

MATLAB EXPO

MATLAB 기반 딥러닝 모델을 활용한 감시 카메라 영상 내 차량 속도 분석 시스템 개발

최영수 감정관, 국립과학수사연구원 대전과학수사연구소



국립과학수사연구원(NFS)



국립과학수사연구원(NFS): 교통과



교통사고 해석

교통사고 원인 분석 및 사고 재현, 블랙박스 및 CCTV 영상에서 차량 속도 분석, 차대차 뺑소니, 보행자 충돌 및 역과 여부 분석



차량안전 검사

차량 결함 분석, 사고기록장치(EDR) 및 운행기록계 분석, 항공·철도·선박 사고 분석



교통범죄 분석

교통 보험사기에 대한 운전자의 고의성 및 인지 분석, 보복 운전, 사고 위장 및 피해 과장 분석, 차량 탑승자 및 보행자의 인체거동해석



충돌지점, 충돌자세, 충돌속도 분석



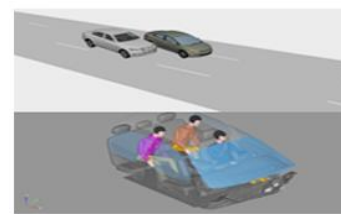
파손된 차량 속도 분석



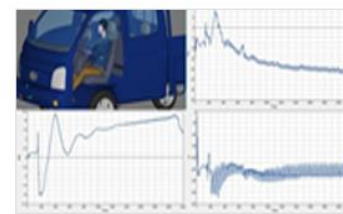
엔진계통 및 샤시 계통 결함 분석



급발진 추정 사고분석



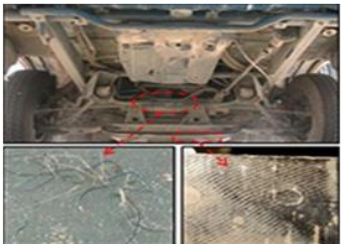
차량 탑승자 운동변화 분석



차량 탑승자 충격력 분석



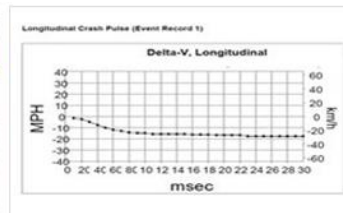
자전거에 부착된 도주차량 페인트



차량하부에 부착된 섬유와 흔적



디지털운행기록계 분석 및 사고해석



EDR 데이터 분석 및 사고해석



진로변경 고의사고 분석



법규위반(꼬리물기) 고의사고 분석

Contents

- **Introduction**

 - Background

 - Limitations

- **Project Overview**

- **Proposed Methods**

 - Module 1: Single object tracking

 - Module 2: Segmentation of wheels

 - Module 3: Prediction of missing data

 - Module 4: Calculation of vehicle speed

- **Conclusion**



Introduction : Background

Importance of vehicle speed at the time of the accident

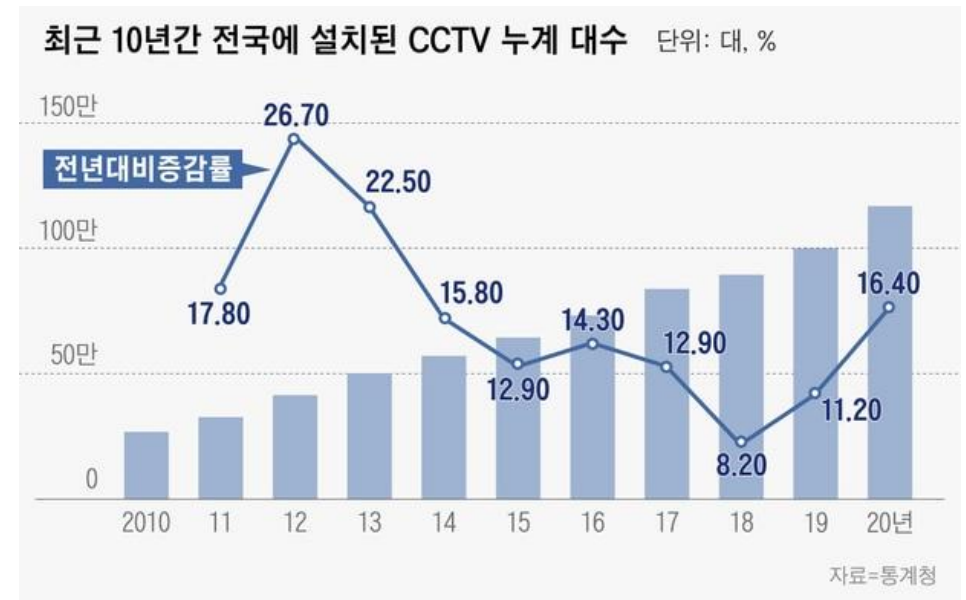
- Criteria for determining the cause and contributing factors of vehicle accident
 - Article 3 of the Act on Special Cases concerning the Settlement of Traffic Accidents (Overspeeding)
 - Traffic accident avoidability
- Used as initial input for car accident reconstruction, which affects the reliability of the reconstruction results.
- Provides critical information to evaluate the effectiveness of safety measures and identify potential areas for improvement
 - Safe Speed 5030 Campaign
 - Install speed bumps and speed cameras



Introduction : Background

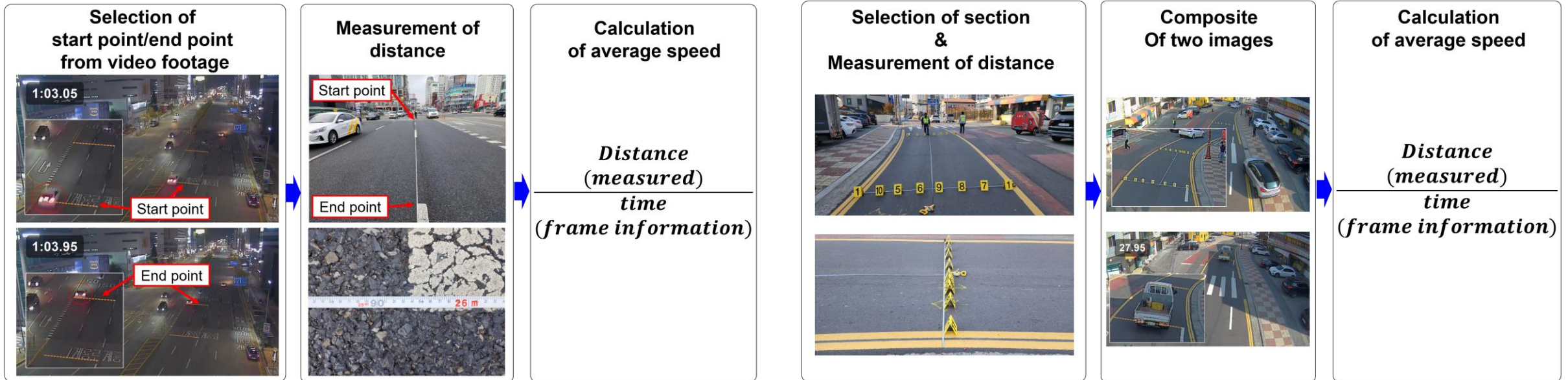
Importance of surveillance camera in accident analysis

- Intuitive understanding of vehicle accident situations
 - Useful in determining the cause and circumstances of an accident
- Provide valuable information about the conditions at the time of the accident
 - Traffic conditions(vehicle position, direction)
- Surveillance camera installation status
 - Status of CCTV installations in public institutions
(1,336,653 as of 2020)



Introduction : Limitations

Limitations of the existing method



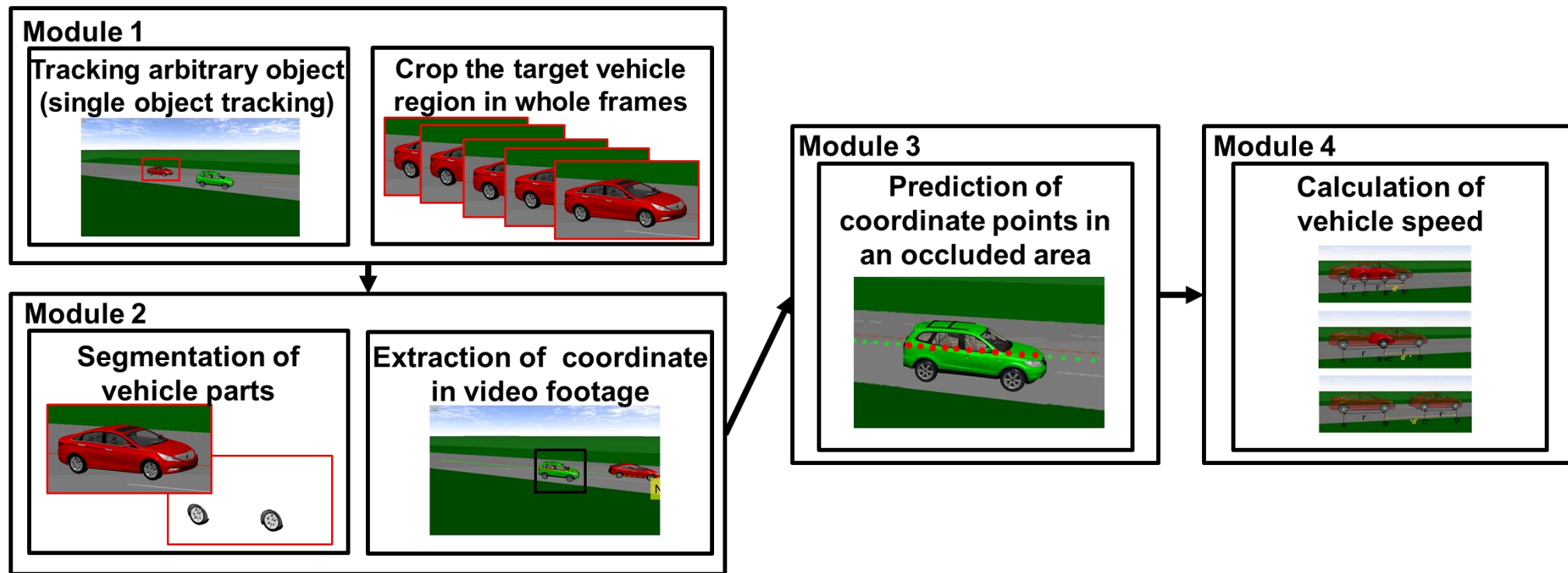
- A start and end point must exist
- Difficulty analyzing speed changes
- Too much on-site investigation

Project Overview

Goal : Vehicle speed analysis system in stationary camera

- Function to analyze instantaneous speed over time
- Function to automatically analyze the speed of a target vehicle in a video

Project pipeline



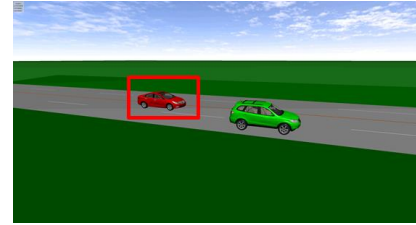
Methods : Module 1

Challenges

- Low resolution
- Occlusion
- Change the target shape
- Scale variation

Module 1

Tracking arbitrary object (single object tracking)



Crop the target vehicle region in whole frames



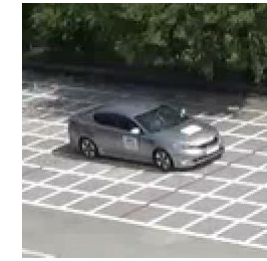
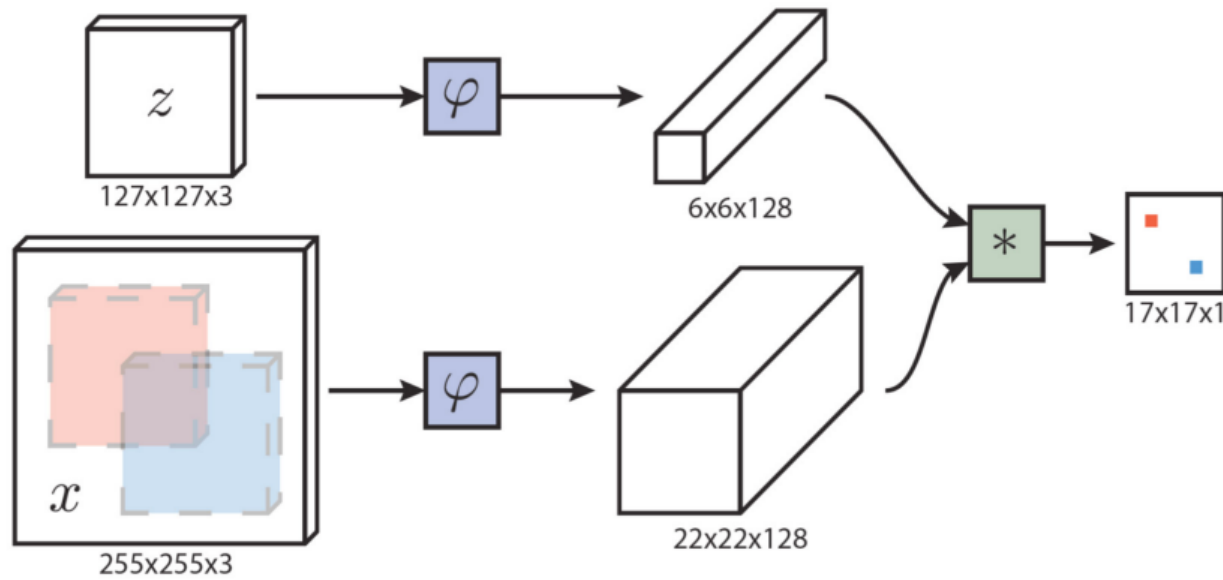
Existing methods for tracking by MATLAB

- MATLAB and Computer Vision Toolbox™ provides video tracking algorithms
 - CAMShift (Continuously Adaptive Mean Shift), KLT (Kanade-Lucas-Tomasi)

Proposal method

- Deep learning model : SiamFC(Siamese Fully Convolutional Networks)
 - Object tracking algorithm published in 2016
 - SiamFC uses a Siamese Network based on a Convolutional Neural Network (CNN) to track objects.

Methods : Module 1

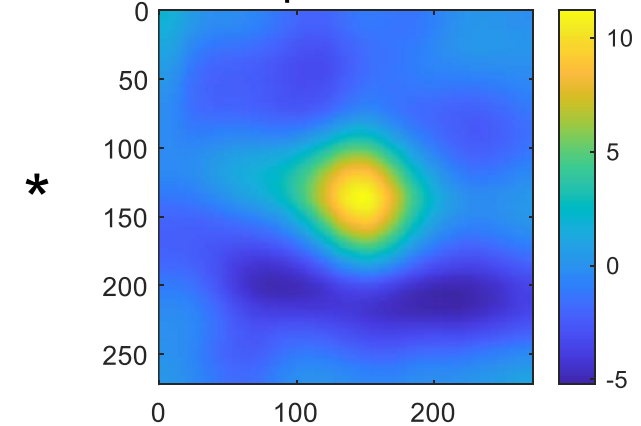


Subplot 1: X_c rops



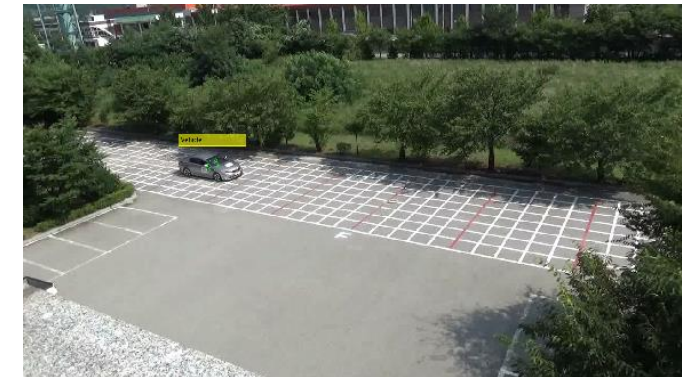
Z

Subplot 2: Score



X

Subplot 3: Image



Methods : Module 1

Improvement : Apply a Kalman filter

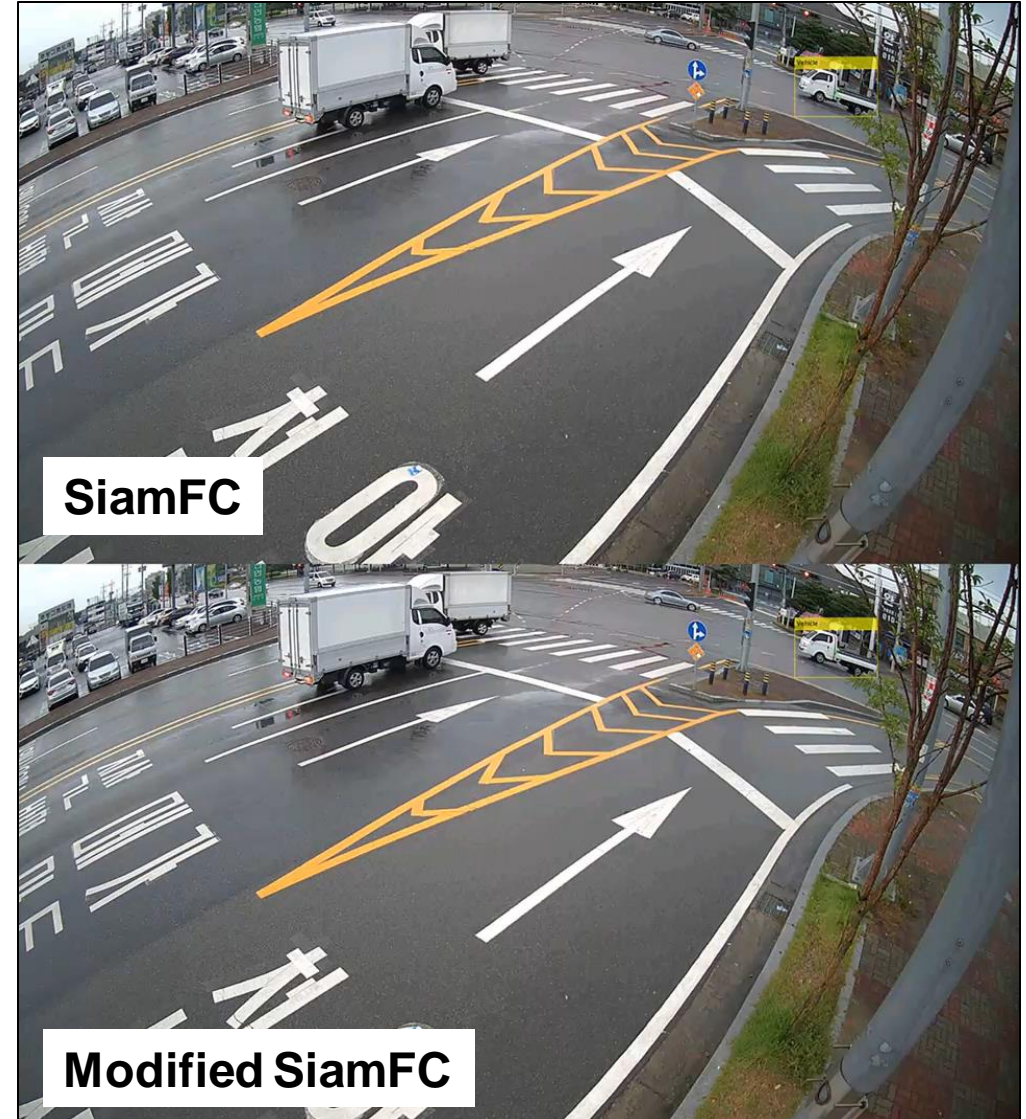
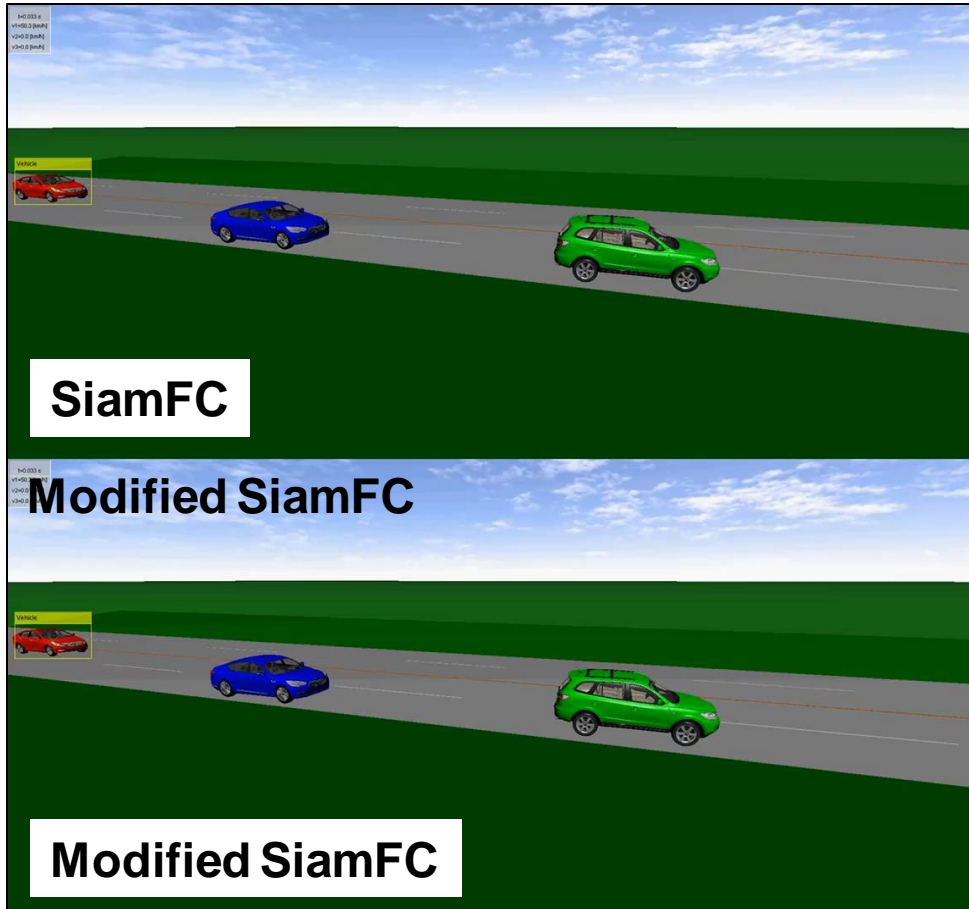


```
function newTargetPosition bestScale, score, response size  
= tracker_eval(dlnetArea, s_x, exemplarFeatures, x_crops,  
targetPosition, window, adjust, numScale, scalePenalty,  
wInfluence, scoreSize, responseUp, totalStride, instanceSize)
```

```
if score_max <= 3  
    targetPosition_cor = predict(filter);  
    targetPosition = targetPosition_cor;  
  
else  
    predict(filter);  
    targetPosition_cor = correct(filter, targetPosition);  
    targetPosition = targetPosition_cor;  
end
```

Methods : Module 1

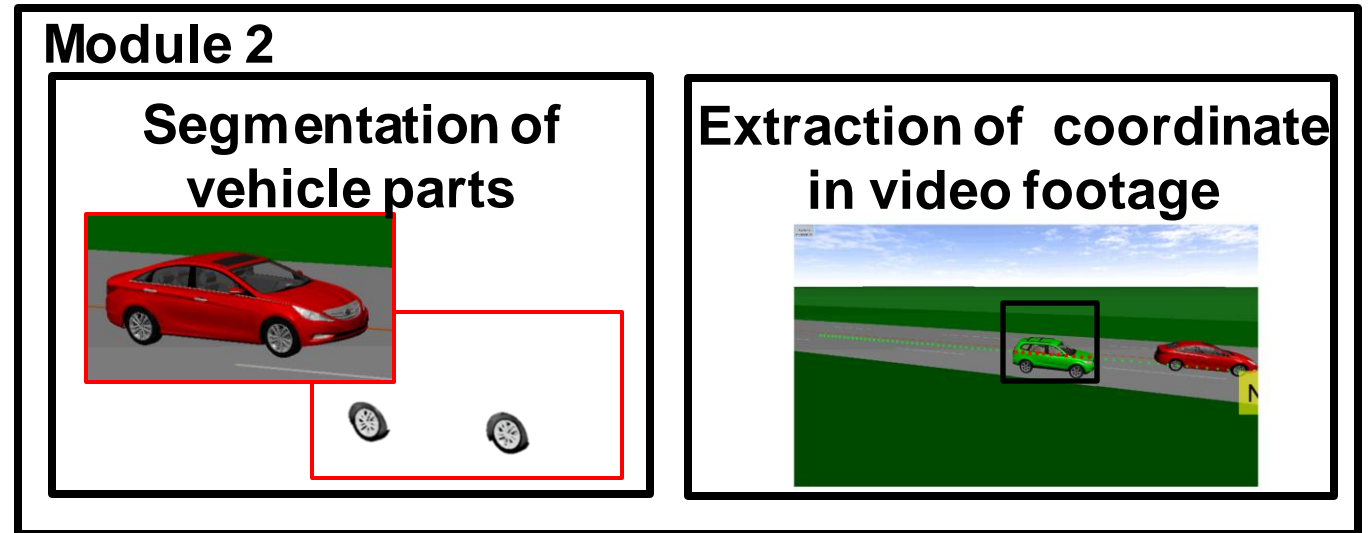
Result : Tracking video example



Methods : Module 2

Challenges

- Low resolution
- Change the target shape
- Scale variation



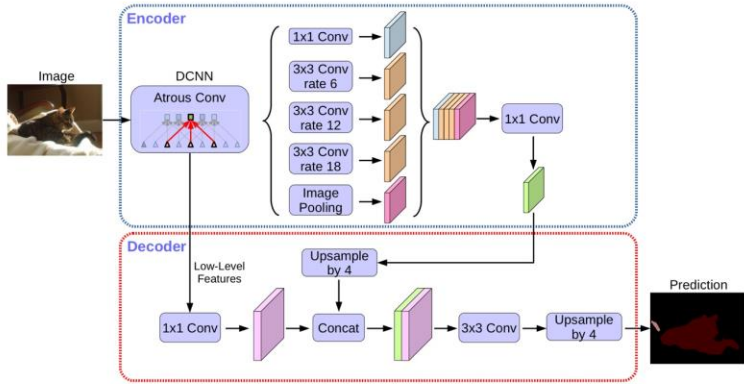


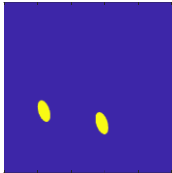
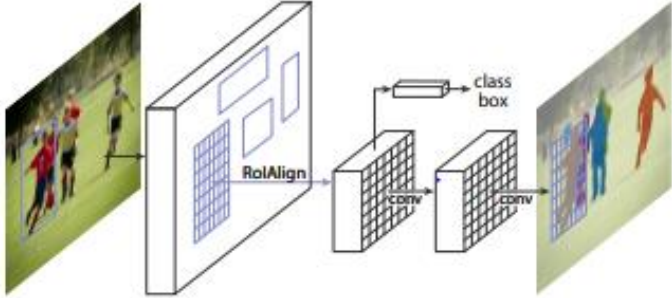


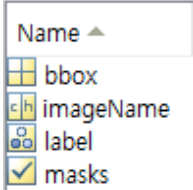
Existing methods for segmentation by MATLAB

- MATLAB and Image Processing Toolbox™ provides Various image segmentation techniques
- binarization, clustering, graph-based segmentation, and region expansion
- Deep Learning Toolbox™ provides a framework for designing and implementing deep neural networks using algorithms, pre-trained models, and apps

Proposal method

- Deep learning model : Deeplabv3+, Mask R-CNN

Methods : Module 2

Model	Output	Input(training)	Difference
<p>DeepLabV3+</p> 	<p>Semantic segmentation</p>  <p>Semantic Map</p>	<p>CamVid dataset type</p> <p>image</p>  <p>Label</p> 	<ul style="list-style-type: none"> ❖ Models based on Fully Convolutional Network (FCN) ❖ Perform per-pixel categorization and segmentation ❖ Perform pixel-by-pixel classification and segmentation for the entire image
<p>Mask R-CNN</p>  <p>Figure 1. The Mask R-CNN framework for instance segmentation.</p>	<p>Instance segmentation</p>  <p>Classification Label Bounding Box Pixel Classification</p>	<p>COCO dataset type</p> <p>image</p>  <p>Label</p> 	<ul style="list-style-type: none"> ❖ Models based on Faster R-CNN ❖ Models that perform object detection and segmentation simultaneously ❖ Predict the bounding box of an object and perform object segmentation in that area

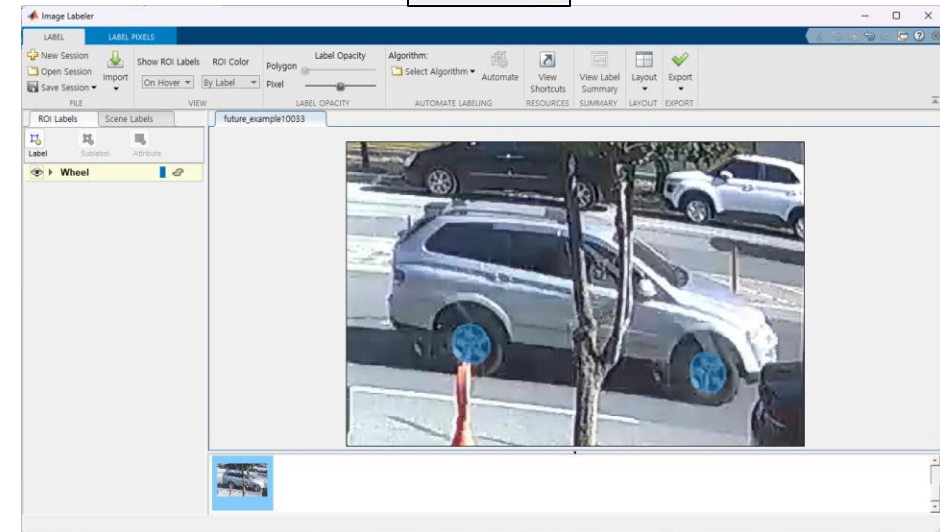
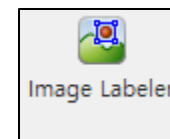
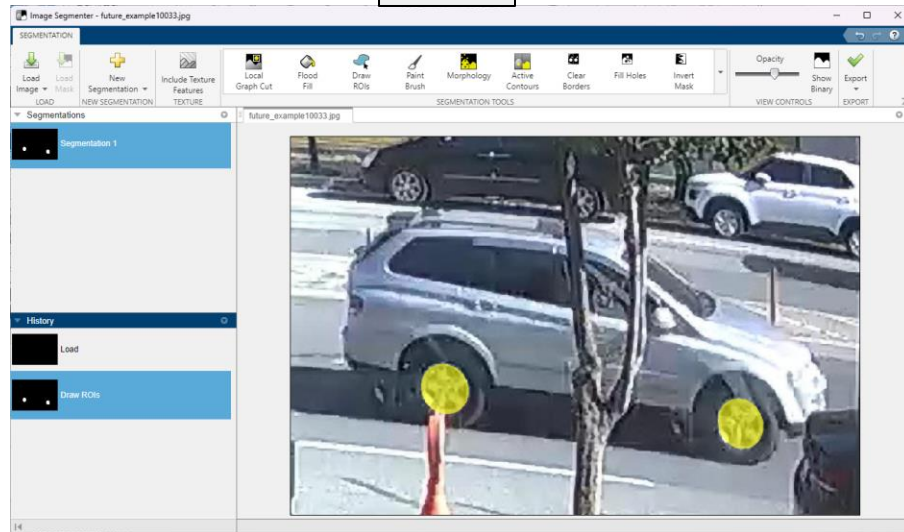
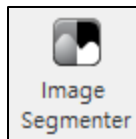
He, Kaiming, et al. "Mask r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2017.

Chen, Liang-Chieh, et al. "Encoder-decoder with atrous separable convolution for semantic image segmentation." *Proceedings of the European conference on computer vision (ECCV)*. 2018. ¹³

Methods : Module 2

Improvement : Prepare new dataset(train:421, val:105)

Data labeling



Methods : Module 2

Training : DeeplabV3+

The screenshot displays the MATLAB Live Editor interface for training a DeeplabV3+ model. The code editor on the left contains the following code:

```

1 clear
2 clc
3 close all

Training DeeplabV3+ model

- Training Deeplab v3+ model by pretrained backbone network
- Backbone network: 'resnet18', 'resnet50','mobilenetv2','xception','inceptionresnetv2'


4 backbone = inceptionresnetv2();

Classes
5 classes = [
6     "background"
7     "wheel"
8 ];
9
10 pixelLabelID = [1 2];

Dataset
11 workingdir = '/media/nfs/NFS/Project/Segmentation/car_parts_segmentation/DeeplabV3plus/datasets/newdat
12 outputFolder = fullfile(workingdir, 'Result');
13
14 if ~exist(outputFolder)
15     mkdir(outputFolder)
16 end
17
18 imgDir = fullfile(workingdir, 'imgs');
19 lblDir = fullfile(workingdir, 'labels');
20
21 imds = imageDatastore(imgDir);
22 pxds = pixelLabelDatastore(lblDir, classes, pixelLabelID);
  
```

The training progress table in the center shows the following data:

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Validation Accuracy	Mini-batch Loss	Validation Loss
1	1	00:00:20	17.21%	98.32%	0.9847	7.7964
1	2	00:00:22	53.31%		2.7348	
1	4	00:00:26	68.62%		1.2888	
1	6	00:00:31	70.28%		0.9287	
1	8	00:00:35	94.22%		0.5432	
1	10	00:00:39	69.69%		0.3738	
1	12	00:00:44	73.82%		0.8046	
1	14	00:00:48	73.72%		0.3044	
1	16	00:00:52	76.37%		0.3429	
1	18	00:00:56	77.39%		0.3564	
1	20	00:01:01	78.15%		0.4060	

The plot on the right shows the training progress, including accuracy and loss over time. The top plot shows accuracy (blue line) and loss (red line) over iterations. The bottom plot shows the training progress over epochs.

Methods : Module 2

Training : Mask R-CNN

Perform Instance Segmentation Using Mask R-CNN

This example shows how to segment individual instances of people and cars using a multiclass mask region-based convolutional neural network (R-CNN).

Instance segmentation is a computer vision technique in which you detect and localize objects while simultaneously generating a segmentation map for each of the detected instances.

This example first shows how to perform instance segmentation using a pretrained Mask R-CNN that detects two classes. Then, you can optionally download a data set and train a multiclass Mask R-CNN using transfer learning.

Download Training Data

Create directories to store the COCO training images and annotation data.

```

1 clear
2 clc
3 close all
4
5 datafolder = '/media/nfs/NFS/Project/module2/car_parts_segmentation/MaskRCNN/GOGO/newdata/Final_dataset (2023.04.0
6
7 img_dir = fullfile(datafolder, 'imgs');
8 label_dir = fullfile(datafolder, 'New_labels');
9
10 img_files = dir(fullfile(datafolder, 'imgs', '*.jpg'));
11 label_files = dir(fullfile(datafolder, 'New_labels', '*.mat'));

```

Prepare Data for Training

```

12 rng(1); % 재현성을 위해 랜덤 시드를 고정합니다.
13 n = numel(img_files);
14 idx = randperm(n);
15 train_idx = idx(1:round(0.9*n)); % 전체 데이터 중 90%를 train set으로 사용합니다.
16 val_idx = idx(round(0.9*n)+1:end); % 전체 데이터 중 10%를 validation set으로 사용합니다.

```

trainset, validationset 저장

```

17 train_img_files = img_files(train_idx);
18

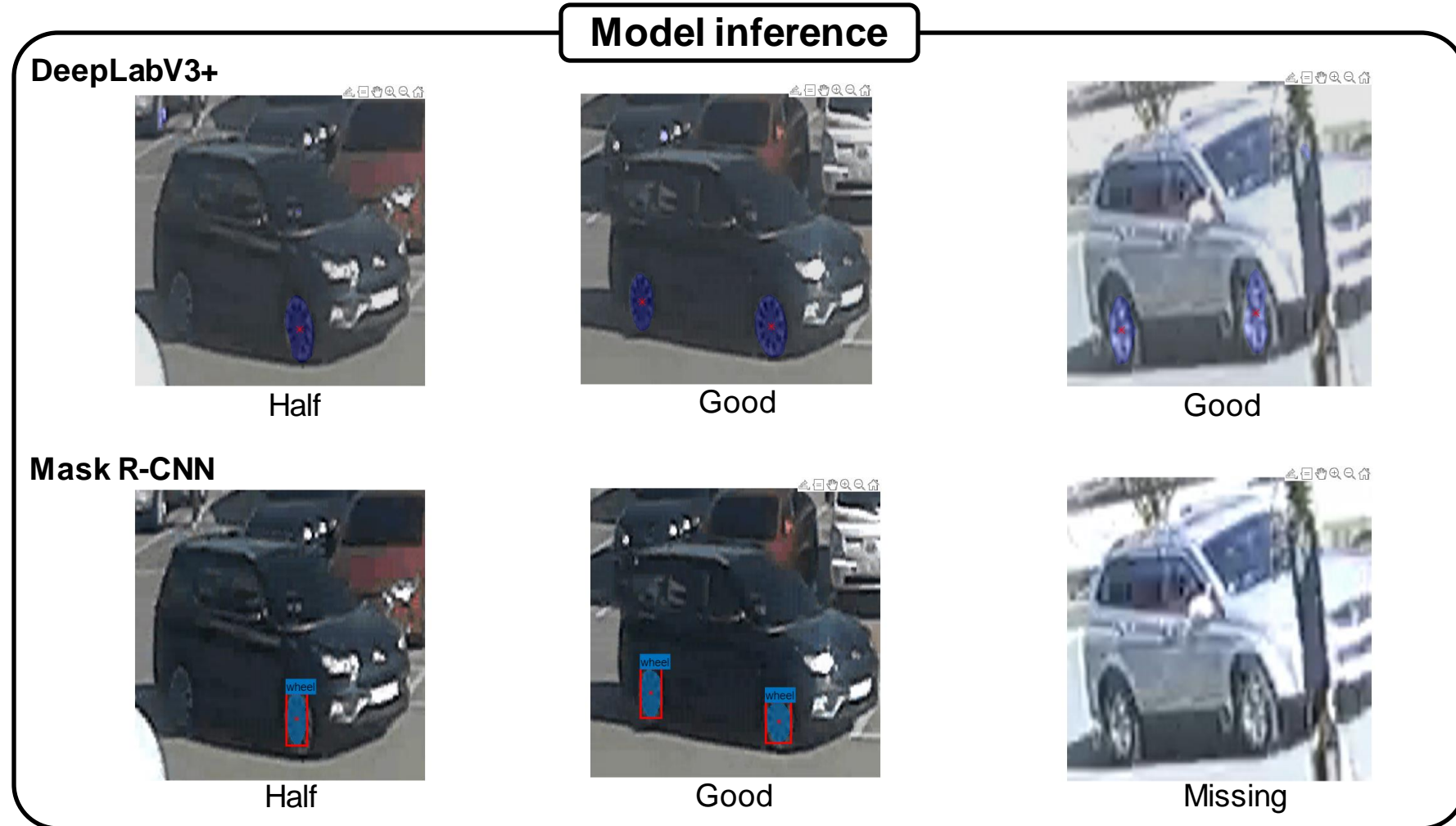
```

Epoch	Iteration	TimeElapsed	LearnRate	TrainingLoss	TrainingRPNLoss
1	50	00:01:13	0.001	2.434	1.7698
1	100	00:02:32	0.001	3.4499	2.6985
1	150	00:03:52	0.001	2.3309	1.6349
1	200	00:05:13	0.001	0.90718	0.42809
1	250	00:06:32	0.001	1.544	0.95475
1	300	00:07:57	0.001	1.3219	0.80353
1	350	00:09:19	0.001	1.2482	0.79007
1	400	00:10:41	0.001	0.81032	0.31103
1	450	00:12:03	0.001	1.0844	0.96008
2	500	00:13:47	0.00095	1.9481	1.2563
2	550	00:15:09	0.00095	1.0634	0.6491
2	600	00:16:32	0.00095	0.77503	0.28582
2	650	00:17:55	0.00095	0.75752	0.1143
2	700	00:19:17	0.00095	1.1176	0.60923

Zoom: 110% UTF-8 LF script

Methods : Module 2

Result : Comparison between DeepLabv3+ and Mask R-CNN



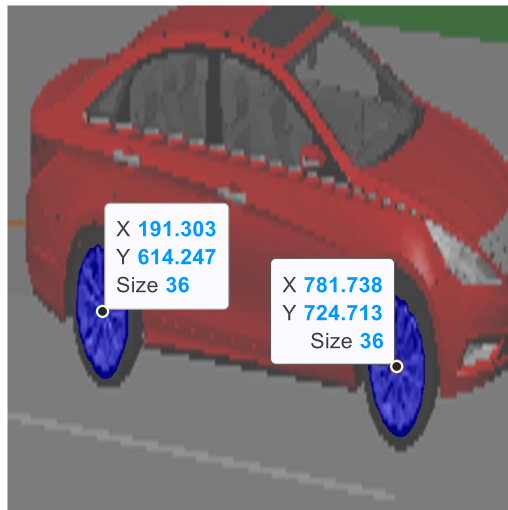
Methods : Module 2

Result : Extracting coordinates, imposing them on the original image

Cropped image



Resize Extraction



Imposing



regionprops

Measure properties of image regions

measures properties such as area, centroid, and bounding box, for each object (connected component) in an image.

Methods : Module 3

Challenges

- Missing coordinates
- Inaccurate coordinate data

Existing methods for estimation by MATLAB

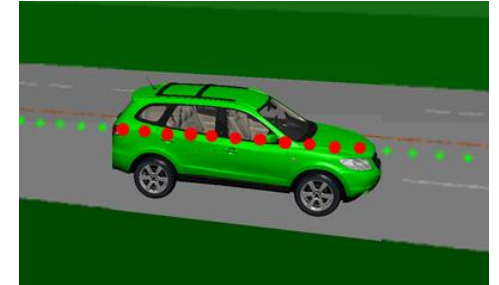
- MATLAB and Automated Driving Toolbox™ provides Various Kalman filter techniques
 - Alpha-Beta filter, Linear Kalman filter, Extended Kalman filter, Unscented Kalman filter
- MATLAB provides 'Preprocessing Data'
 - anymissing, ismissing, fillmissing, rmmissing, standardizeMissing functions

Proposal method

- **Kalman filter with smoothing**

Module 3

Prediction of coordinate points in an occluded area



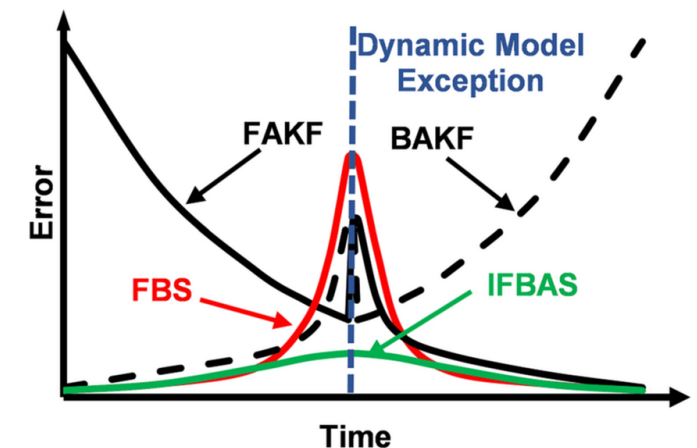
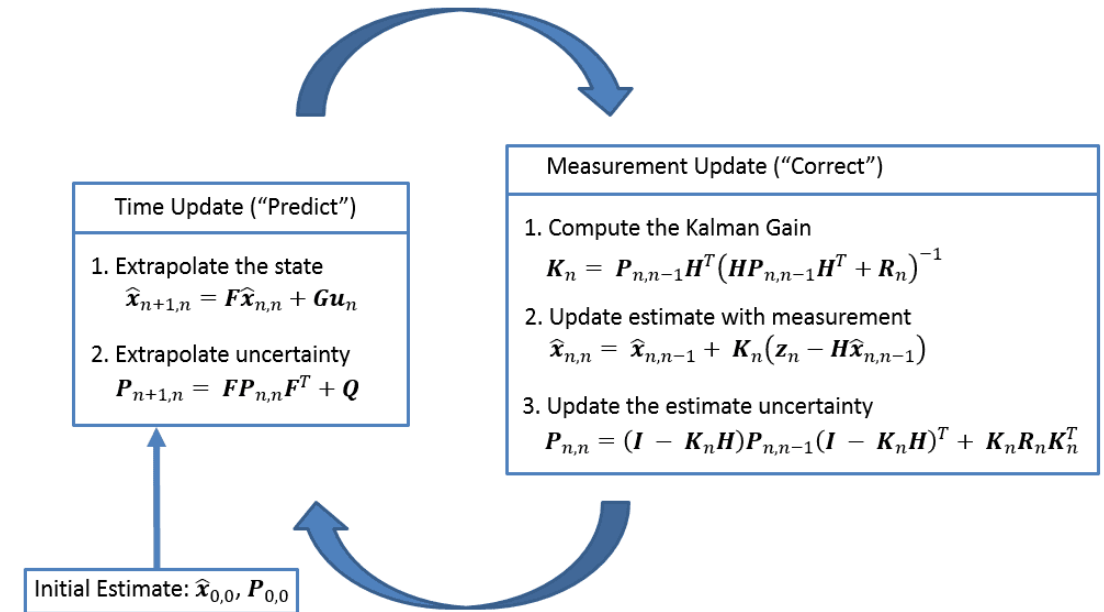
Methods : Module 3

What is Kalman filter

- Prediction(system model) and Update(measurement)
- Recursive algorithm
- Real time operation

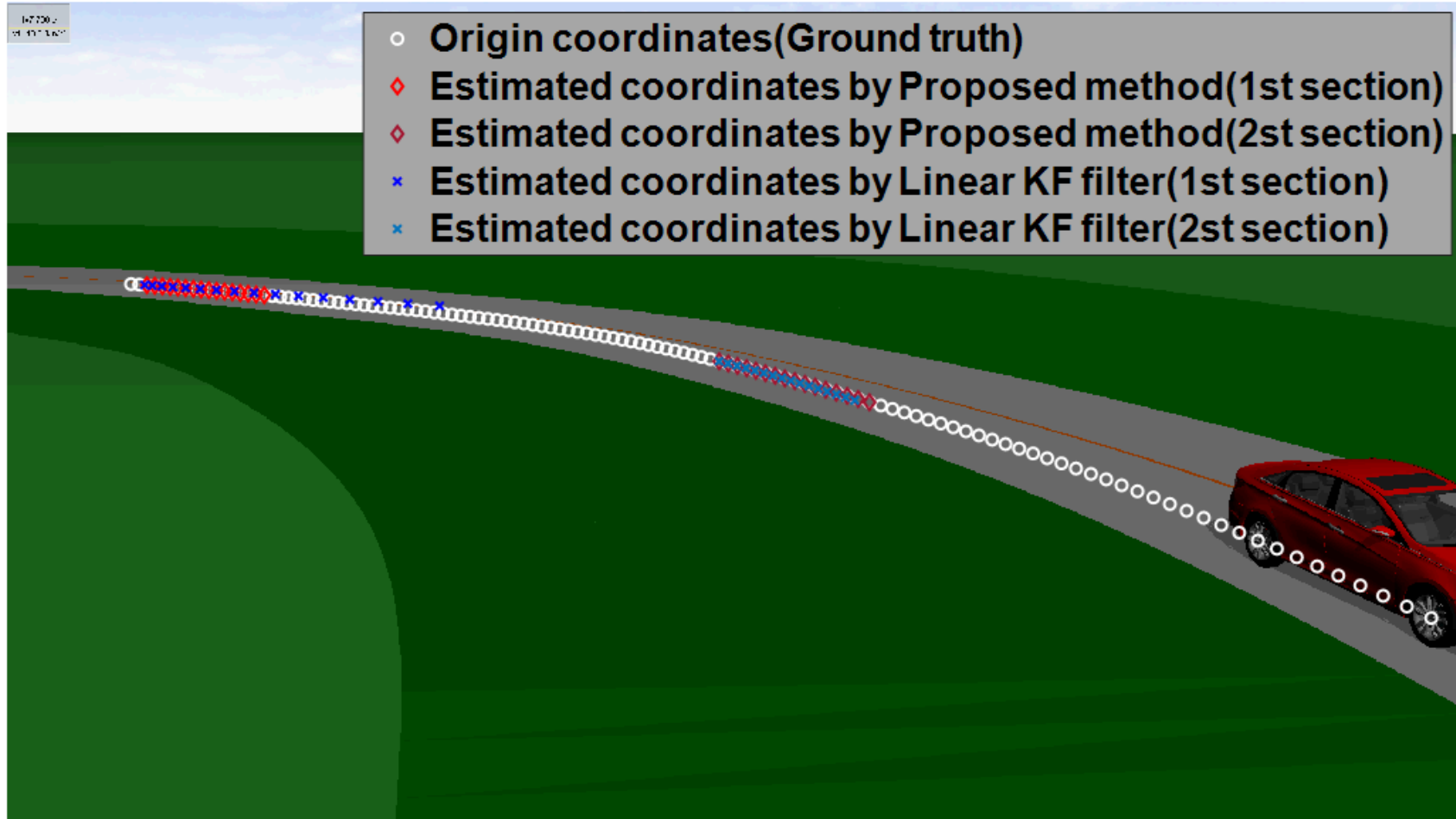
Improvement : Smoothing

- Forward pass and Backward pass
- Use the state estimated by the Kalman filter and the error covariance to estimate the final state, taking into account all observations from past time periods.



Methods : Module 3

Result : Estimation of missing data



Methods : Module 4

Challenges

- Using known information (Vehicle specification)
- Reflection the geometry of Cross-Ratio

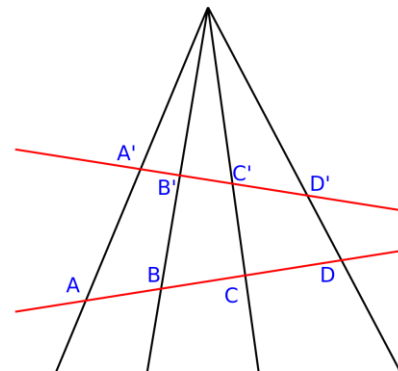
Implementation in MATLAB

- Function to automatically calculate the speed of a vehicle
- Function to analyze how a vehicle' speed changed over time

What is Cross-ratio?

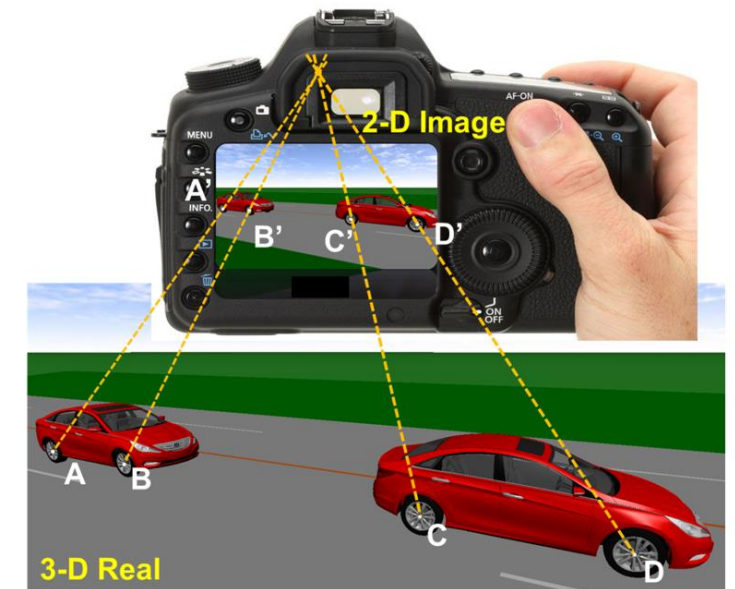
- Invariant value

$$(A, B; C, D) = \frac{AC \cdot BD}{BC \cdot AD}$$



Module 4

Calculation of vehicle speed

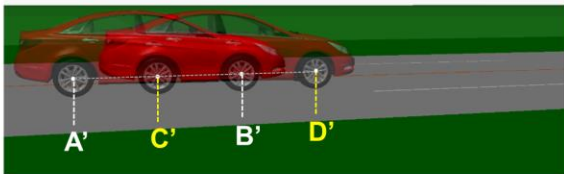


Methods : Module 4

How to calculate vehicle speed?

- Using invariant property
- Using vehicle' wheel length

Case 1: $|A'B'| > |A'C'|$

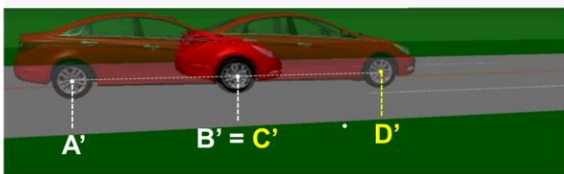


$$CR(\text{Cross - ratio}) = (A'C'B'D') = \frac{|A'B'| \times |C'D'|}{|A'D'| \times |B'C'|}$$

$$d_{i,j}(\text{distance}) = \sqrt{l^2 - \frac{l^2}{CR}}$$

$$v_{i,j}(\text{speed}) = \frac{d_{i,j} \times FPS}{|i - j|}$$

Case 2: $|A'B'| = |A'C'|$

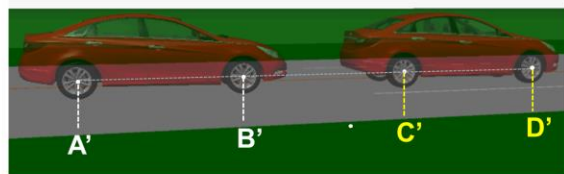


$$CR(\text{Cross - ratio}) = \infty$$

$$d_{i,j}(\text{distance}) = l$$

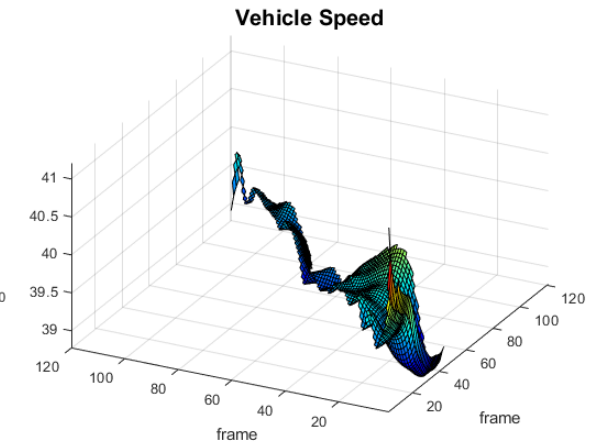
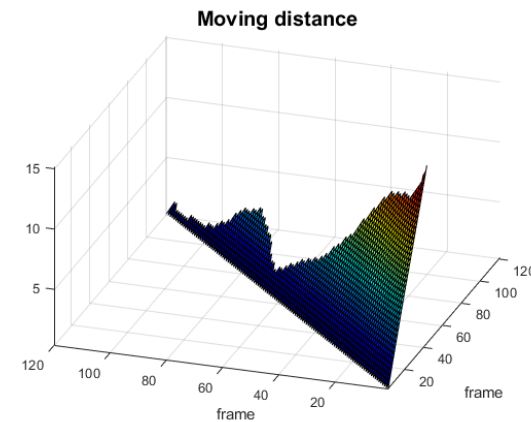
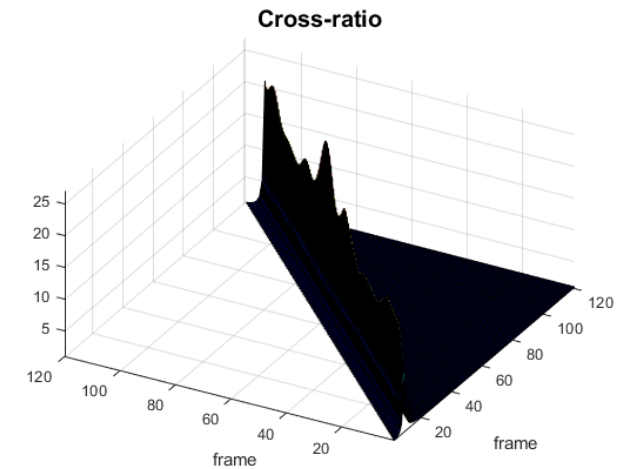
$$v_{i,j}(\text{speed}) = \frac{d_{i,j} \times FPS}{|i - j|}$$

Case 3: $|A'B'| < |A'C'|$



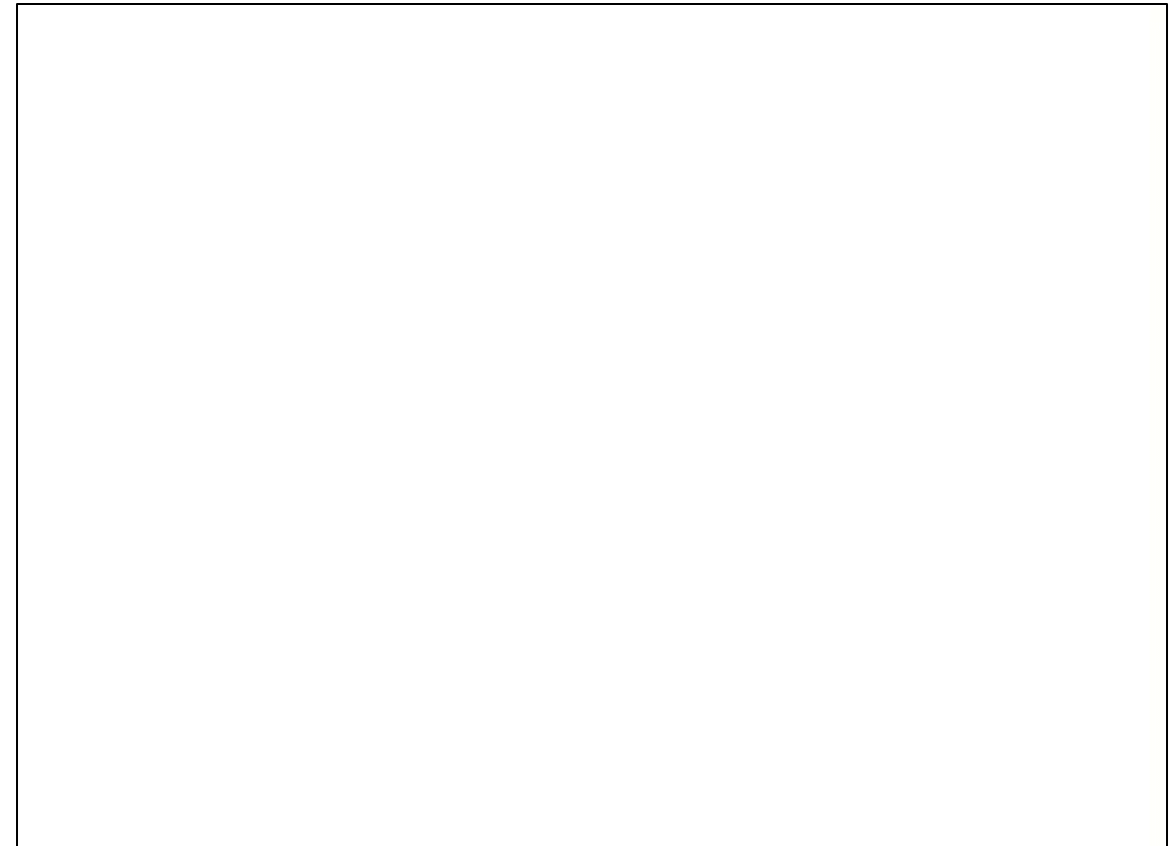
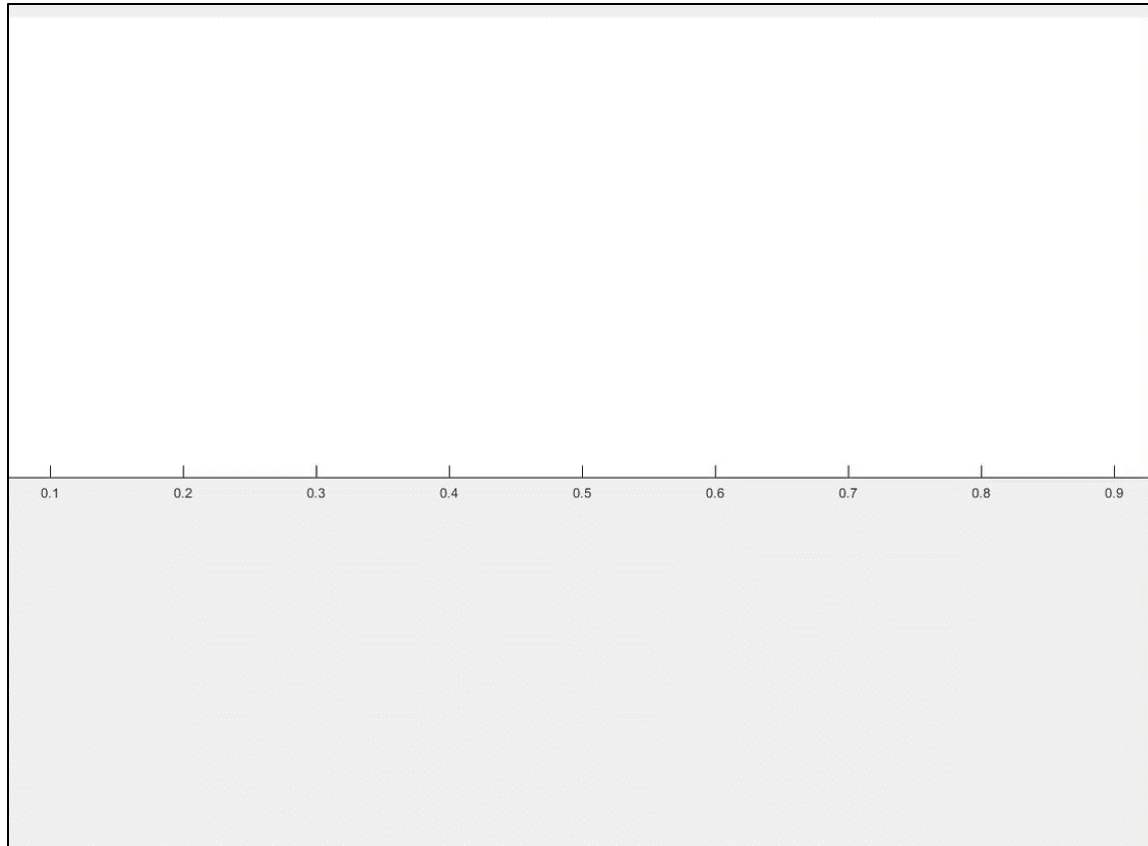
$$CR(\text{Cross - ratio}) = (A'B'C'D') = \frac{|A'C'| \times |B'D'|}{|A'D'| \times |B'C'|}$$

$$d_{i,j}(\text{distance}) = \sqrt{\frac{CR}{(CR - 1)}} l^2$$

$$v_{i,j}(\text{speed}) = \frac{d_{i,j} \times FPS}{|i - j|}$$


Methods : Module 4

Result : Estimation of vehicle speed over time



Conclusion

- **Vehicle speed and Video** are important factors in accident analysis
- Analyze the speed of a vehicle using the **vehicle's specs**, not the on-site investigation
- Analyze the Instantaneous speed of a vehicle in a video "**automatically**" and "**over time**"
- Implement functional modules
 - Single object tracking, Segmentation based on Deep Learning
 - Kalman filter technique, Cross-ratio
- Implement the functionality of all modules as a MATLAB and Toolbox
 - Plenty of useful examples on the MathWorks website
 - **Deep engagement** project with **MathWorks Korea**

MATLAB EXPO

Thank you



© 2023 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.