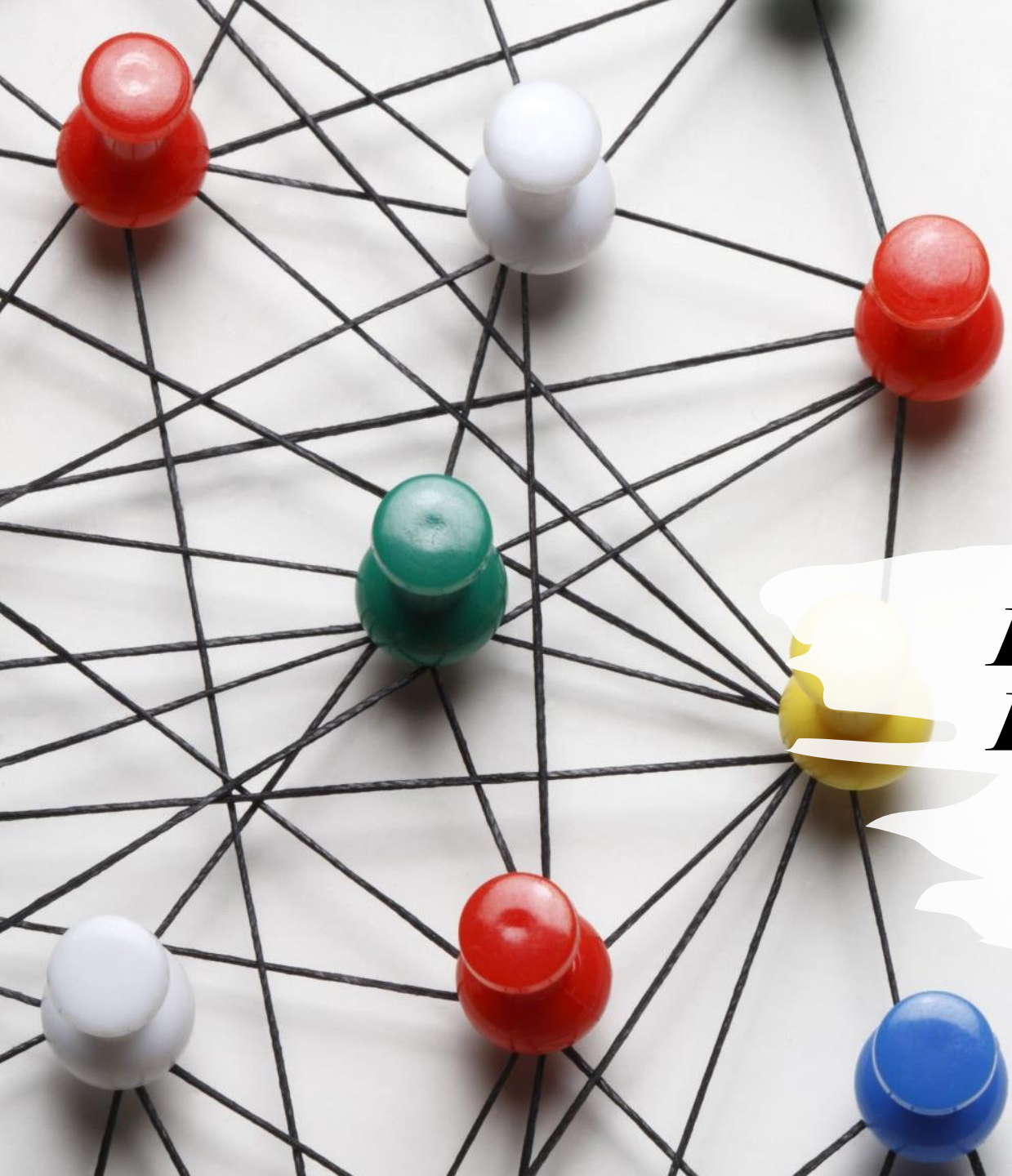


Data Analytics Series in Intelligent Transportation

ZIJUN ZHANG

Outline

- Machine learning methods have been applied into a wide range of transportation related problems to make it smarter
- Some are related to data mining using numerical data
- Some are related to computer vision problems
- High-level Traffic analysis in transportation
- Detailed-level Scene or Object Recognition

Machine Learning in Metro System Studies

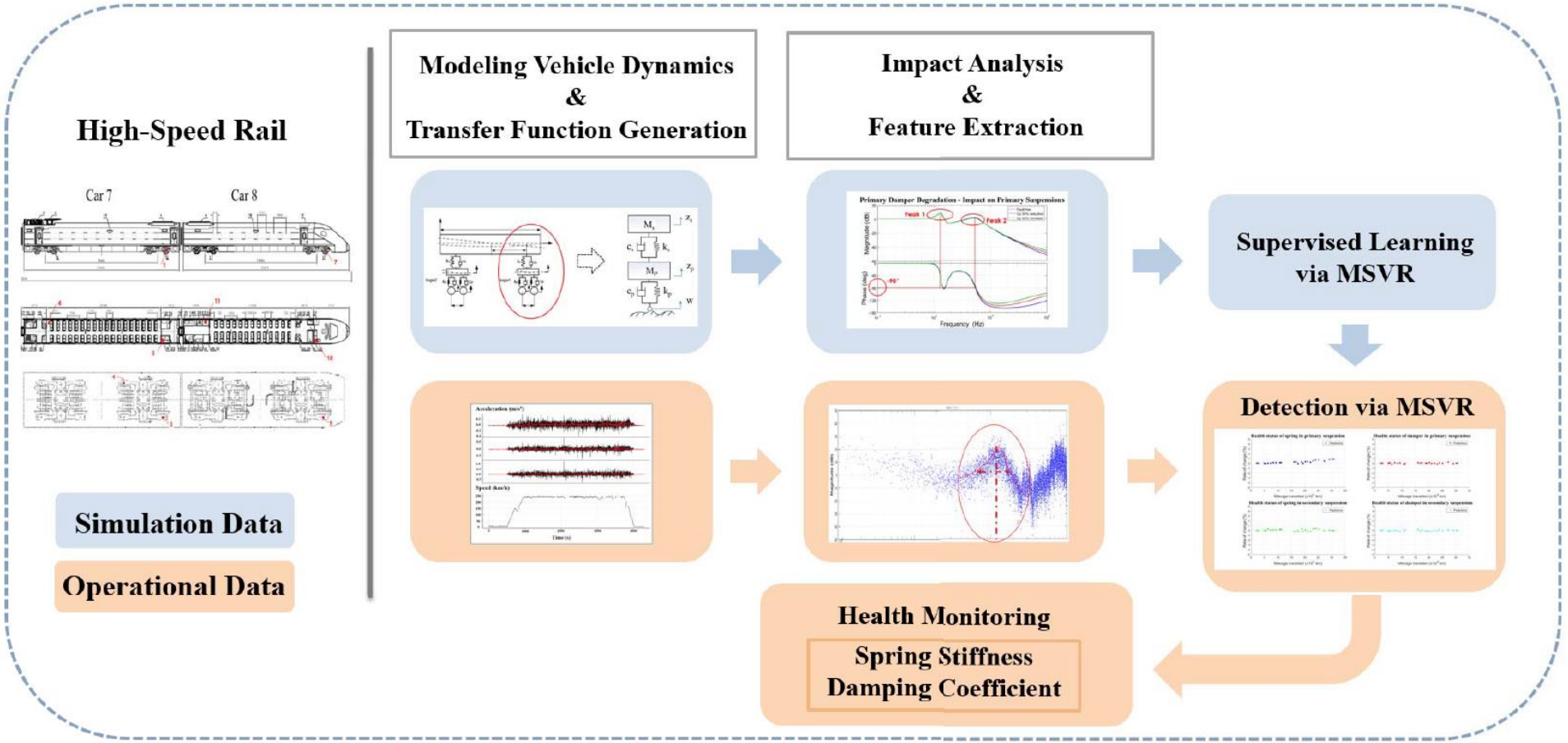
Two studies generated at Data Analytics and Computational Intelligence Laboratories at CityU HK are shared – A system monitoring and health diagnostics perspective:

1. N. Hong, L. Li, W. Yao, et al., "High-Speed Rail Suspension System Health Monitoring Using Multi-Location Vibration Data," IEEE Trans on Intelligent Transportation Systems, 2019, In press.
2. L. Zhuang, Z. Zhang, L. Wang, et al., "Automated vision inspection of rail surface cracks: A double-layer data-driven framework," Transportation Research Part C: Emerging Technologies, Vol. 92, pp. 258-277, 2018.

MSS Part 1

- A data-driven method for monitoring health status of HSR suspension system
- Vibration data are considered
- A framework integrating data-driven models and domain knowledge
- Validation via simulation and real data

MSS Part I


High-Speed Rail
Car 7
Car 8
Simulation Data
Operational Data
Modeling Vehicle Dynamics
&
Transfer Function Generation
Impact Analysis
&
Feature Extraction
Supervised Learning
via MSVR
Detection via MSVR
Health Monitoring
Spring Stiffness
Damping Coefficient

MSS Part I

Suspension System

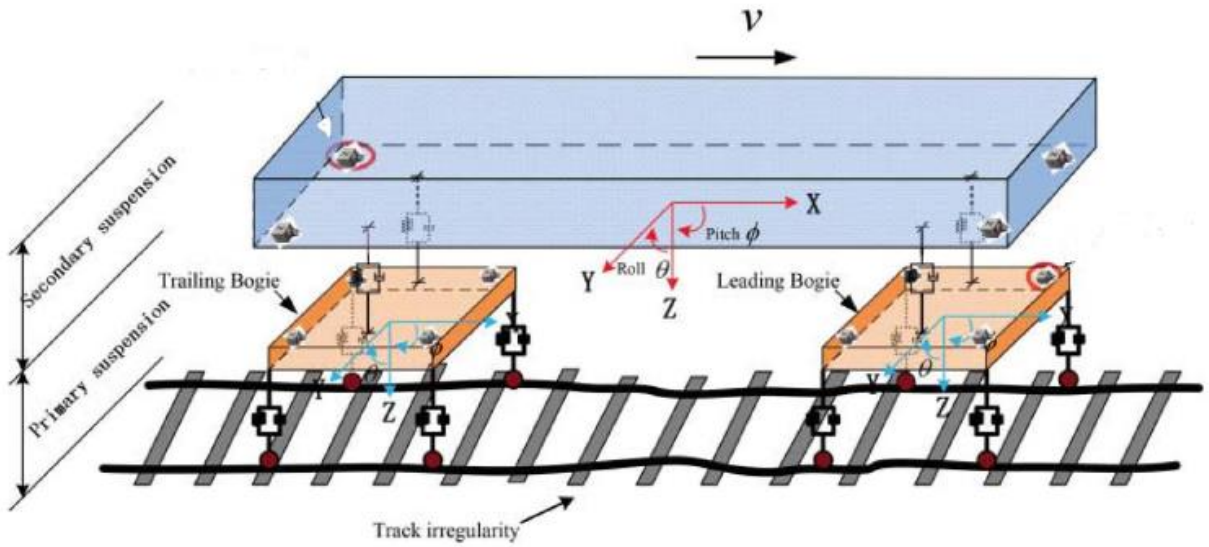


Fig. 2. Vertical suspension system of the rail vehicle.

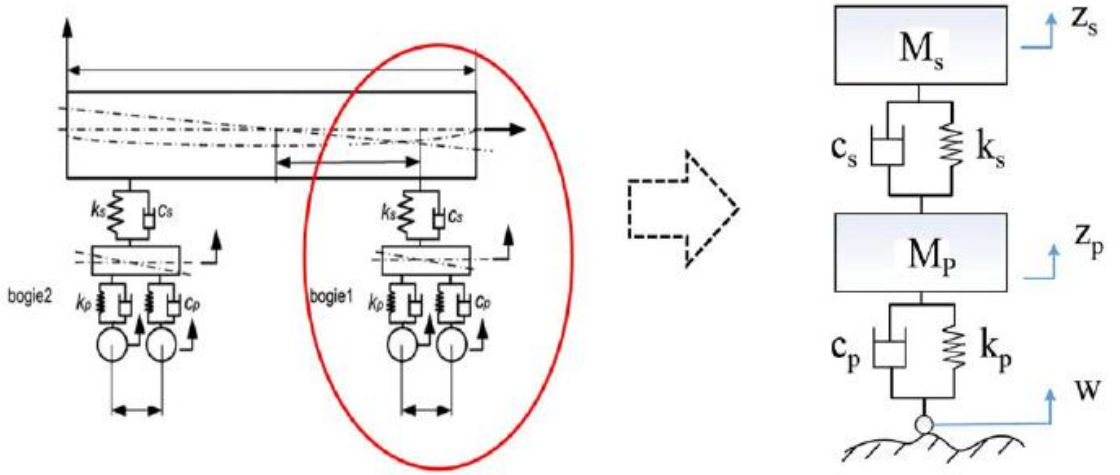


Fig. 3. Simplified 2-DOF vertical suspension system of the rail vehicle.

MSS Part I

Feature extraction

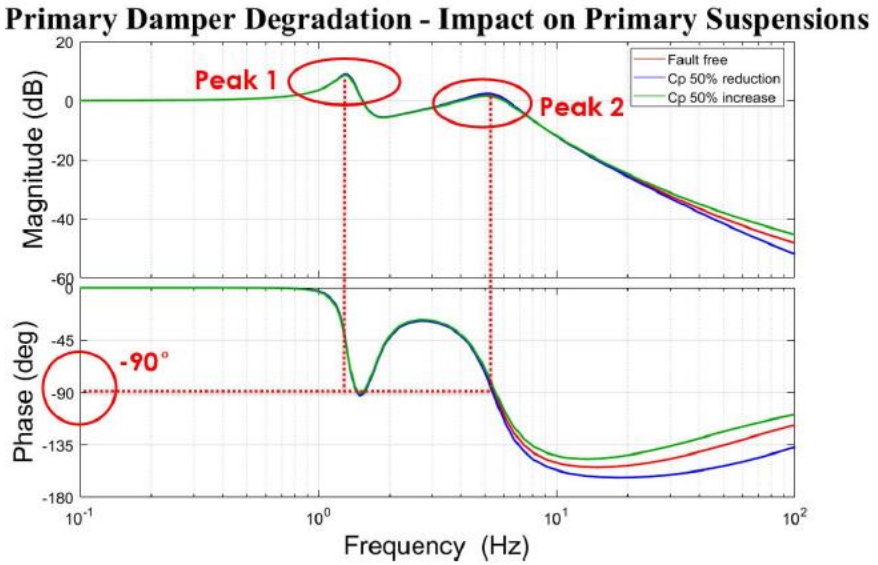


Fig. 6. Peaks at resonant frequencies of the primary suspension system.

$$x_i = \begin{bmatrix} position_i \\ width_i \\ height_i \end{bmatrix}$$

Position, Height and Width of the largest peak

MSS Part I

Once identified inputs and outputs, next match the most suitable data-driven models:

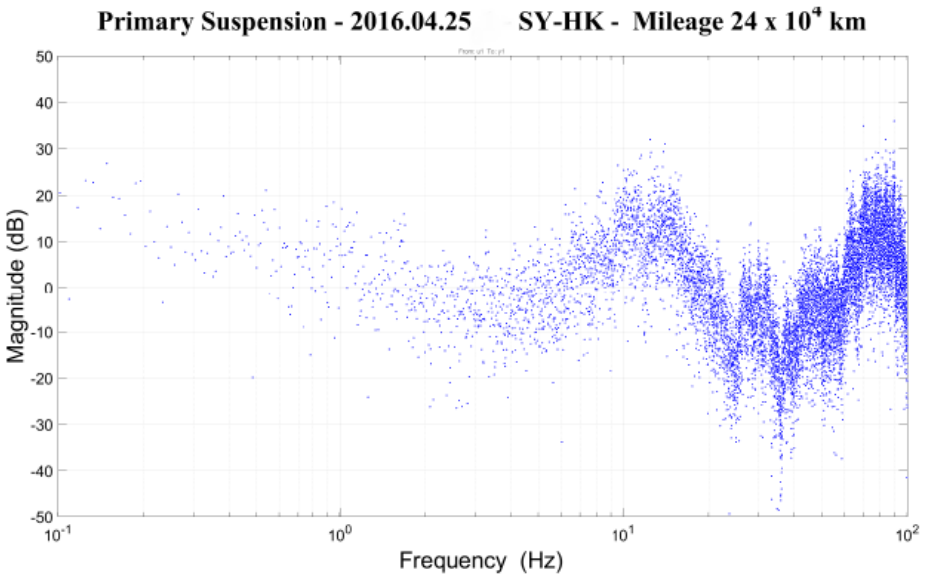
Support Vector Regression, Gaussian Regression Process, Random Forest (Not selected), Artificial Neural Networks (Not selected), Linear Regression

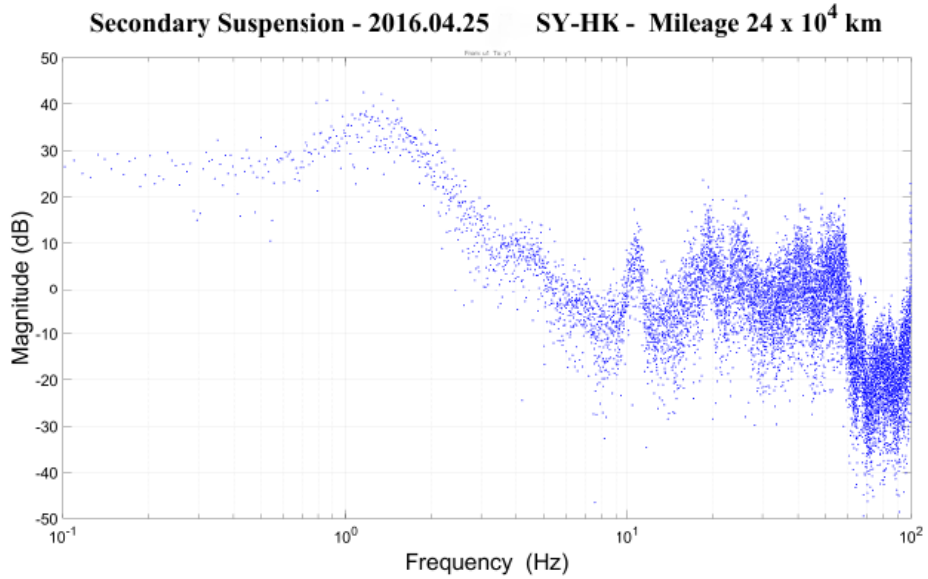
MSS Part I

PERFORMANCES OF MSVR, SVR, MV-GPR, AND MV-LR MODELS IN TERMS OF MAPE AND RMSE

Simulation Group	Output	Metrics	MSVR	SVR	MV-GPR (MEAN)	MV-LR
I	Primary Spring Stiffness	MAPE (%)	**5.89**	7.34	14.76	19.69
		RMSE	**0.0579**	0.0721	0.1438	0.1953
	Primary Damper Coefficient	MAPE (%)	**8.97**	10.29	17.26	21.71
		RMSE	**0.0871**	0.1013	0.1703	0.2151
II	Secondary Spring Stiffness	MAPE (%)	**4.37**	6.96	13.27	18.06
		RMSE	**0.0426**	0.0673	0.1311	0.1791
	Secondary Damper Coefficient	MAPE (%)	**7.53**	9.92	16.43	20.84
		RMSE	**0.0739**	0.0972	0.1632	0.2073

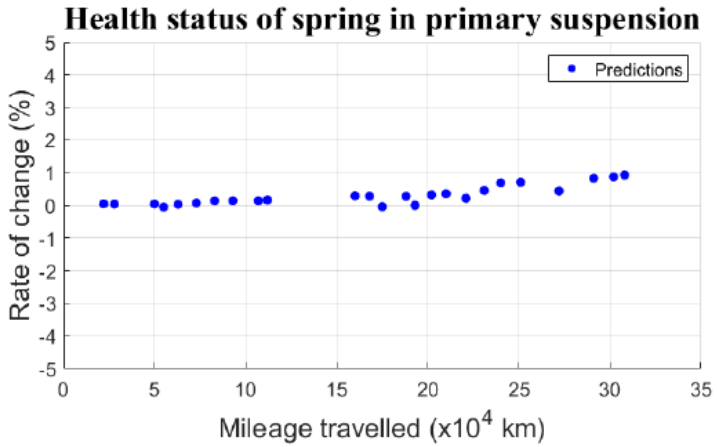
MSS Part I

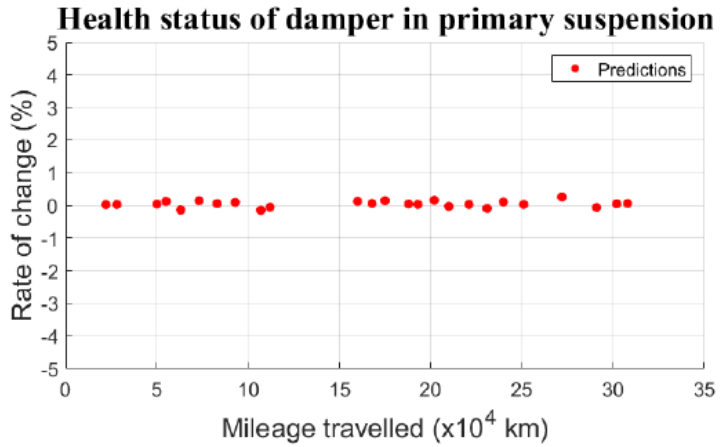


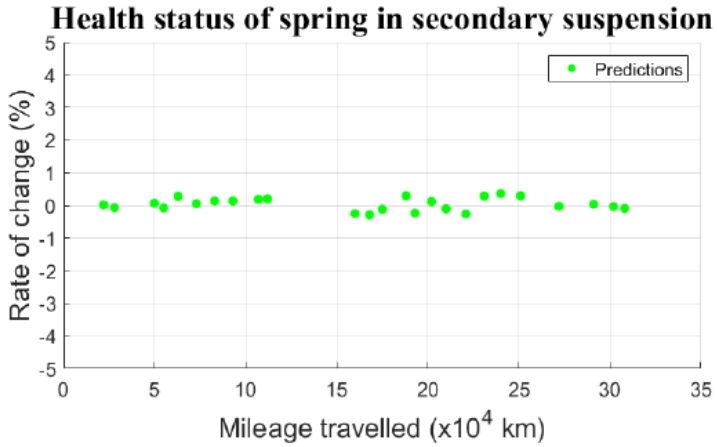


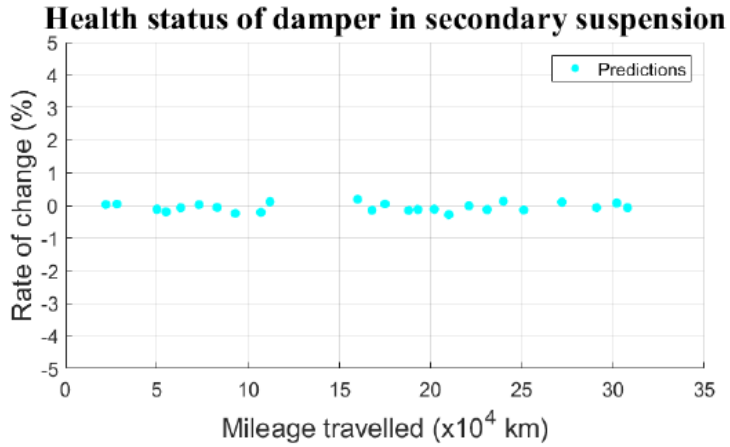
ig. 9. Samples of magnitude frequency response curves for the primary and secondary suspension system.

MSS Part I


Health status of spring in primary suspension
Predictions
Rate of change (%)
Mileage travelled (x10^4 km)


Health status of damper in primary suspension
Predictions
Rate of change (%)
Mileage travelled (x10^4 km)


Health status of spring in secondary suspension
Predictions
Rate of change (%)
Mileage travelled (x10^4 km)


Health status of damper in secondary suspension
Predictions
Rate of change (%)
Mileage travelled (x10^4 km)

MSS Part II

- A image based data-driven study
- Attack the rail track crack detection problem
- Traditional feature extraction + classifier taken by sliding window
- We detect both position and shape of the crack

MSS Part II

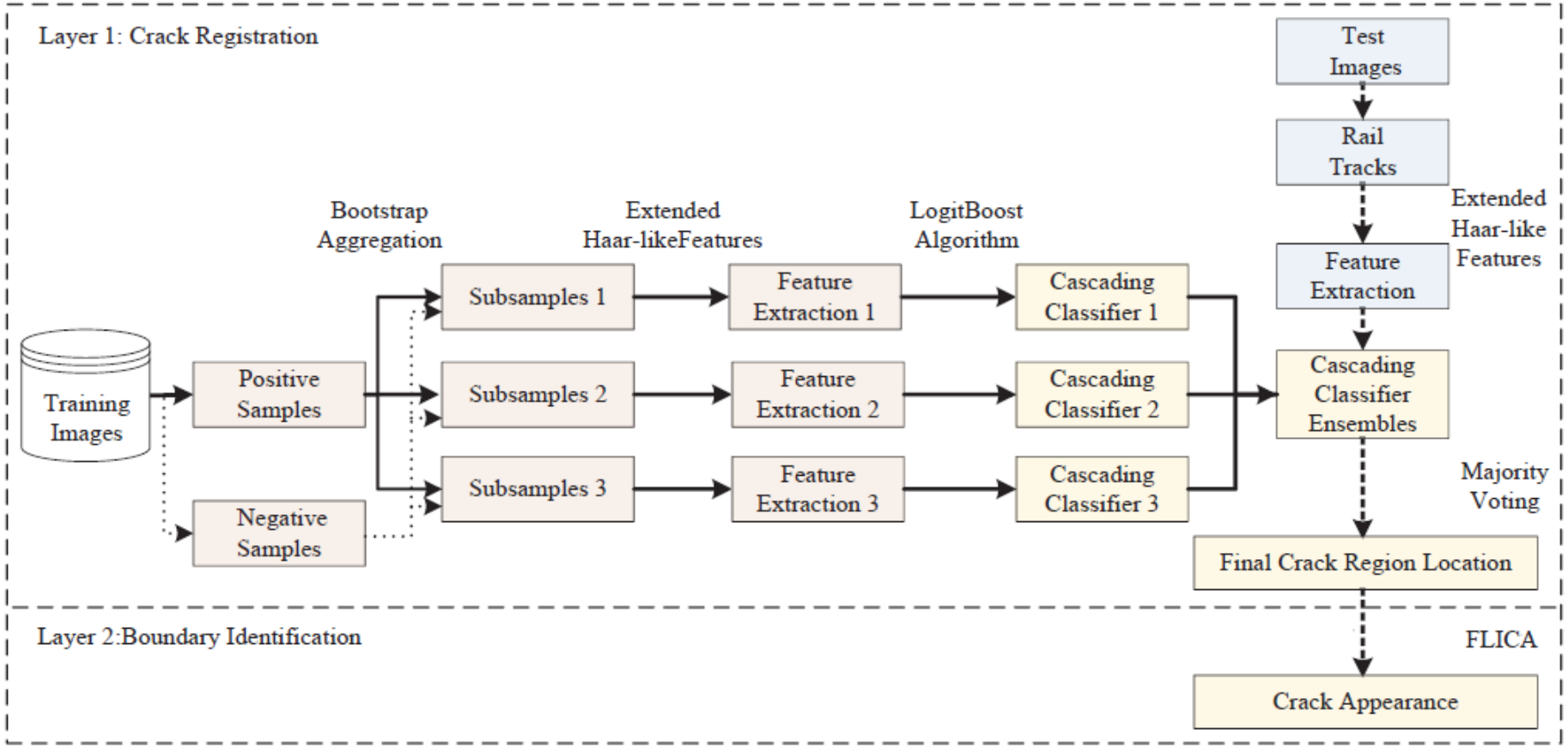


Fig. 1. The schematic diagram of the bi-layer data-driven crack detection framework.

MSS Part II

Feature extraction

$$ii(x,y) = \sum_{x' \leqslant x, y' \leqslant y} i(x',y')$$

$$ii_R(x,y) = \sum_{x' \leqslant x,\ x' \leqslant x-|y-y'|} i(x',\ y')$$

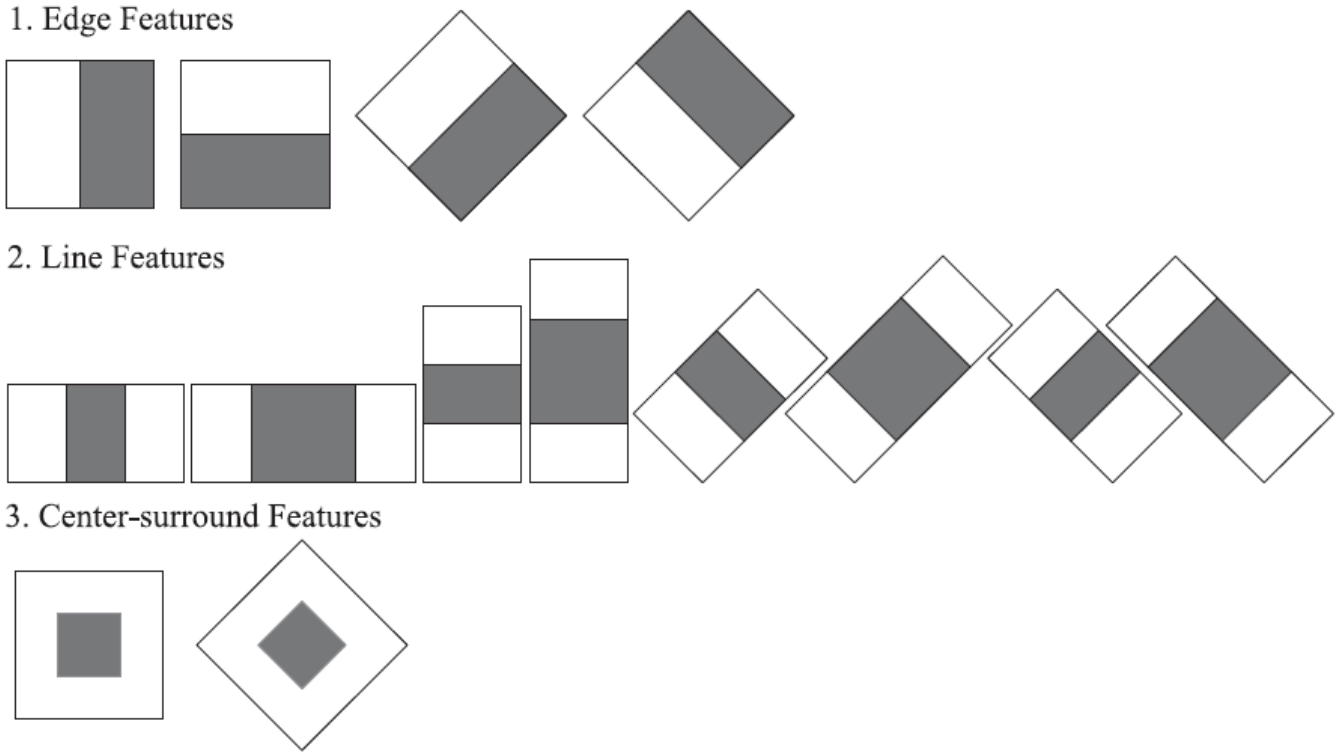


Fig. 2. An illustration of extended Haar-like features.

MSS Part II

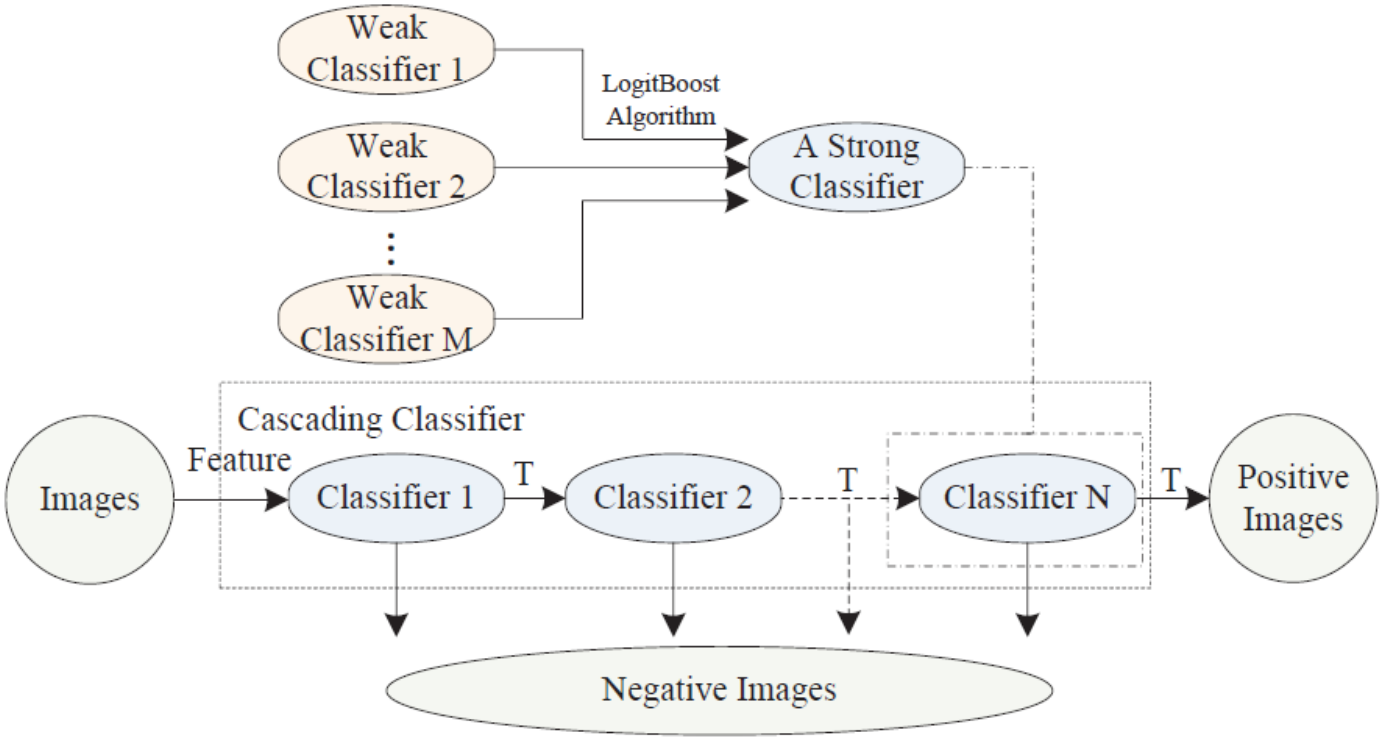


Fig. 4. The structure of an individual cascading classifier.

MSS Part II

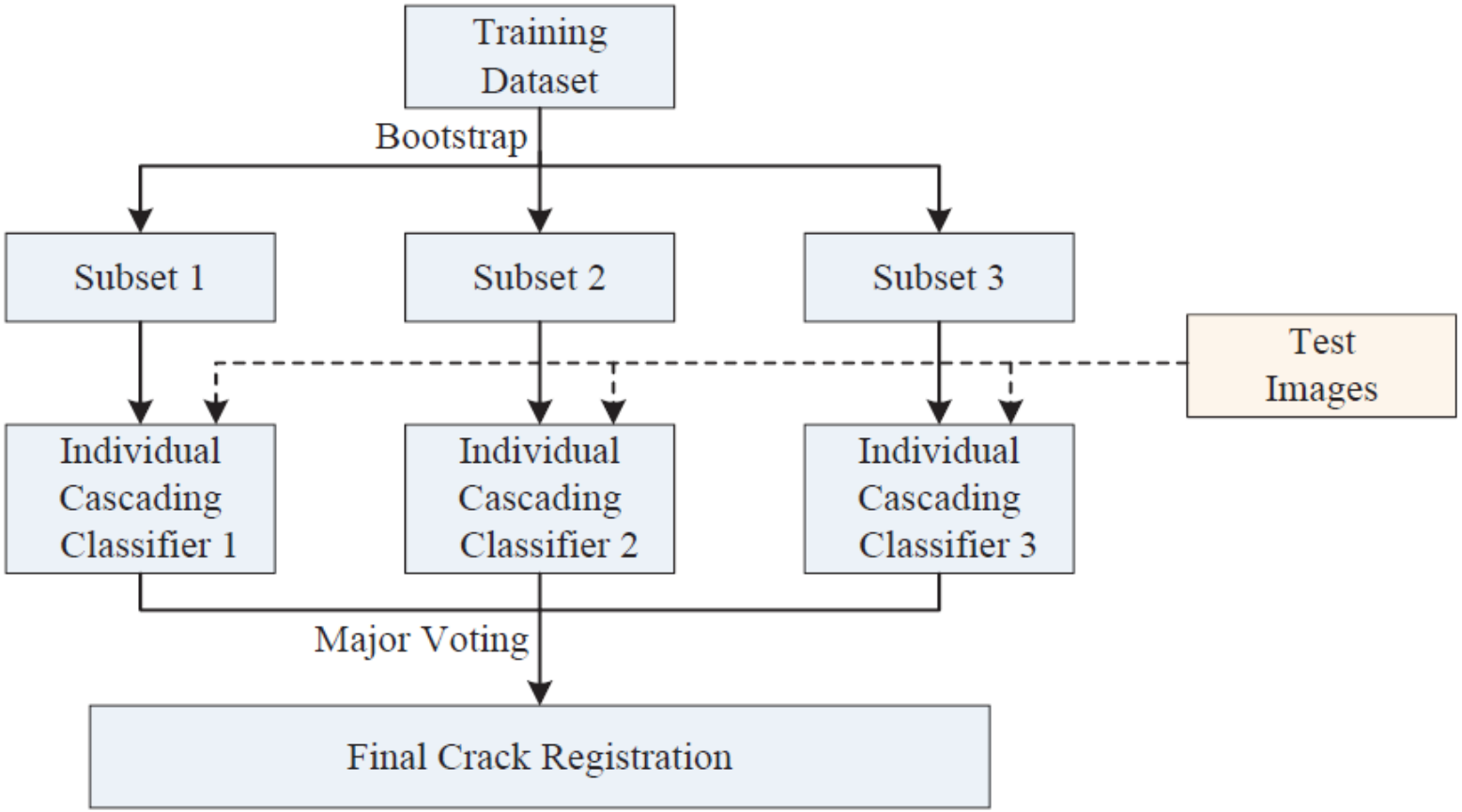


Fig. 5. The structure of the proposed cascading classifier ensemble.

MSS Part II

Crack boundary detection: FLICA

$$\mathbf{F}_i = [l_i, a_i, b_i, \mu_{l_i}, \mu_{a_i}, \mu_{b_i}, \sigma_{l_i}^2, \sigma_{a_i}^2, \sigma_{b_i}^2, t_i, e_i]^T$$

CIELAB, mean variance of neighbors, adaptive local threshold, Canny-edge detection

MSS Part II

Adaptive local threshold

$$T_{local} = \mu_0 - C$$

MSS Part II

Canny-edge detection

$$G(x,y) = \frac{1}{2\pi\sigma} e^{\left(-\frac{x^2+y^2}{\sigma^2}\right)}$$

$$S_x = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}, \quad S_y = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

$$Q_x(i,j) = \tfrac{1}{2}[f(i,j+1) - f(i,j) + f(i+1,j+1) - f(i+1,j)]$$

$$Q_y(i,j) = \tfrac{1}{2}[f(i,j) - f(i+1,j) + f(i,j+1) - f(i+1,j+1)]$$

$$M(i,j) = \sqrt{Q_x(i,j)^2 + Q_y(i,j)^2}$$

$$\theta(i,j) = \arctan[Q_y(i,j), Q_x(i,j)]$$

$$e(i,j) = \begin{cases} 0, & non-edge \\ 1, & edge \end{cases}$$

MSS Part II

Algorithm 2 (*FLICA crack boundary generation*).

Input: Image;
Maximum iteration: *maxIter*;
Output: Boundaries of the crack;
Initialize K cluster centers

$$\mathbf{F}_k = [l_k, a_k, b_k, \mu_{l_k}, \mu_{a_k}, \mu_{b_k}, \sigma^2_{l_k}, \sigma^2_{a_k}, \sigma^2_{b_k}, t_k, e_k]^T;$$

Set label $l(i) \leftarrow$ -1 for each pixel i;
Set distance $d(i) \leftarrow \infty$ for each pixel i;
Set $numIter \leftarrow 0$;
while $numIter < maxIter$ **do**
  for *each* $\mathbf{F}_k$ **do**
    for each pixel $\mathbf{F}_i$ in the input image **do**
      Compute the distance D_{ki} between $\mathbf{F}_k$ and $\mathbf{F}_i$;
      if $D_{ki} < d(i)$ **then**
        Set $d(i) = D_{ki}$;
        Set $l(i) = k$;
      end
    end
  end
  Update cluster centers;
  Set $numIter = numIter + 1$;
end

$$\mathbf{F}'_i = \frac{\mathbf{F}_i - \min(\mathbf{F}_i)}{\max(\mathbf{F}_i) - \min(\mathbf{F}_i)}$$

$$D_{ki} = \sqrt{w_1 \|\mathbf{C}_i - \mathbf{C}_k\|^2 + w_2 \|\mathbf{M}_i - \mathbf{M}_k\|^2 + w_3 \|\mathbf{V}_i - \mathbf{V}_k\|^2 + w_4 \|\mathbf{T}_i - \mathbf{T}_k\|^2 + w_5 \|\mathbf{E}_i - \mathbf{E}_k\|^2}$$

$$\sum_{k=1}^{5} w_k = 1$$

MSS Part II

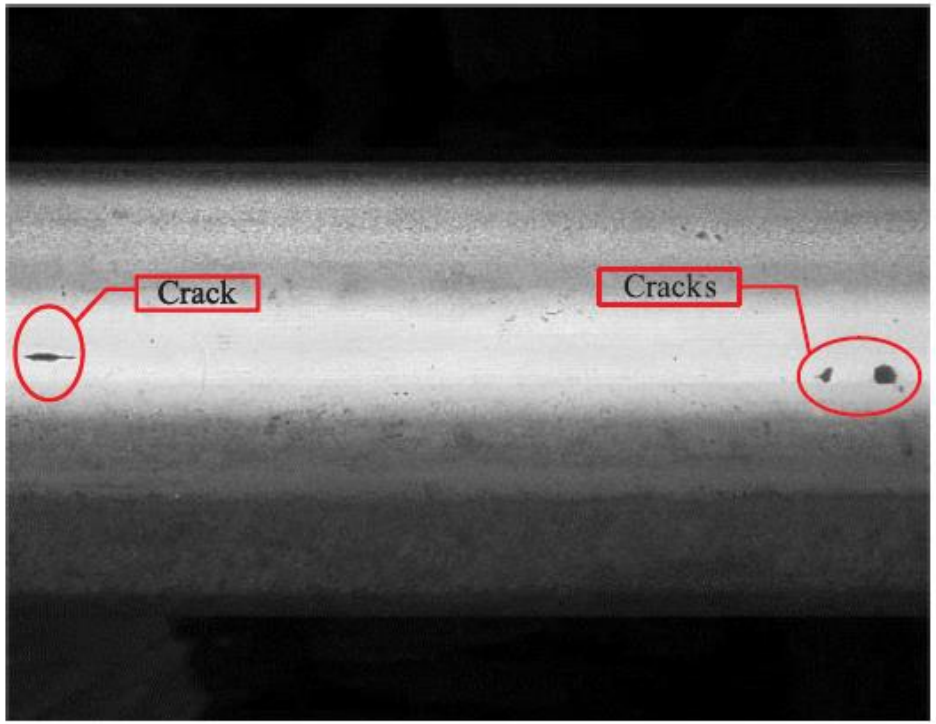
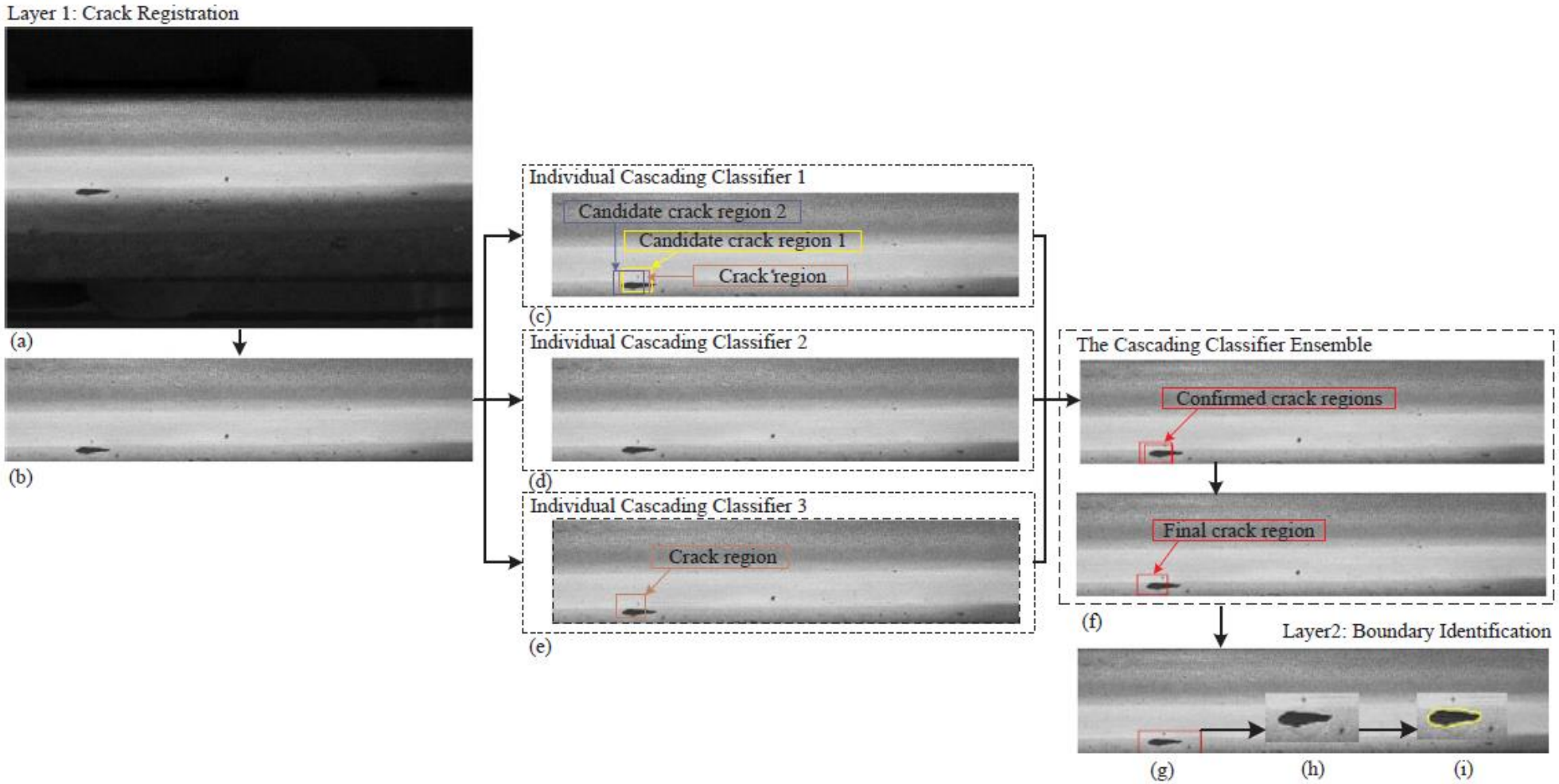


Fig. 6. Illustrative examples of rail images with 512 × 512 size.

MSS Part II


Layer 1: Crack Registration
(a)
(b)
Individual Cascading Classifier 1
Candidate crack region 2
Candidate crack region 1
Crack region
(c)
Individual Cascading Classifier 2
(d)
Individual Cascading Classifier 3
Crack region
(e)
The Cascading Classifier Ensemble
Confirmed crack regions
Final crack region
(f)
Layer2: Boundary Identification
(g)
(h)
(i)

MSS Part II

Table 4
Performance of different algorithms.

Algorithm	Crack registration			Crack boundary identification		
	Applicability	*Precision*	*Recall*	Applicability	*Precision*	*Recall*
The proposed BDF	Yes	85.71%	96.00%	Yes	65.97%	67.65%
The VDS	Yes	50.00%	60.00%	No	N.A.	N.A.
The geometric approach	Yes	83.34%	40.00%	No	N.A.	N.A.
FCN-VGG	No	N.A.	N.A.	Yes	21.91%	17.33%
U-net	No	N.A.	N.A.	No.	N.A.	N.A.
Unet-VGG	No	N.A.	N.A.	Yes	40.39%	48.09%

Vehicle System Studies

- Road scene or object detection (Vehicle recognize the environment)
- Vehicle characteristics recognition (System recognize vehicle)

Two articles referenced:

1. Y. Xu, G. Yu, X. Wu, et al., "An Enhanced Viola-Jones Vehicle Detection Method From Unmanned Aerial Vehicles Imagery," IEEE Transactions on Intelligent Transportation Systems, Vol. 18, No. 7, pp. 1845-1856, 2017.

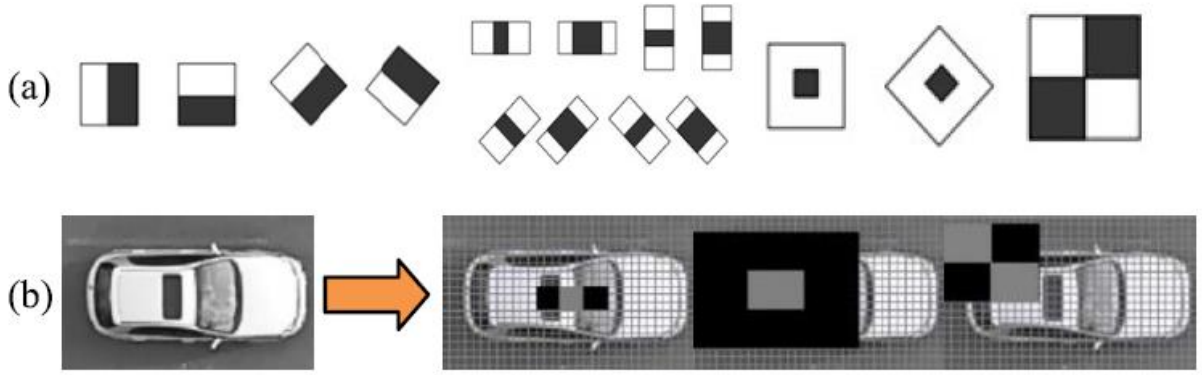
2. L. Xie, T. Ahmad, L. Jin, et al., "A New CNN-Based Method for Multi-Directional Car License Plate Detection," IEEE Transactions on Intelligent Transportation Systems, Vol. 19, No. 2, pp. 507-517, 2018

VSS Part I

- A similar Haar-like feature + classification framework
- To detect vehicles from UAV images

VSS Part I

Haar-like feature extraction



VSS Part I

Fig. 2. Training samples images. (a) Positive samples; (b) Negative samples.

VSS Part I

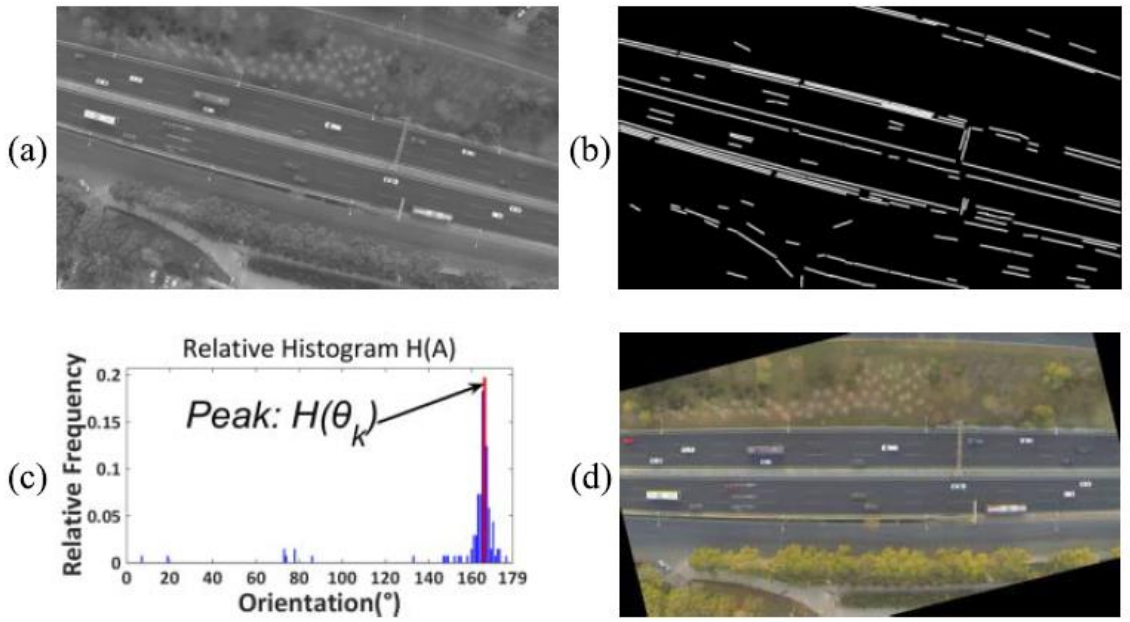


Fig. 3. (a) Grayscale image; (b) Line segments detection; (c) Relative histogram (the red bin represents the highest rectangle in relative frequency histogram.); (d) Rotated image.

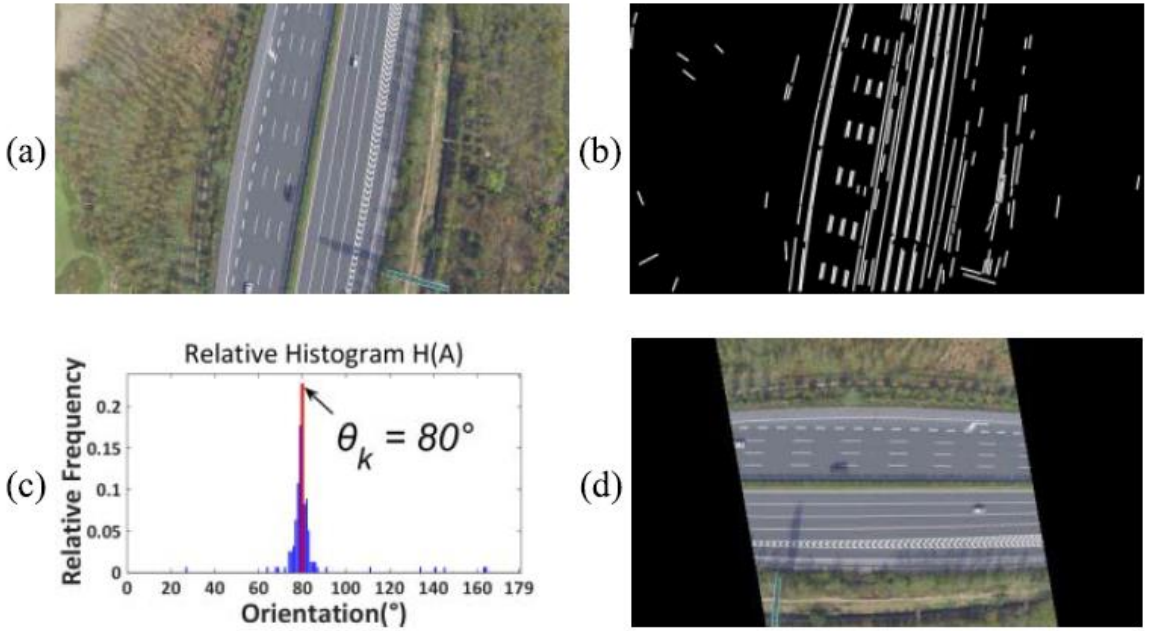


Fig. 5. Suburban road. (a) Color image; (b) Line segments detection using LSD; (c) Relative histogram; (d) Rotated image.

VSS Part I

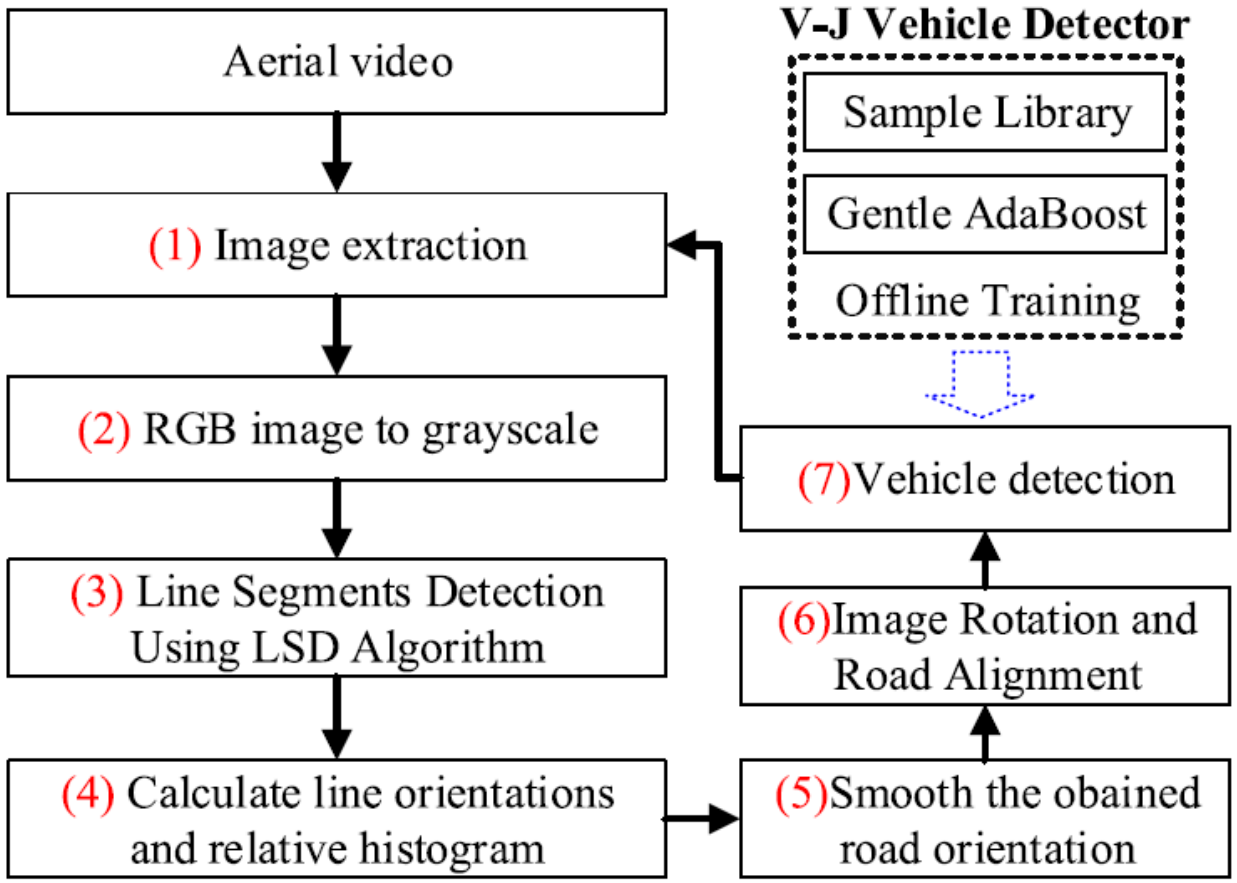
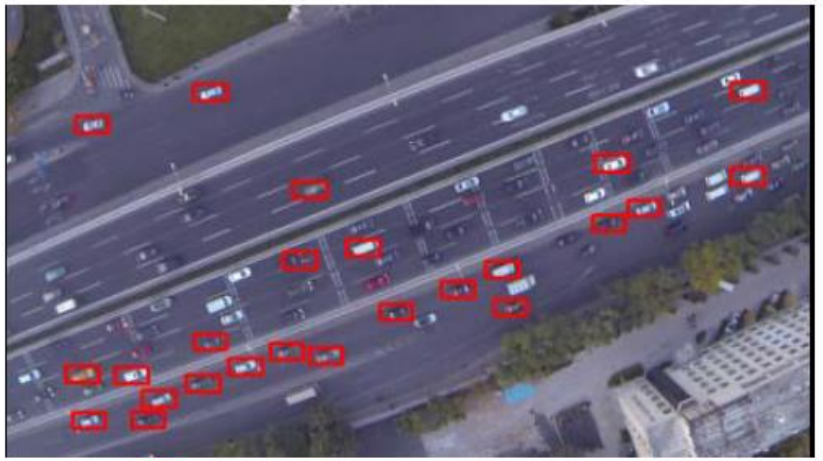


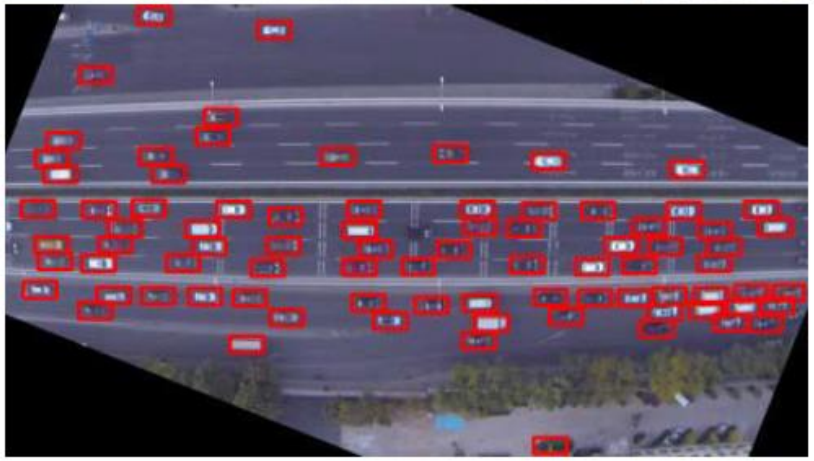
Fig. 6. The frame of the enhanced V-J method.

VSS Part I

(a)



(b)

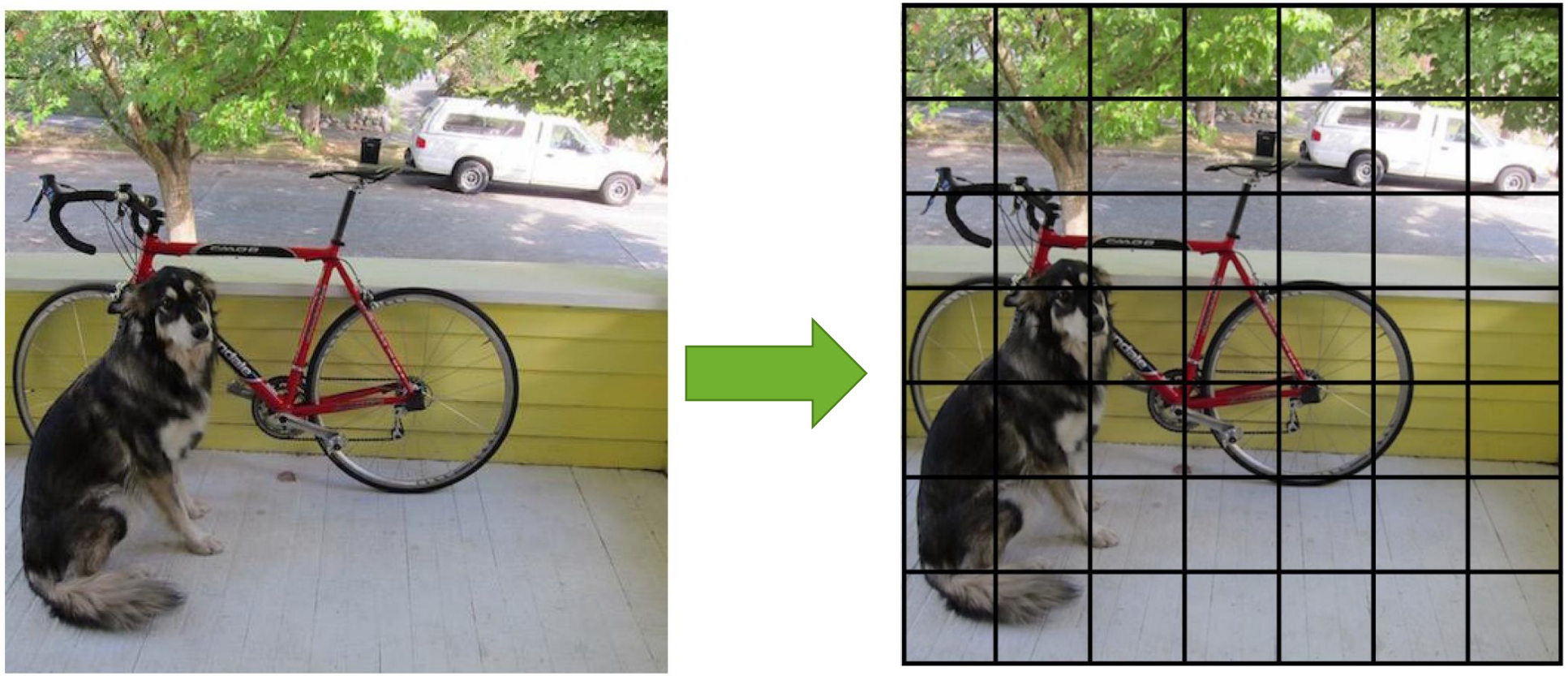


VSS Part II

A YOLO based method for automatically detect the license plate of vehicles

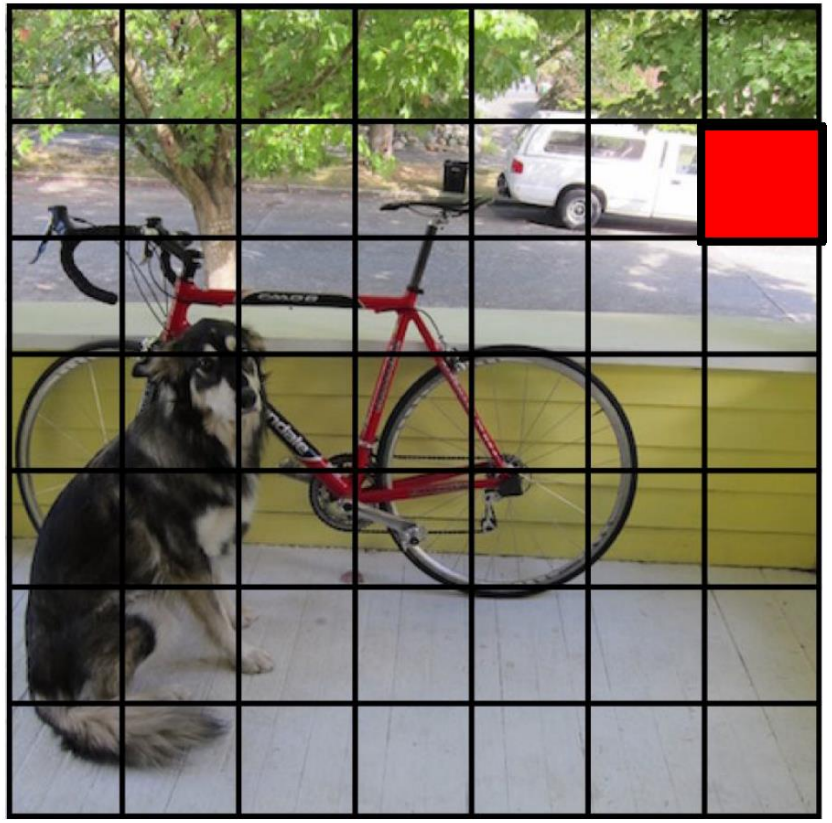
Before we go to its application details, let's get some basics of YOLO

YOLO

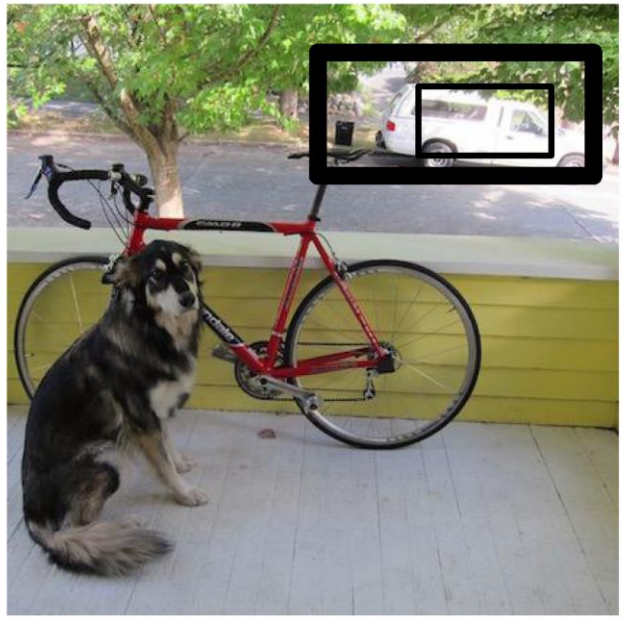


YOLO

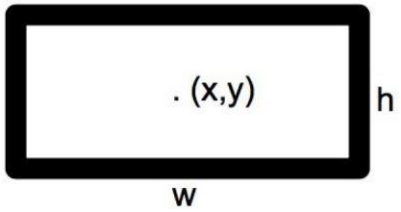
Predict the value of bounding box



B = 2

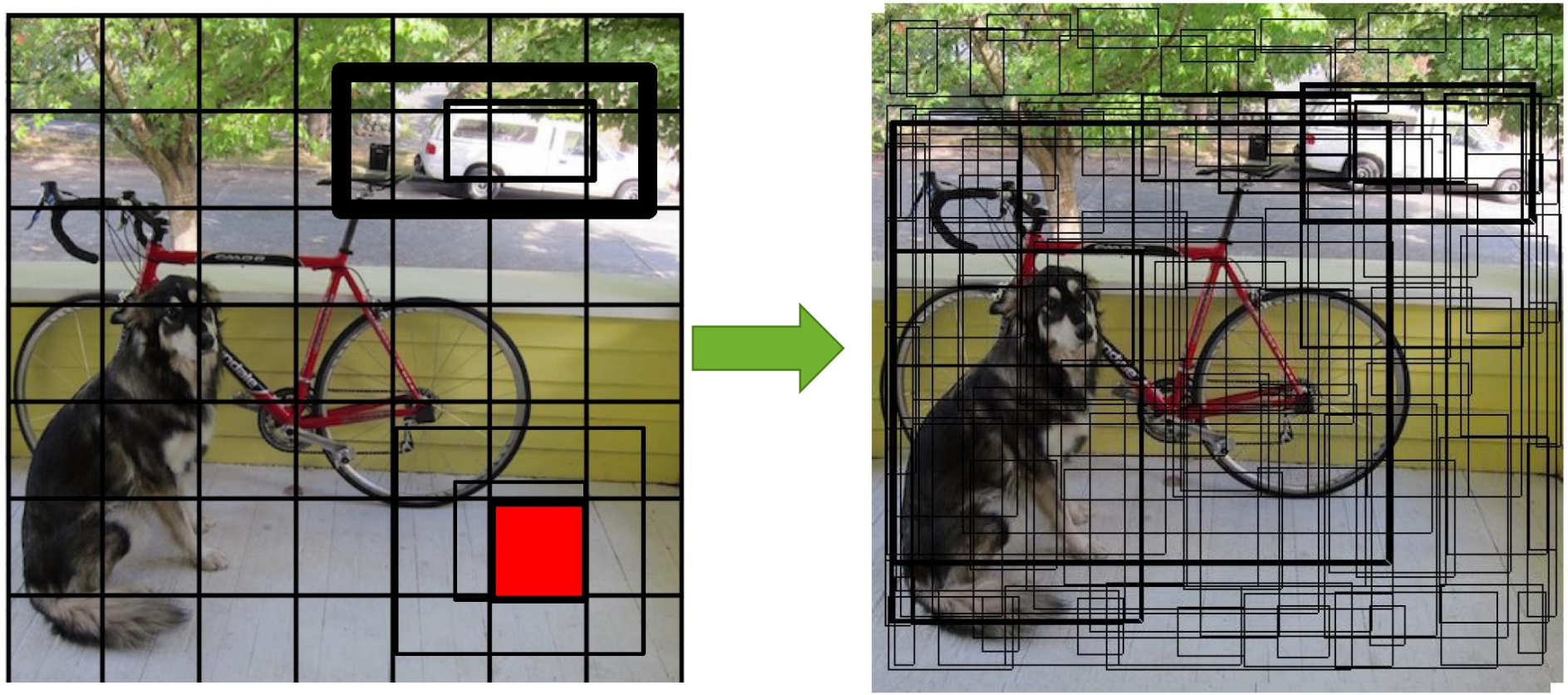


each box predict:

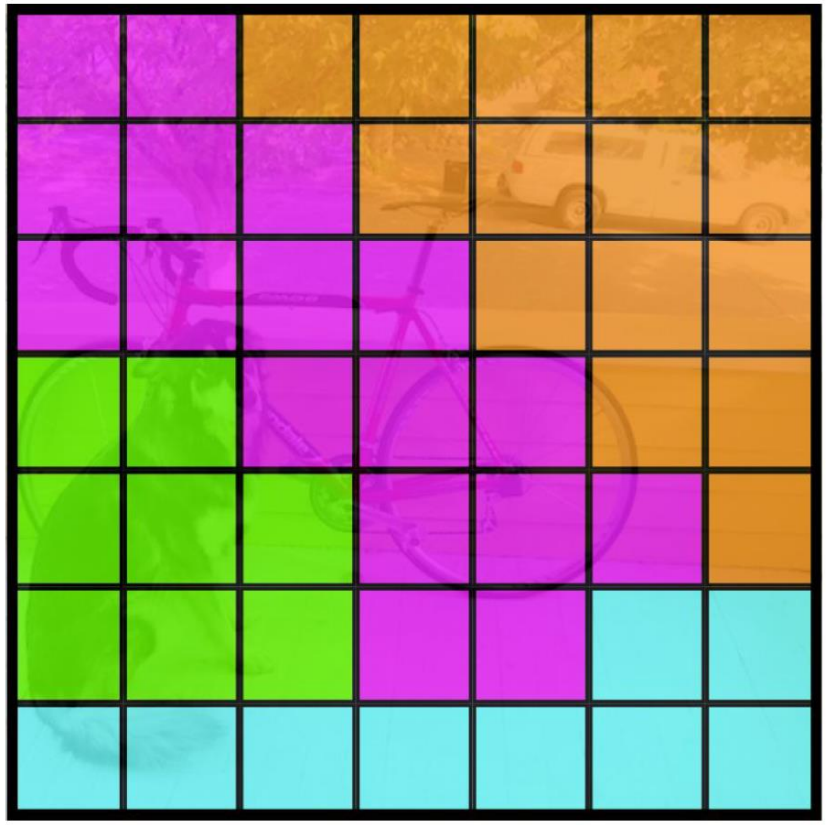


P(Object): probability that the box contains an object

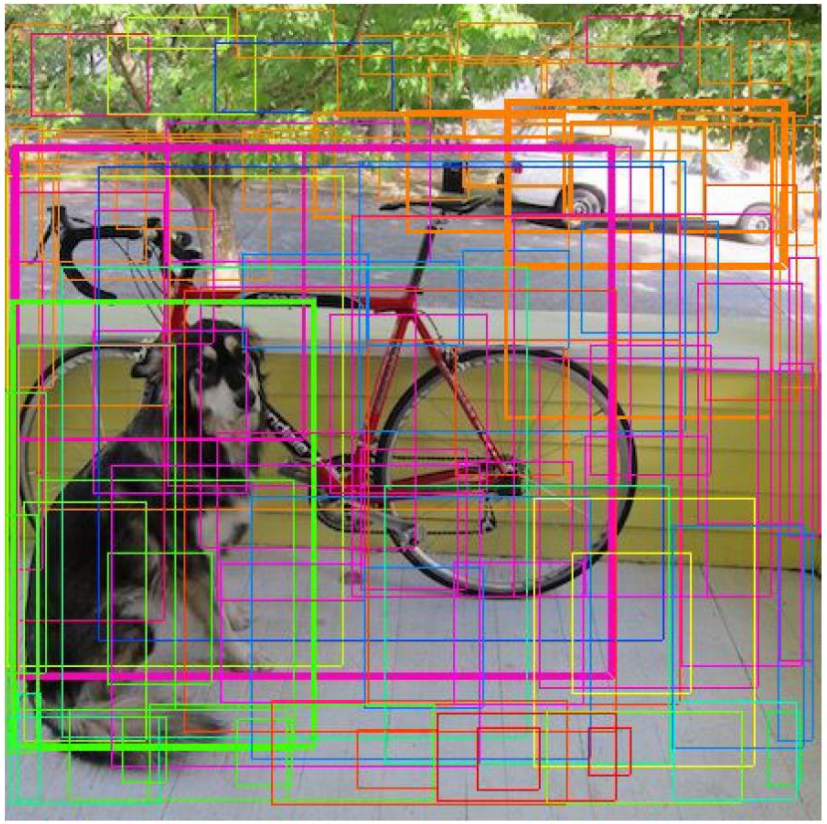
YOLO



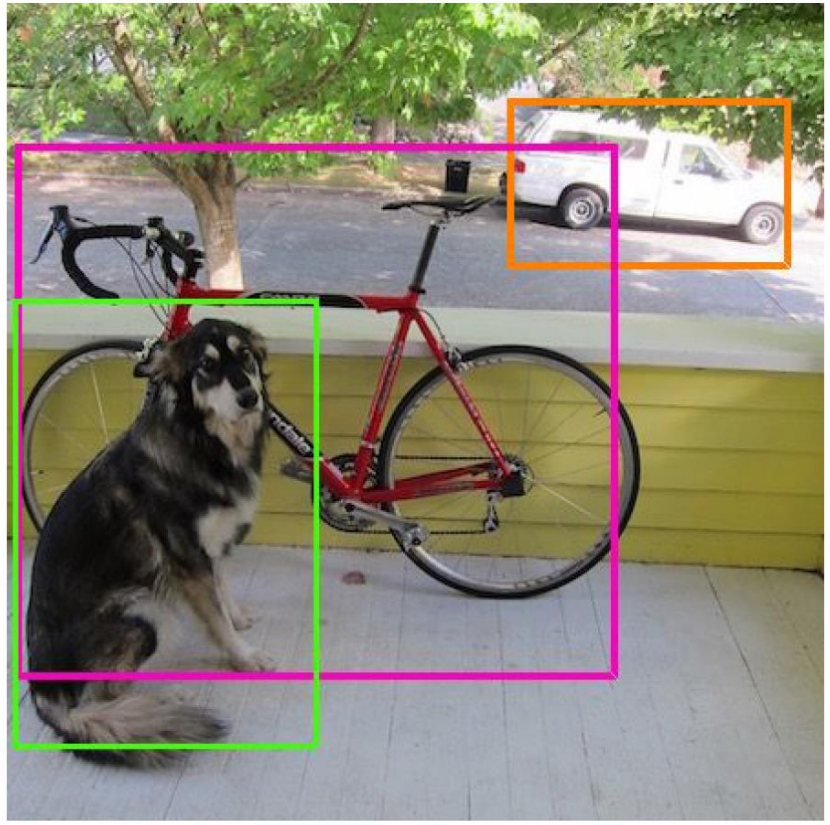
YOLO



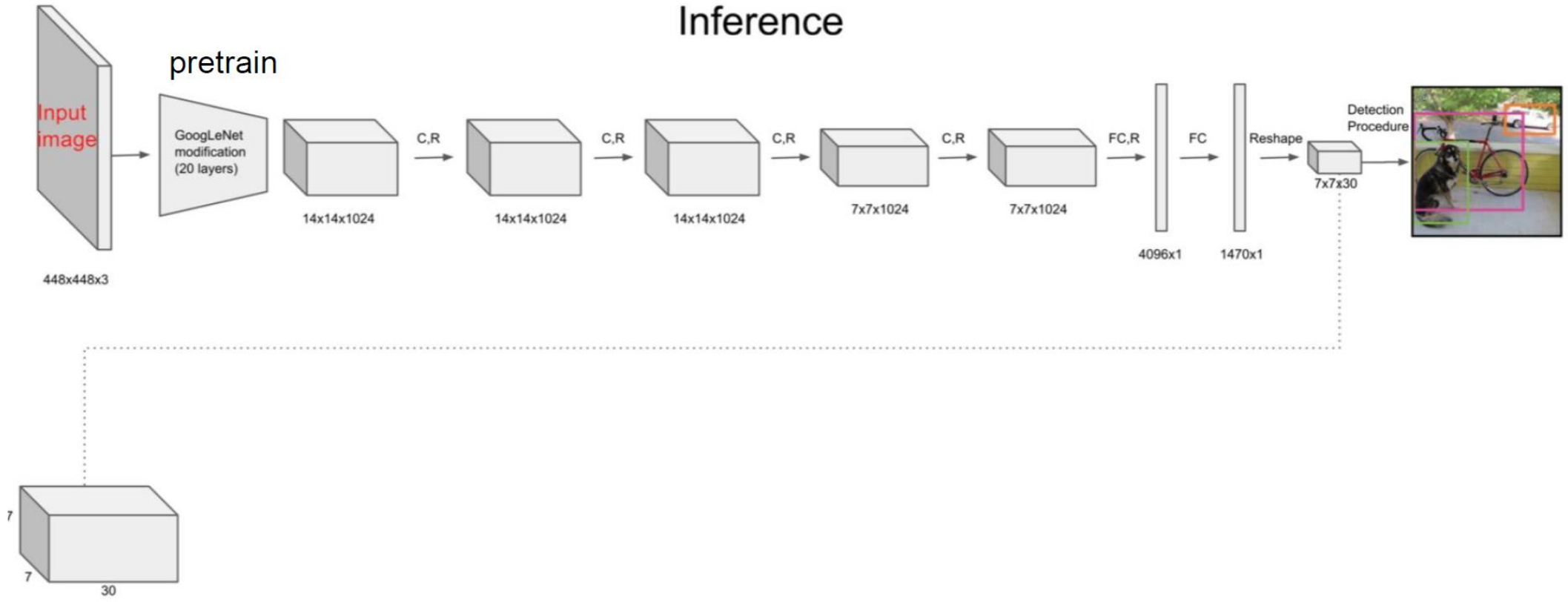
YOLO



P(class|Object) * P(Object)
=P(class)



YOLO


Inference
pretrain
Input image
448x448x3
GoogLeNet modification (20 layers)
14x14x1024
C,R
14x14x1024
C,R
14x14x1024
C,R
7x7x1024
C,R
7x7x1024
FC,R
4096x1
FC
1470x1
Reshape
7x7x30
Detection Procedure
7
7
30

YOLO

loss function:

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

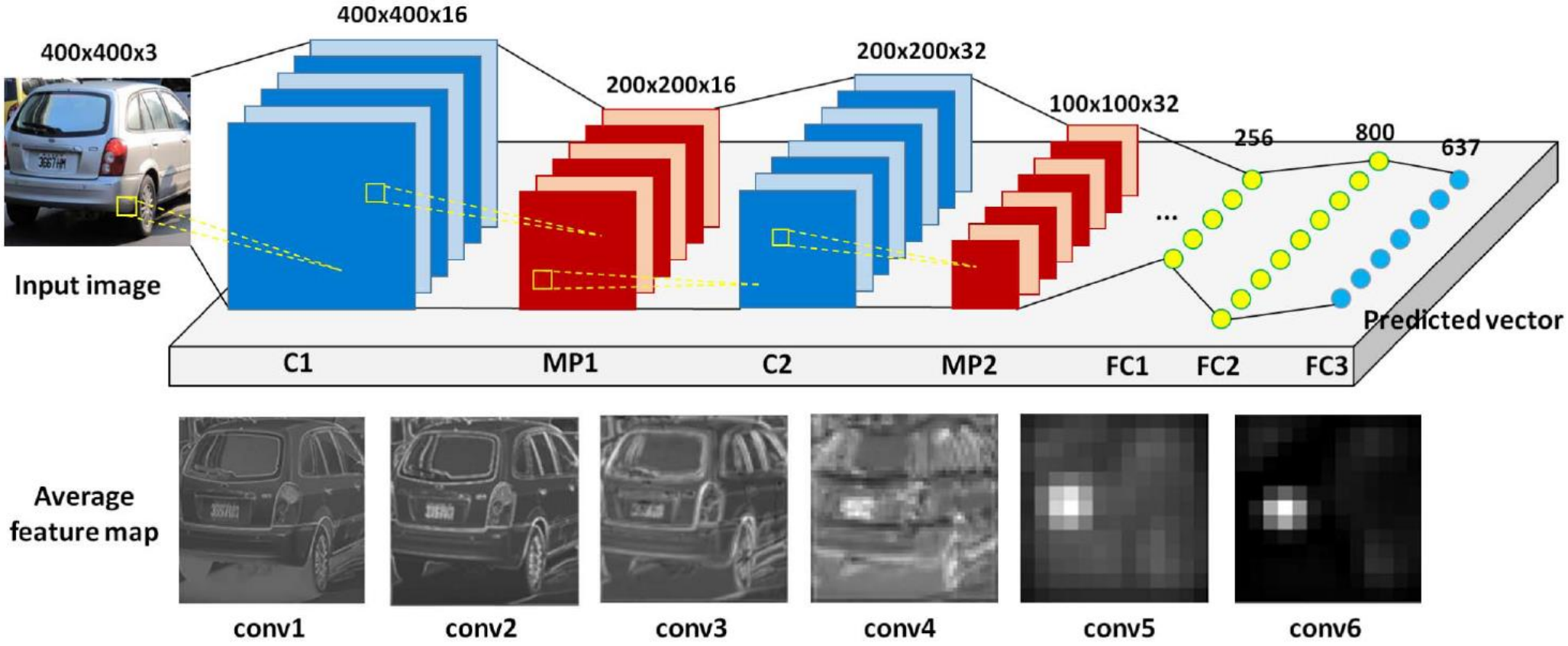
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

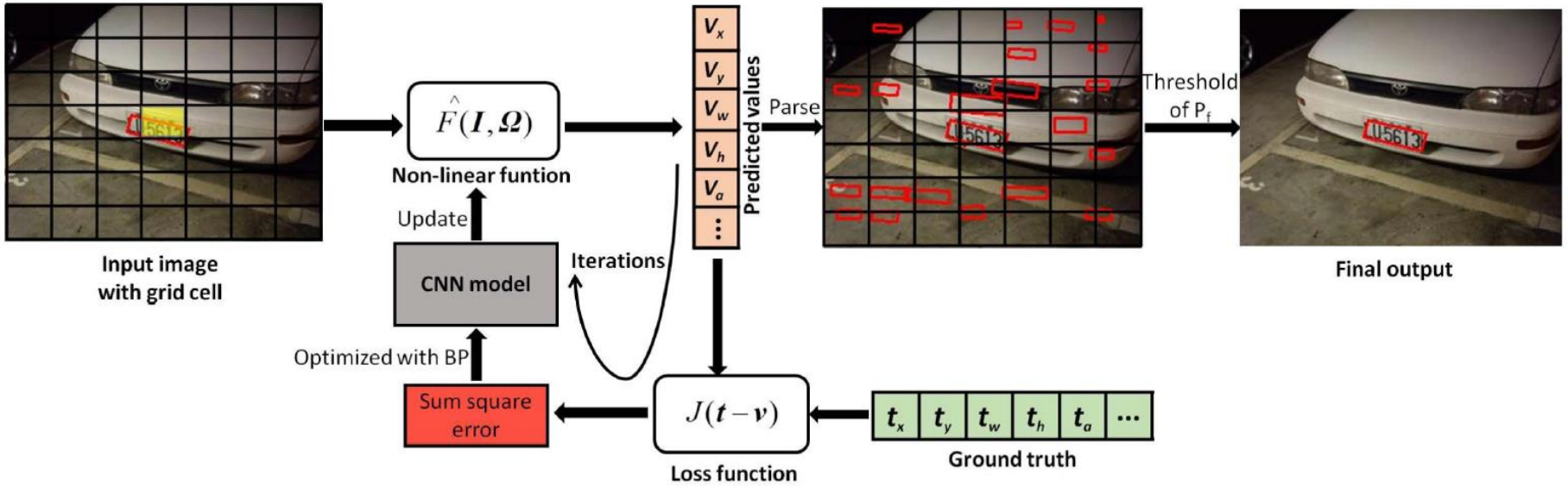
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

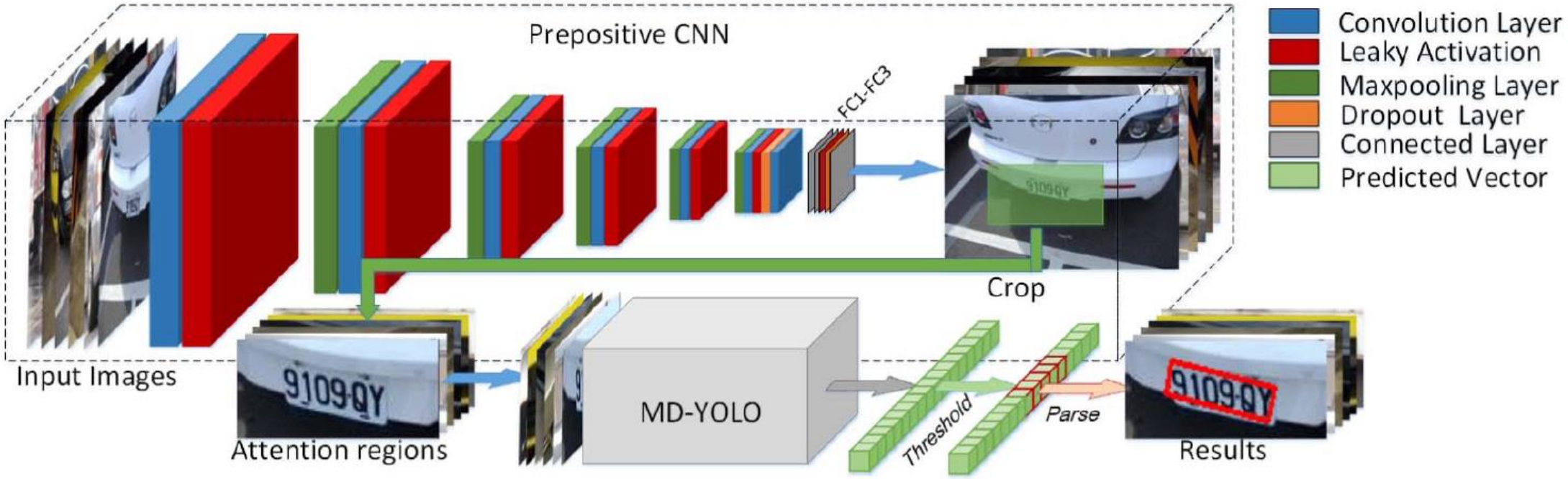
VSS Part II


400x400x3
400x400x16
200x200x16
200x200x32
100x100x32
256
800
637
...
Input image
Predicted vector
C1
MP1
C2
MP2
FC1
FC2
FC3
Average feature map
conv1
conv2
conv3
conv4
conv5
conv6

VSS Part II


Input image
with grid cell
$\hat{F}(I,\Omega)$
Non-linear funtion
Update
CNN model
Optimized with BP
Sum square
error
Iterations
V_x
V_y
V_w
V_h
V_a
⋮
Predicted values
Parse
Threshold
of P_f
Final output
$J(t-v)$
Loss function
t_x t_y t_w t_h t_a ⋯
Ground truth

VSS Part II


Prepositive CNN
FC1-FC3
Crop
Input Images
Attention regions
MD-YOLO
Threshold
Parse
Results
Convolution Layer
Leaky Activation
Maxpooling Layer
Dropout Layer
Connected Layer
Predicted Vector

VSS Part II

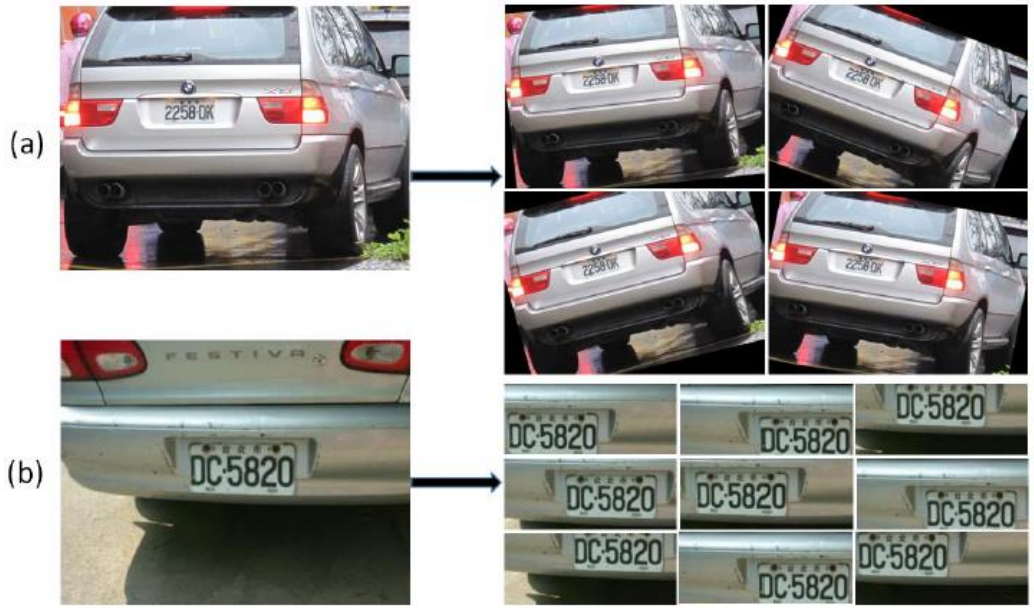
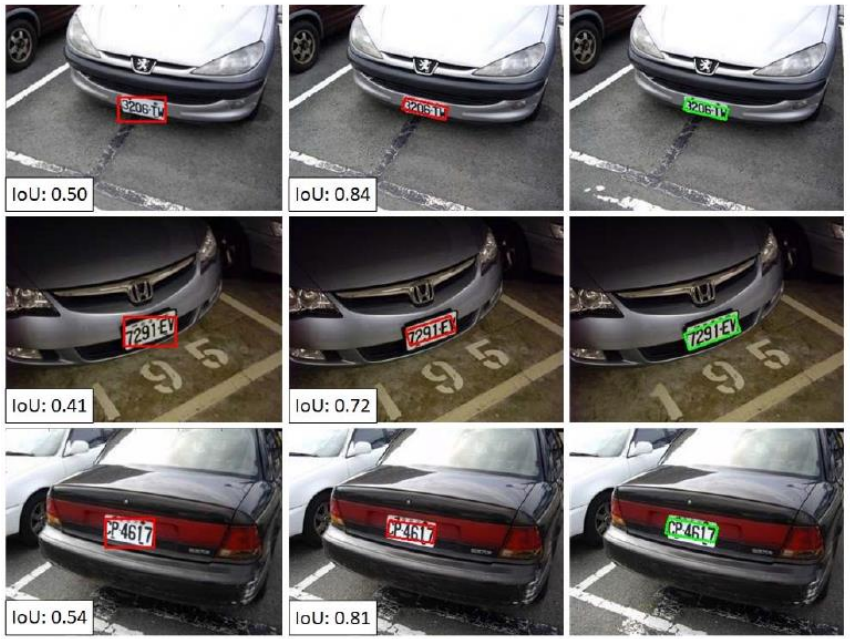


Fig. 5. (a) Random rotation strategy. Each image is randomly rotated within $[-30°, +30°]$. (b) Attention region training set for MD-YOLO. Each attention region has the size of $(2 \times width) \times (3 \times height)$ relative to the ground truth (car license plate).

VSS Part II

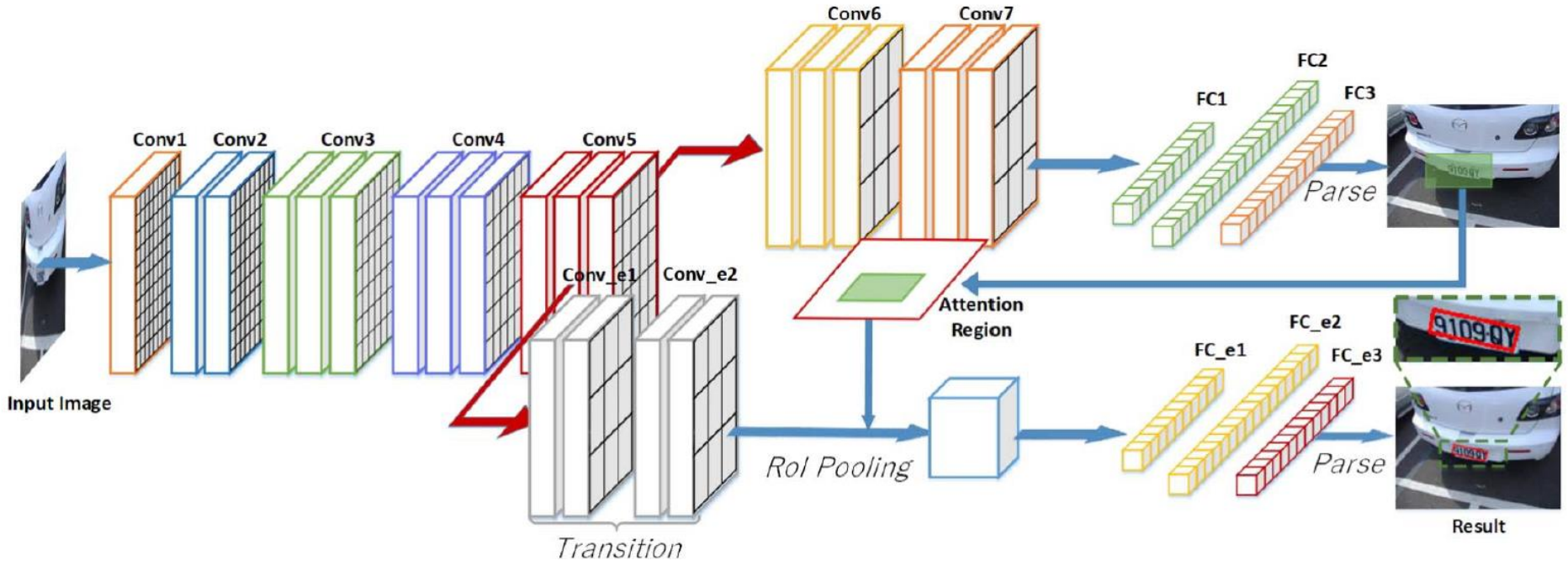
Method \ Dataset	UCSD [17]	PKU [43]
ACF [7]	76.56	75.17
Faster-rcnn [28] (VGG16)	85.54	83.61
SSD [19]	85.32	86.63
R-FCN [4]	89.78	84.40
ALMD-YOLO (ours)	**98.32**	**97.38**



(a) R-FCN (b) ALMD-YOLO (c) Ground truth

VSS Part II

VSS Part II



g. 8. An end-to-end model which has employed the same regression mechanism as ALMD-YOLO.

VSS Part II

TABLE VIII

THE F-SCORE OF AN END-TO-END MODEL AND SPLIT-TRAINING MODEL (ALMD-YOLO)

Method \ IoU	0.5	0.6	0.7
End-to-end	98.6	90.1	61.8
ALMD-YOLO	**99.5**	**93.8**	**79.5**