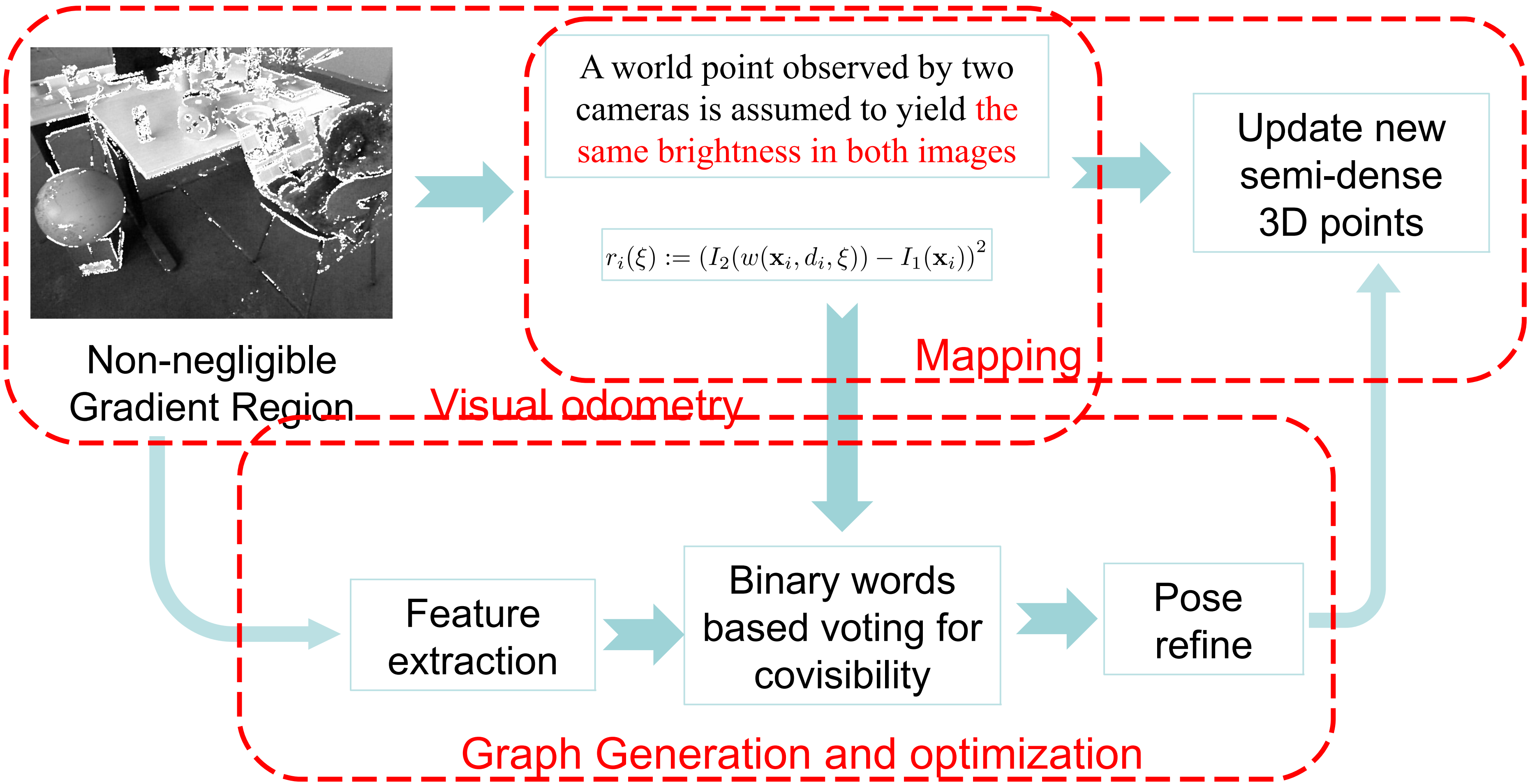
$$d' = \left[\mathbf{T}_{\boldsymbol{\xi}}\pi^{-1}(\boldsymbol{u}, d)\right]_3^{-1}$$

$$\sigma_{d'}^2 = \left(\frac{d}{d'}\right)^4 \sigma_d^2,$$

Semi-sense Mapping

Localization And Navigation

Framework of Semi-dense SLAM:



My Research

- Problem Statement
- Proposed System Framework
- Localization And Navigation
- Deep Based 3D Registration
- Experiments And Conclusion

Deep Based 3D Registration

Data Preparation：

•Searching

- Manually search:
- Web crawler

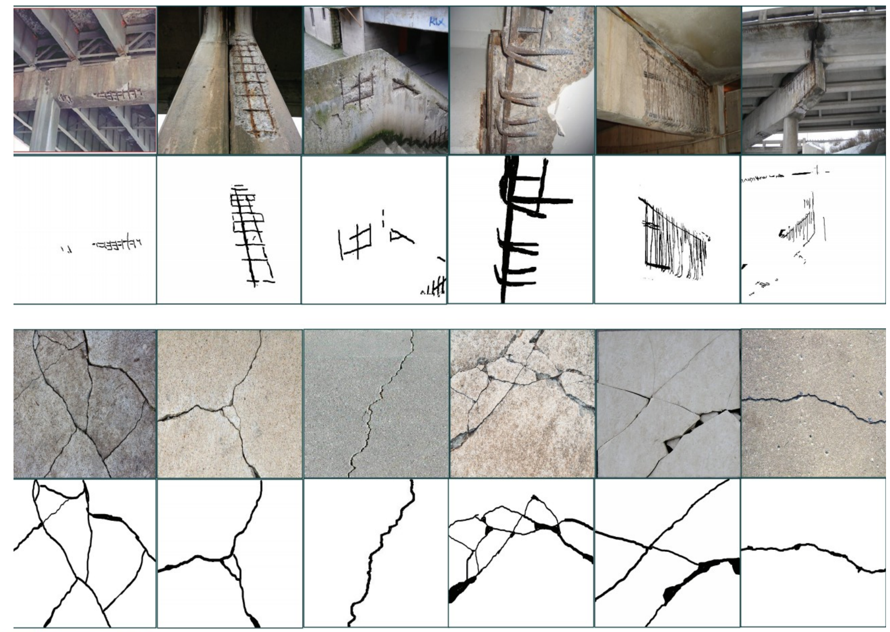
Google, Yahoo, Bing, flicker

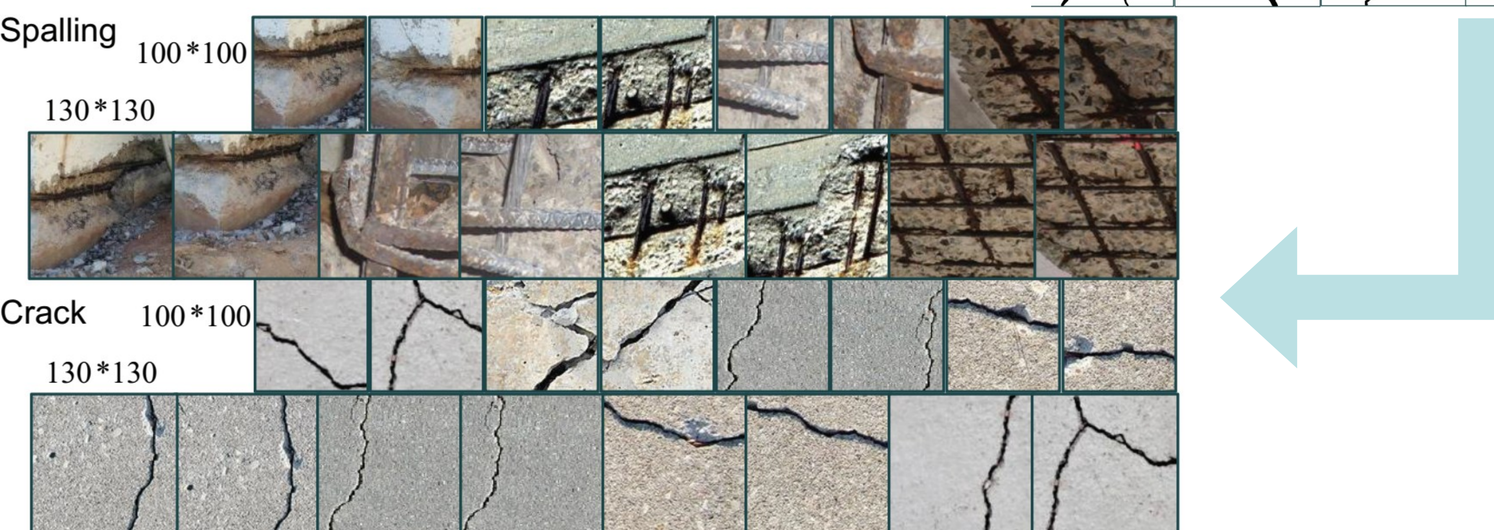
•Labeling

- Most manually
- Pay attention to information you want

Deep Based 3D Registration

- Labeling
  - Most manually
  - Pay attention to information you want





Deep Based 3D Registration

3D Registeration:

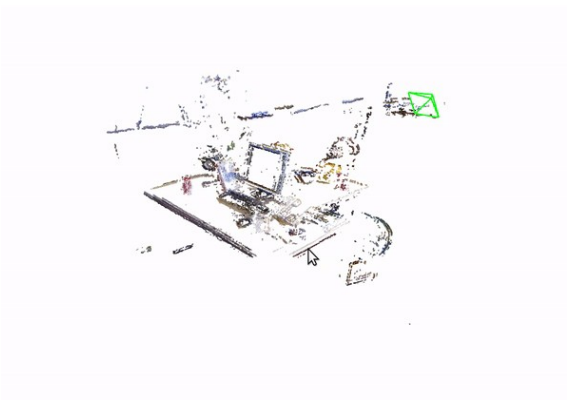
$$R_{S/C} = \{(x,y)\} = \begin{cases} x_1 <= x <= x_2 \&\& y_1 <= y <= y2 \\ \ldots \\ x_{n1} <= x <= x_{n2} \&\& y_{n1} <= y <= yn2 \end{cases}$$

Memorize with filtering

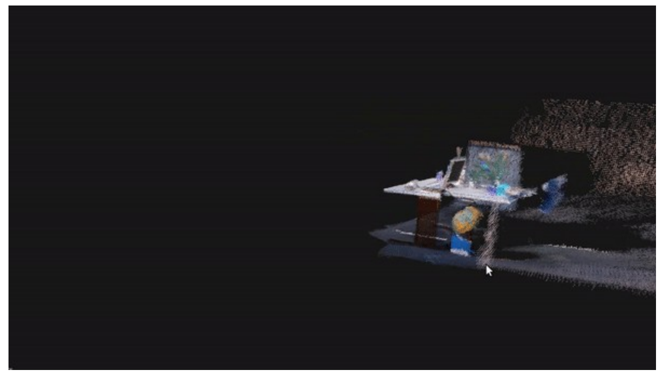
My Research

- Problem Statement
- Proposed System Framework
- Localization And Navigation
- Deep Based 3D Registration
- Experiments

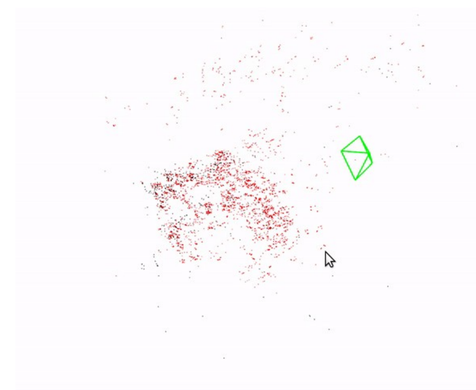
Experiments —— SLAM



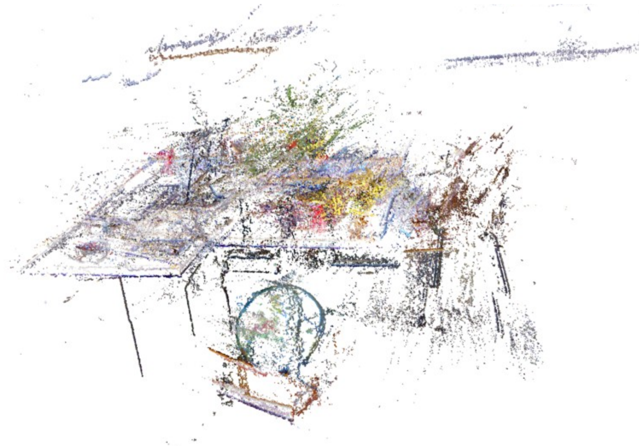
Semi-dense SLAM

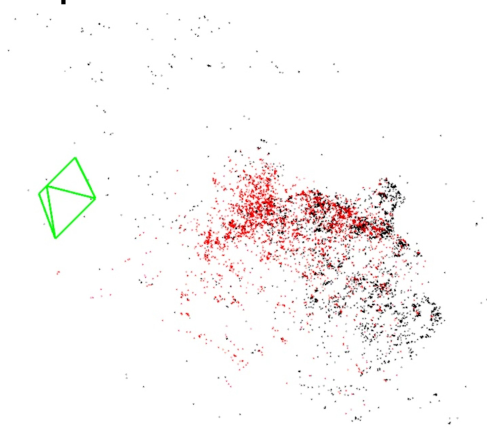


Dense SLAM



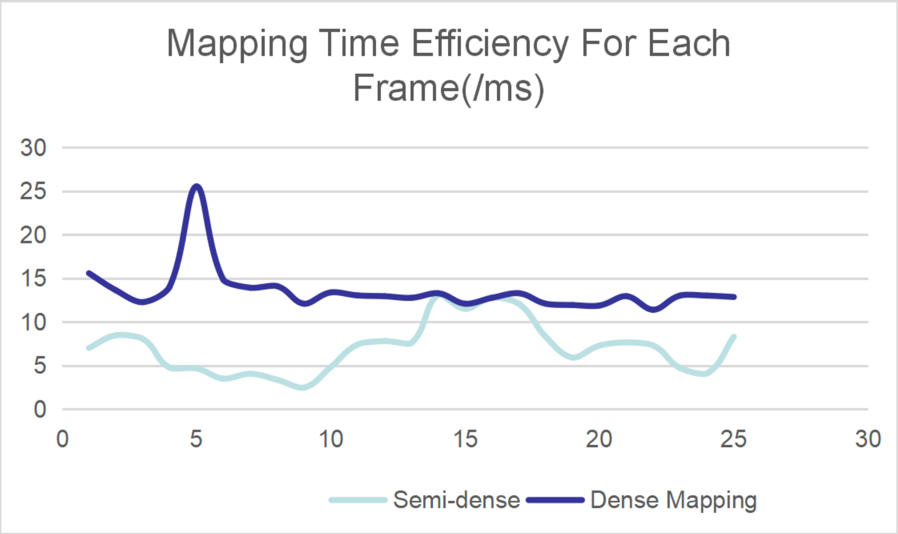
Sparse Feature SLAM

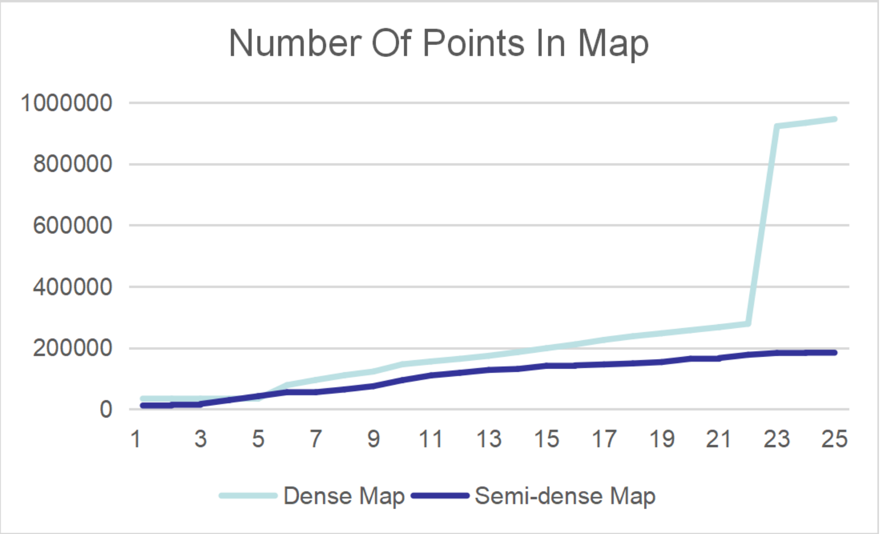




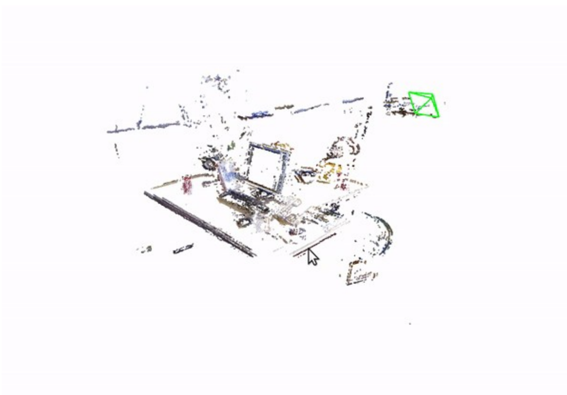
Experiments —— SLAM

	NP	AMT(ms)	AVOT(ms)	NOK	NP	AMT(ms)	AVOT(ms)	NOK
Semi	186256	7.1	5.6	160	41306	3.5	5.2	130
Dense	982163	13.6	---	160	1111697	12.9	--	130
Feature	16257	---	4.0	85	9167	--	4.1	47

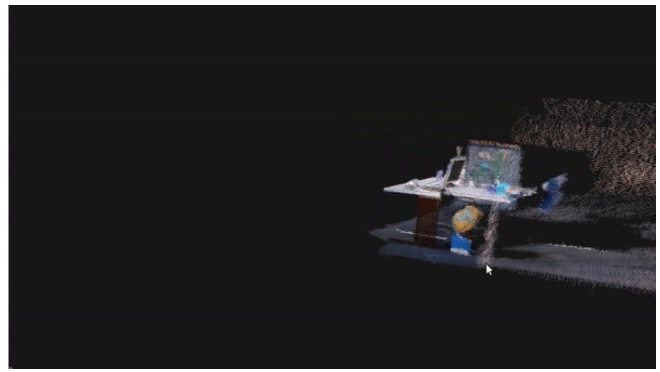




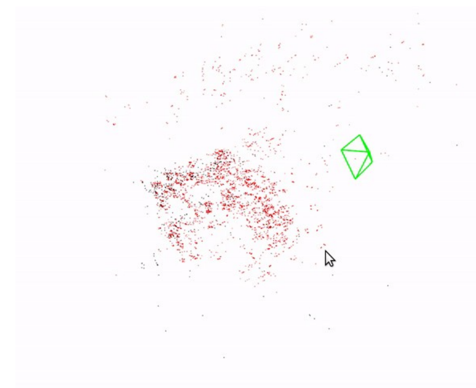
Experiments —— SLAM



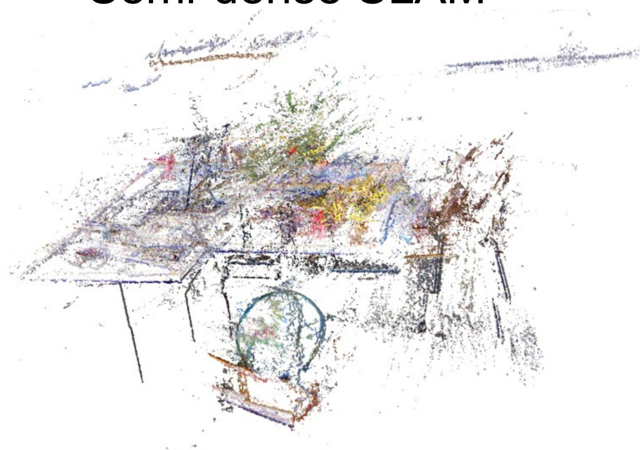
Semi-dense SLAM

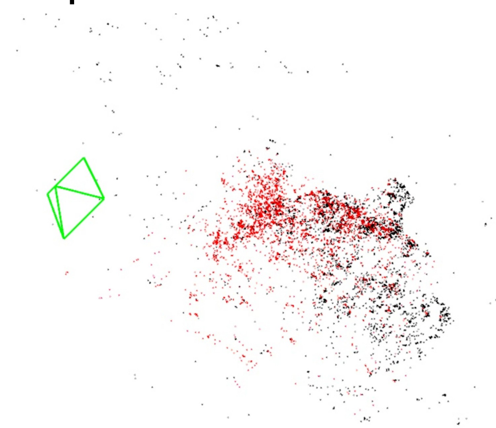


Dense SLAM



Sparse Feature SLAM





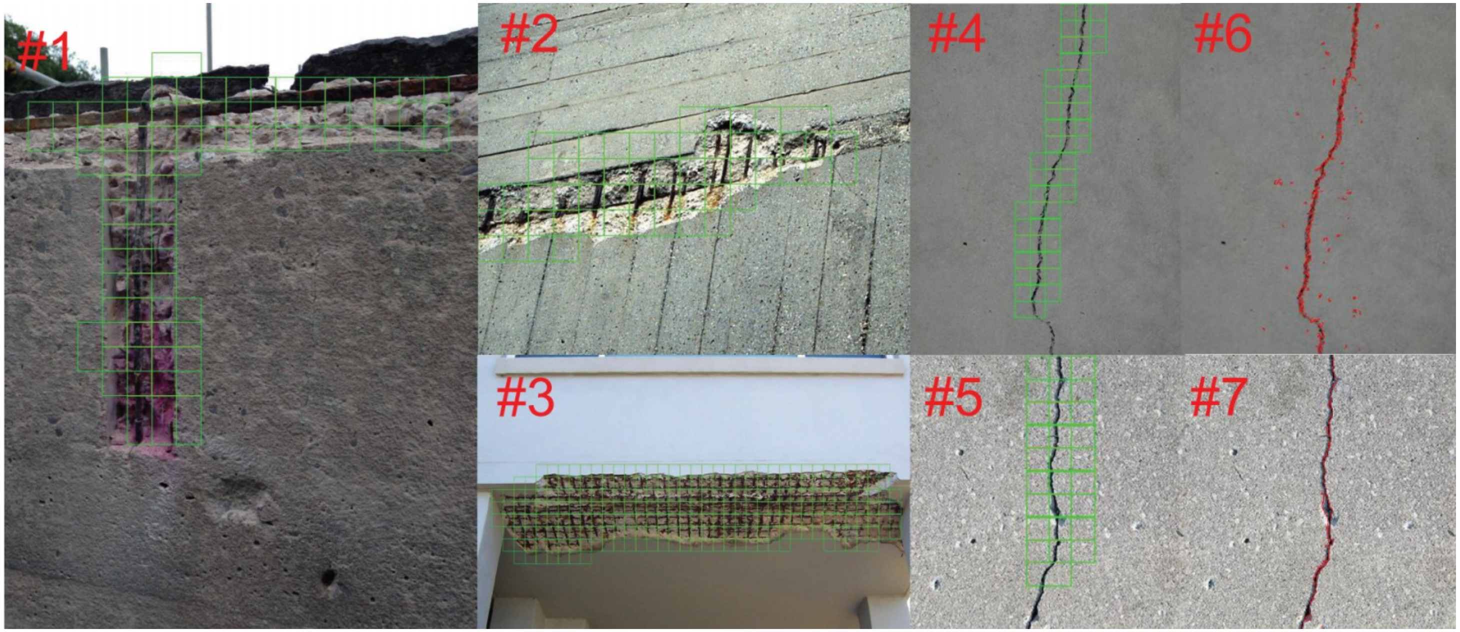
Experiments —— Detection

TABLE II
QUANTIFIED RESULT OF DETECTION WITH CCNY-CSSC DATASET

Dataset	Average Precision (%)	Partial Incomplete Detection (%)	Total Image
CCNY-CSSC	93.36	6.64	1232

TABLE III
FIELD TEST RESULT AT MANHATTAN 155 ST

Test No.	Average Precision (%)	Blurred Image (frames)	Average Precision Without Blur(%)	Over Estimated (%)	Total Image
No.1	72.45	149	76.73	97.18	4998
No.2	67.65	55	71.19	24.3	2650



Experiments —— Detection

Fig. 7. The detection results achieved by CityFlyer in field test 1. Image #1 denotes the trajectory of the CityFlyer, #2 and #3 are detected results, and #4 is the 3D registered model.

Fig. 8. The detection results achieved by CityFlyer in field test 2. Image #1 denotes the trajectory of the CityFlyer, #2 and #3 are detected results, and #4 is the 3D registered model.

Experiments —— Detection