SUSTech Southern University of Science and Technology

Object Detection for Self-Driving Applications

Group Members:

车凯威

唐雨顺

刘　凯

SUSTech Southern University of Science and Technology

CONTENTS

Review of Proposal

SUSTech Southern University of Science and Technology

✓Detec

✓Detec

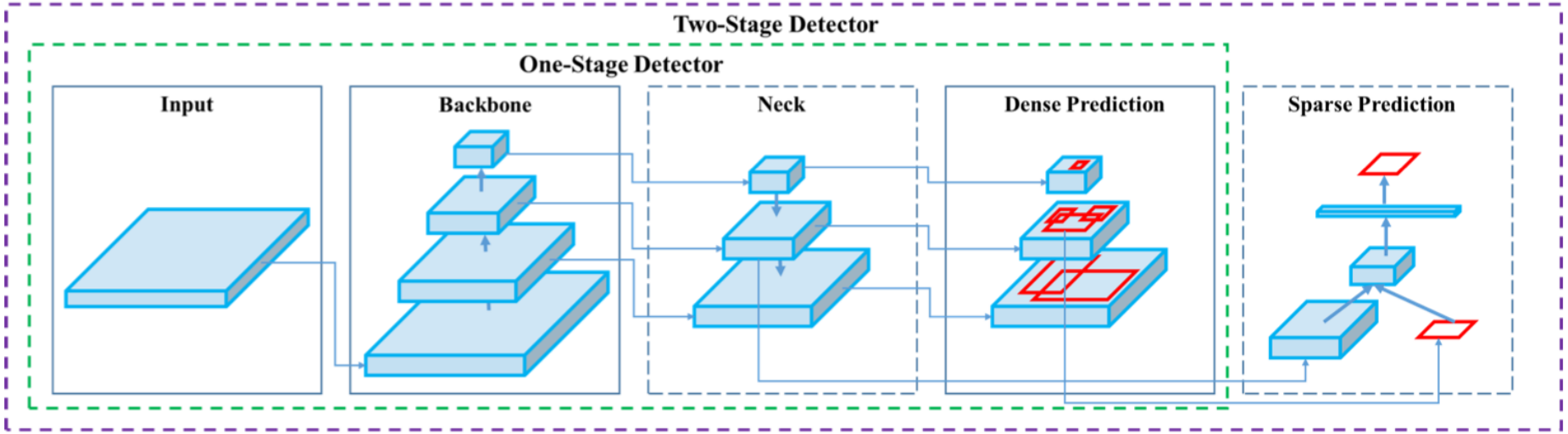
✓Detec

✓Goals

t-Lane-Detection

Object Detection based on YOLOv5
——Structure of YOLO

SUSTech Southern University of Science and Technology



Input: { Image, Patches, Image Pyramid, … }

Backbone: { VGG16, ResNet-50, ResNeXt-101, Darknet53, … }

Neck: { FPN, PANet, Bi-FPN }

Head:

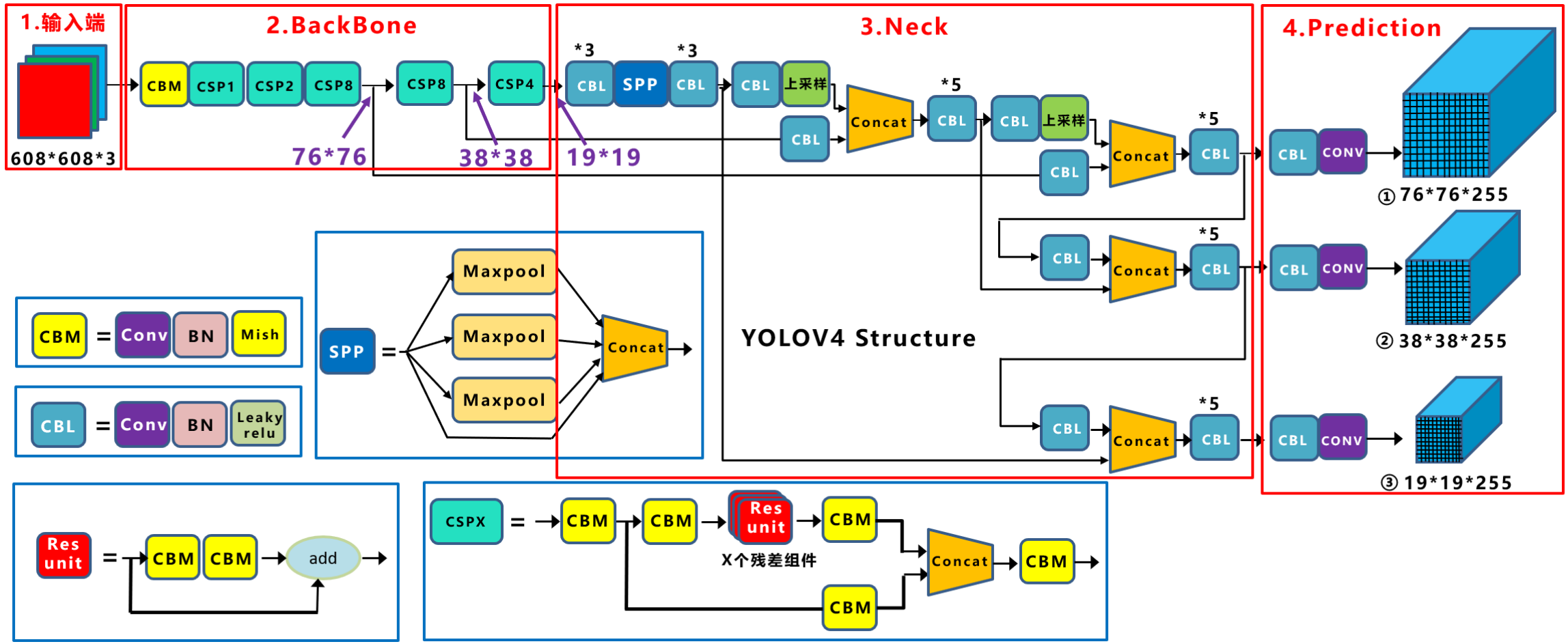
Dense Prediction: { YOLO, SSD, RetinaNet, FCOS }

Sparse Prediction: { Faster R-CNN, R-FCN }

Bochkovskiy, Alexey et al. “YOLOv4: Optimal Speed and Accuracy of Object Detection.” *ArXiv* abs/2004.10934 (2020): n. pag.

Object Detection based on YOLOv5
——Review of YOLOv4

SUSTech Southern University of Science and Technology



Object Detection based on YOLOv5 ——Weakness and Improvement

SUSTech Southern University of Science and Technology

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9

- Incomplete detection
- Inaccurate bounding box
- Poor detection of small targets

Redmon, Joseph and Ali Farhadi. "YOLOv3: An Incremental Improvement." *ArXiv* abs/1804.02767 (2018): n. pag.

	yolo v1	yolo v2	yolo v3	yolo v4	yolo v5
backbone	GoogleNet	DarkNet19	DarkNet53	CSPDarkNet53	Focus, CSP
neck	None	None	None	SPP, PAN	FPN, PAN
head	yolo: fc(1570)→ 7*7*(5*2+20)	yolo v2: conv→ 13*13*5*(5+20)	yolo v3: conv→ N*N*3*(5+80)	yolo v3: conv→ N*N*3*(5+80)	conv→ N*N*3*(5+80)

Object Detection based on YOLOv5
——Loss Function of v1-v4

SUSTech Southern University of Science and Technology

yolo v1:

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

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

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

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

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

yolo v2-v4:

Loss
- (x, y)
- (w, h)
- Confidence
- Class

Redmon, Joseph et al. “You Only Look Once: Unified, Real-Time Object Detection.” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016): 779-788.

Object Detection based on YOLOv5
——Here is YOLOv5

SUSTech Southern University of Science and Technology

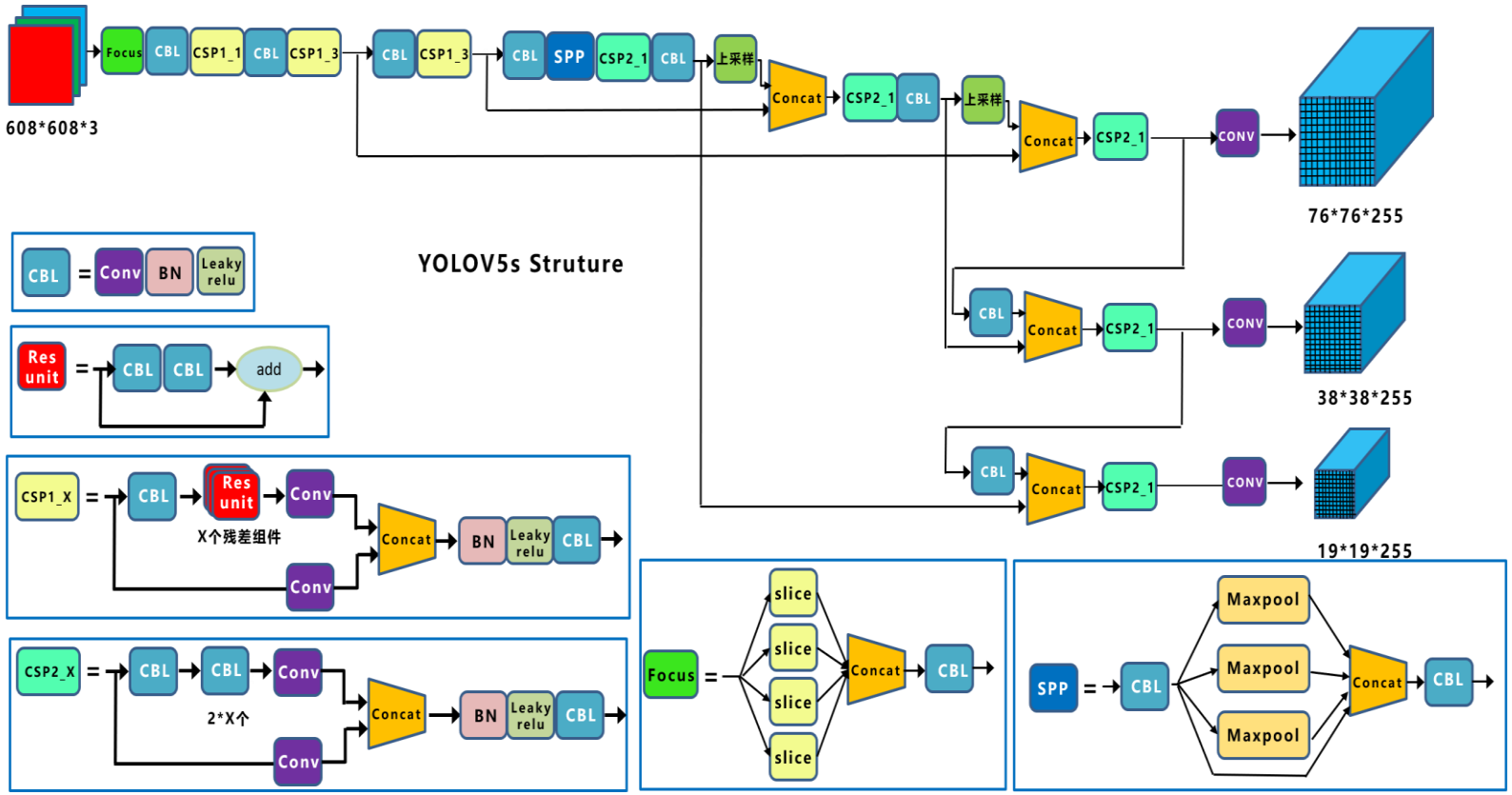
Data Augmentation

Auto Learning Bounding Box Anchors

Structure:
CSP PANET SPP

Cost Function

Activation Function



Object Detection based on YOLOv5
——Data Augmentation

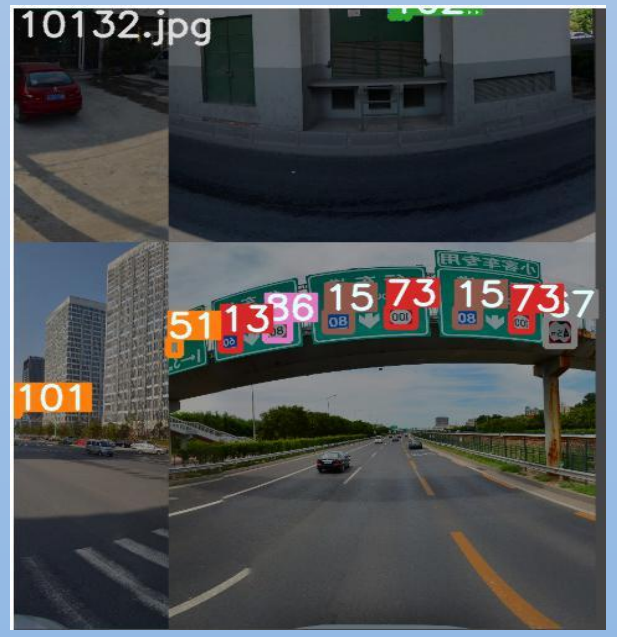
SUSTech Southern University of Science and Technology

Data Augmentation

Scaling

Color Space Adjustment

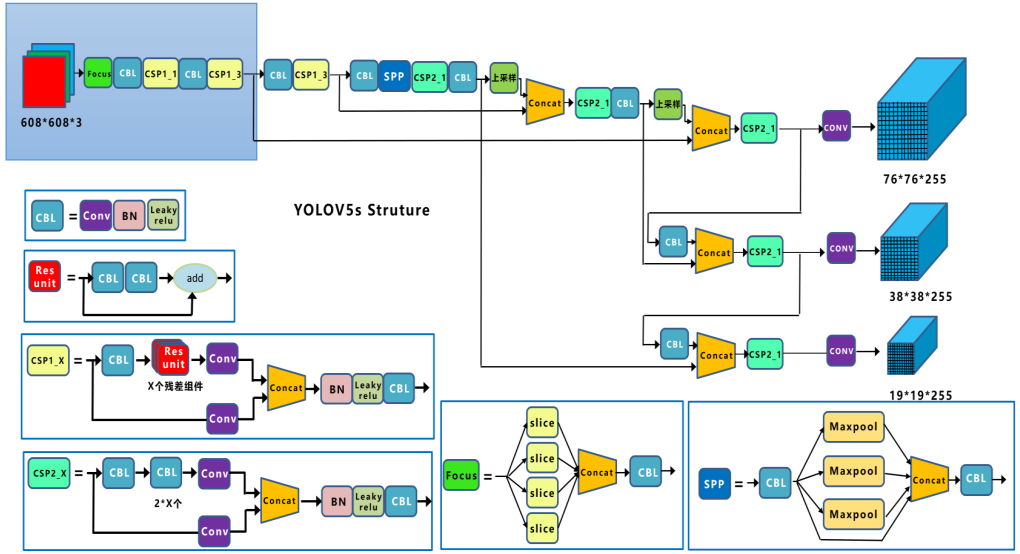
Mosaic Augmentation



Enrich the background of the detected object

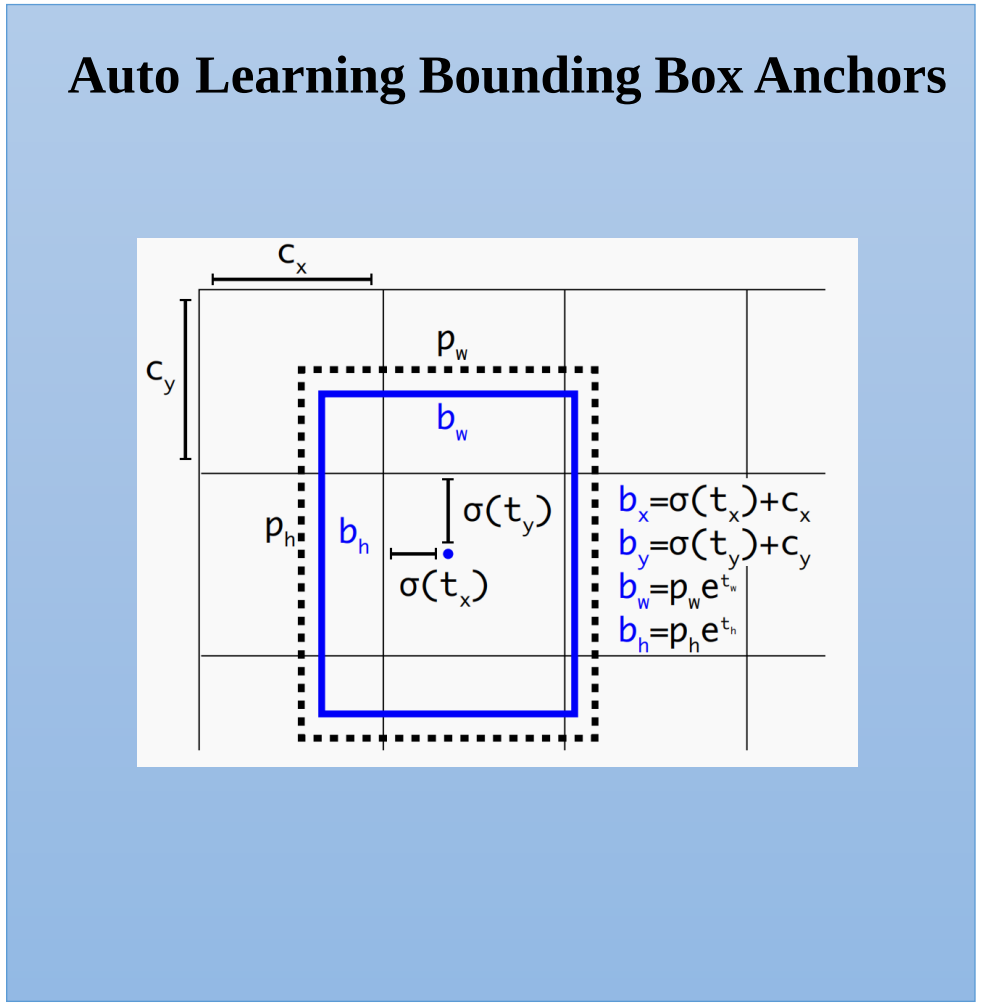
Encourages the model to localize different types of images in different portions of the frame

Identify objects at a smaller scale than normal



Object Detection based on YOLOv5
——Auto Learning Bounding Box Anchors

SUSTech Southern University of Science and Technology



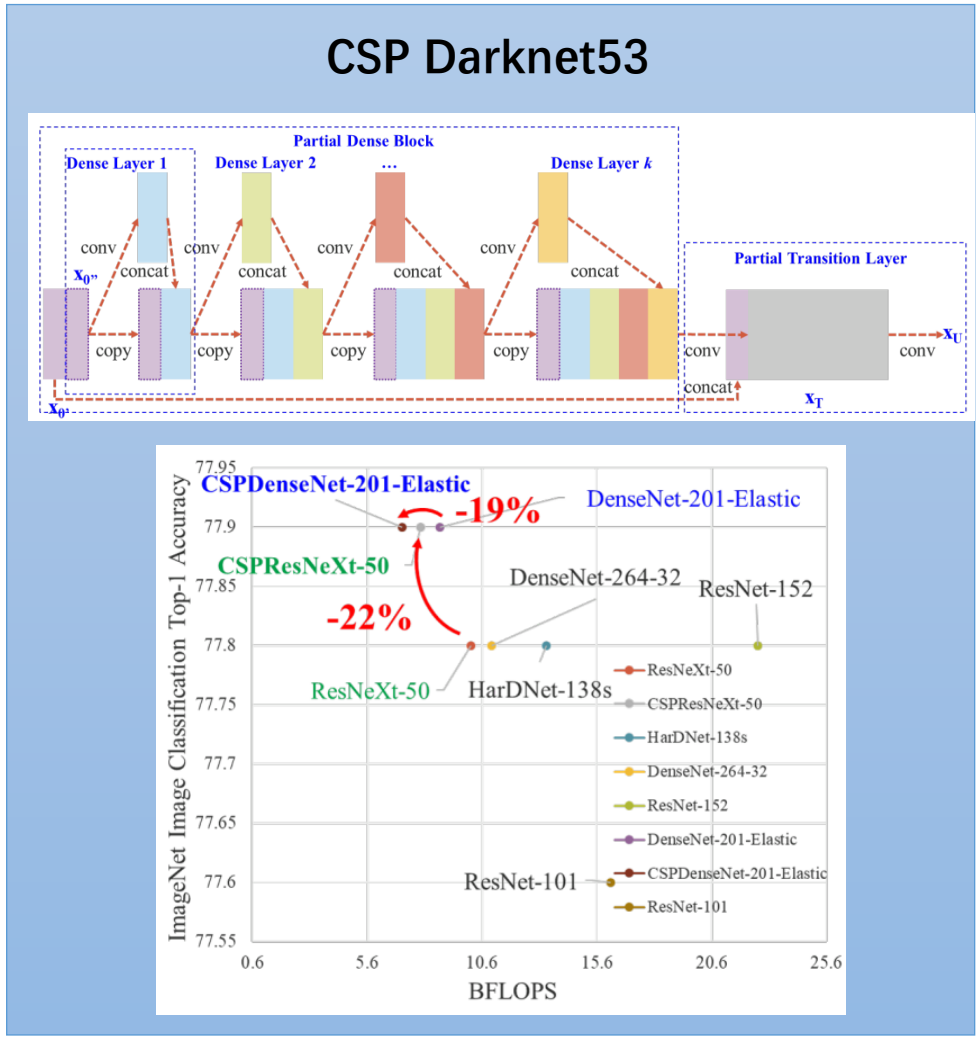
YOLOv2,3,4:
Anchor is calculated from the dataset.

YOLOv5:
Auto Learning

Redmon, Joseph and Ali Farhadi. “YOLO9000: Better, Faster, Stronger.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017): 6517-6525.

Object Detection based on YOLOv5
——Backbone: CSP

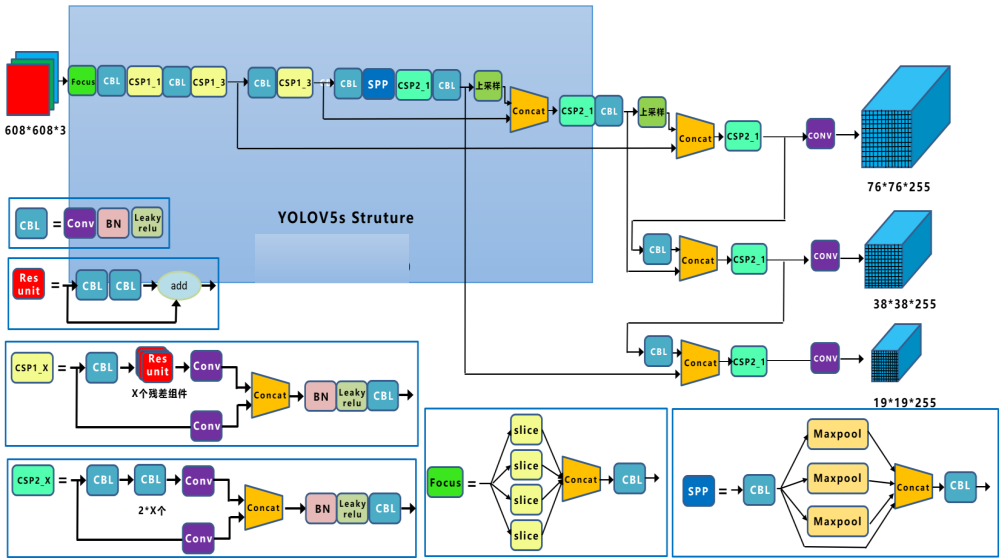
SUSTech Southern University of Science and Technology



Backbone: extracting rich information features from input image

CSPNet: solves the problem of gradient information repetition in other large convolutional neural network framework backbone

Method： the gradient changes are integrated into the feature map.

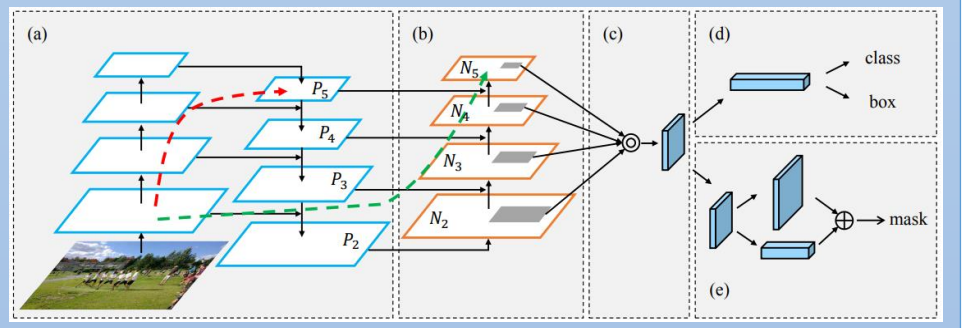


Wang, Chien-Yao et al. “CSPNet: A New Backbone that can Enhance Learning Capability of CNN.” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2020): 1571-1580.

Object Detection based on YOLOv5
——Neck: PANet

SUSTech Southern University of Science and Technology

PANet



	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Champion 2015 [10]	28.4	51.6	28.1	9.4	30.6	45.6
Champion 2016 [33]	37.6	59.9	40.4	17.1	41.0	56.0
Our Team 2017	**46.7**	**69.5**	**51.3**	**26.0**	**49.1**	**64.0**
PANet baseline	38.2	60.2	41.4	19.1	41.1	52.6
+ DCN [11]	39.5	62.0	42.8	19.8	42.2	54.7
+ testing tricks	42.0	63.5	46.0	21.8	44.4	58.1
+ larger model	44.4	67.0	48.5	23.6	46.5	62.2
+ ensemble	**46.7**	**69.5**	**51.3**	**26.0**	**49.1**	**64.0**

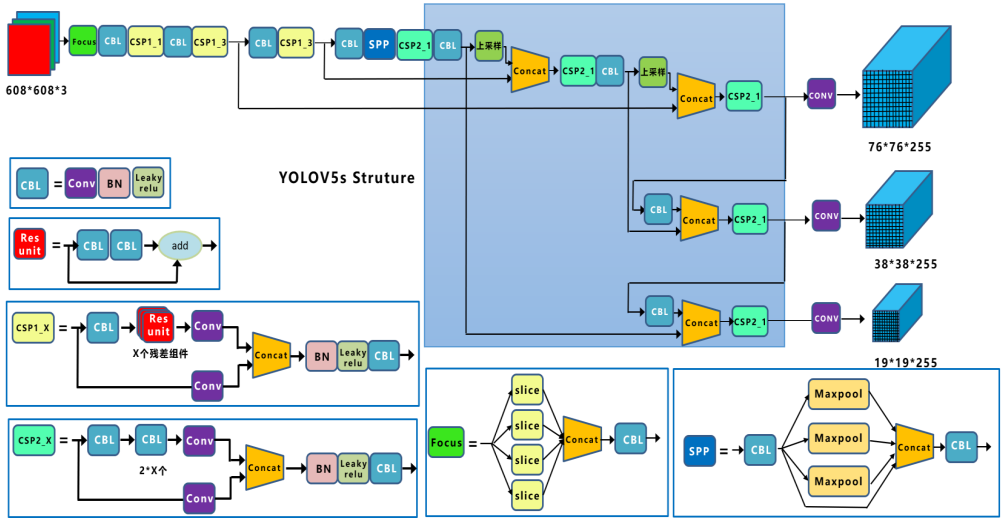
Table 6. Mask AP of COCO Instance Segmentation Challenge in different years on *test-dev*.

Neck: generate feature pyramids and enhance the detection of objects with different scales

FPN: always used as the feature aggregation layer of object detection framework.

PANET: learn from Mask R-CNN and FPN structure and strengthened information dissemination

PAN = FPN + Bottom-up path augmentation + Adaptive feature pooling



Object Detection based on YOLOv5
——Cost Function

Cost Function

定位损失

$$\lambda_{iou}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}L_{ciou}$$

目标置信度损失

$$+\lambda_{cls}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{obj}\lambda_c(C_i-\hat{C}_i)^2+\lambda_{cls}\sum_{i=0}^{S^2}\sum_{j=0}^{B}1_{ij}^{noobj}\lambda_c(C_i-\hat{C}_i)^2$$

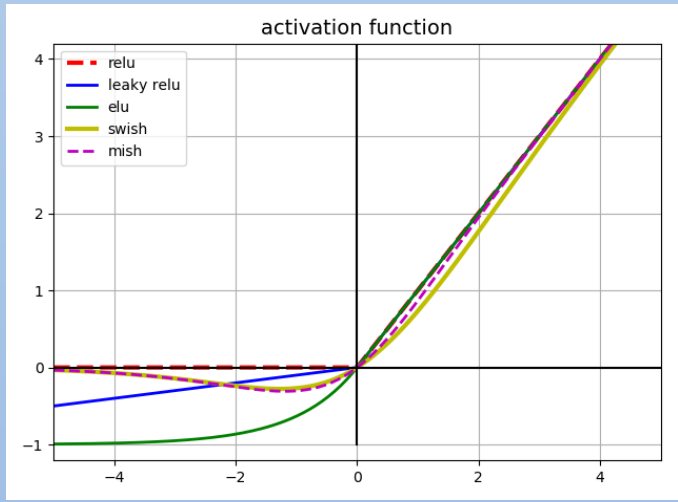
分类损失

$$-\sum_{i=0}^{S^2}\sum_{j=0}^{B^2}1_{ij}^{obj}\sum_{c\in classes}\lambda_c[\hat{p}_i(c)\log(p_i(c))+(1-\hat{p}_i(c))\log(1-p_i(c))]$$

- λ_{iou} iou_normalizer
- λ_{cls} cls_normalizer
- λ_c classes_multipliers

$$\mathcal{L}_{CIoU}=1-IoU+\frac{\rho^2\left(\mathbf{b},\mathbf{b}^{gt}\right)}{c^2}+\alpha v$$

Activation Function



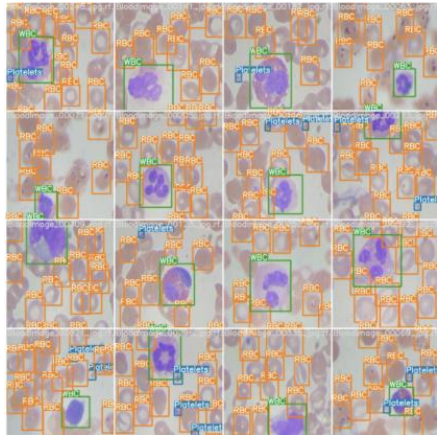
YOLOv5:
The hidden layer
uses Leaky ReLU activation function, and the final detection layer uses Sigmoid activation function.

YOLOv4 uses the Mish activation function.

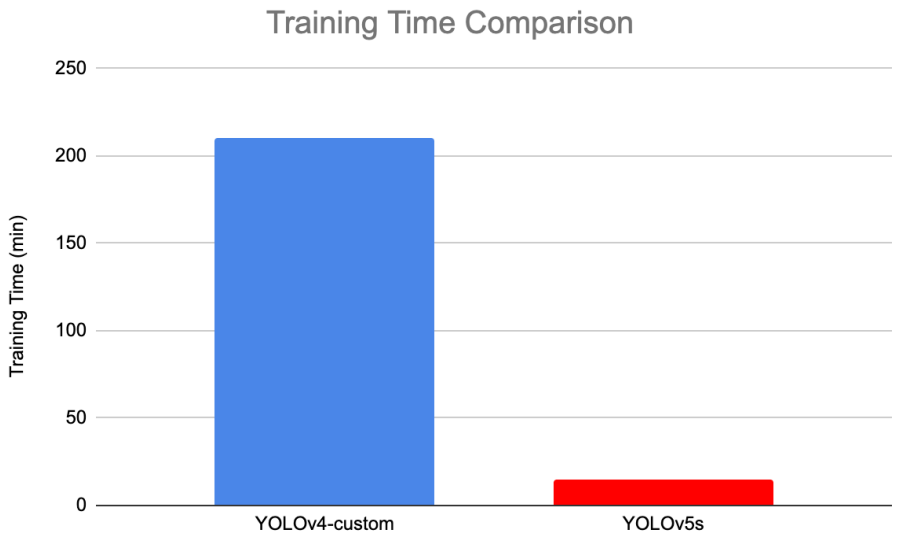
Zheng, Zhaohui et al. “Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression.” ArXiv abs/1911.08287 (2020): n. pag.

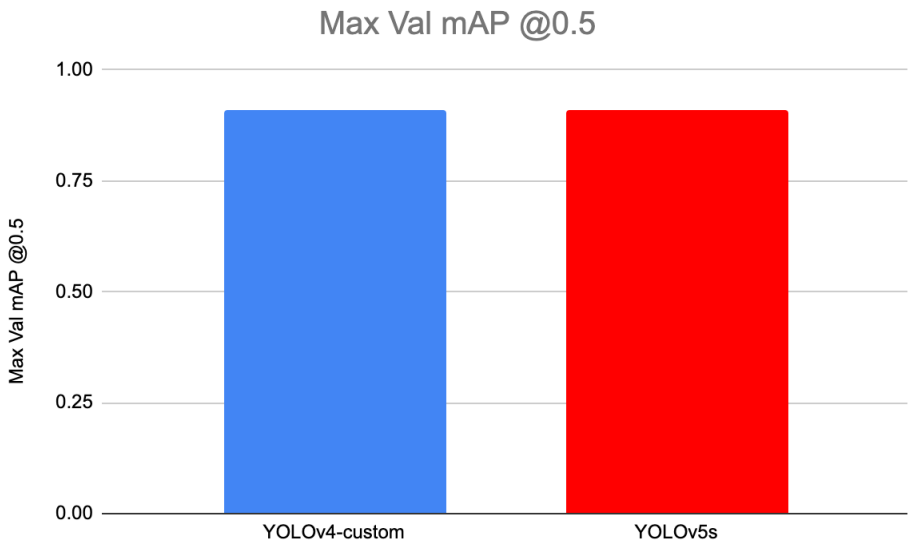
Object Detection based on YOLOv5 ——Metric

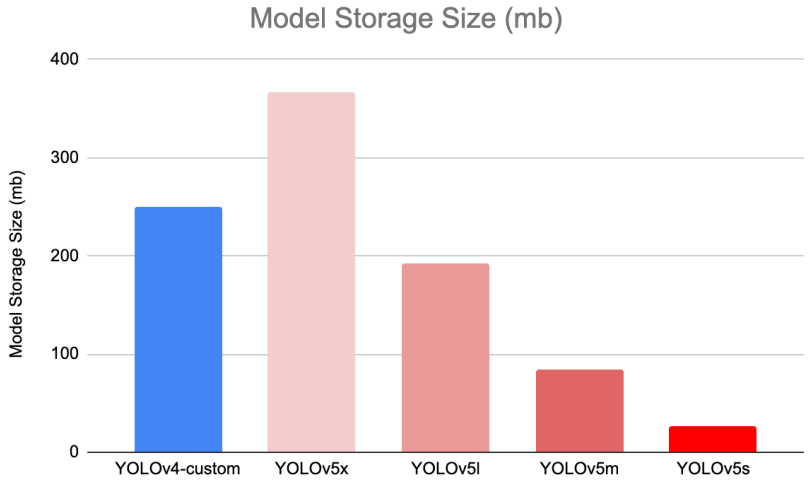
SUSTech Southern University of Science and Technology



[blood cell count and detection dataset]



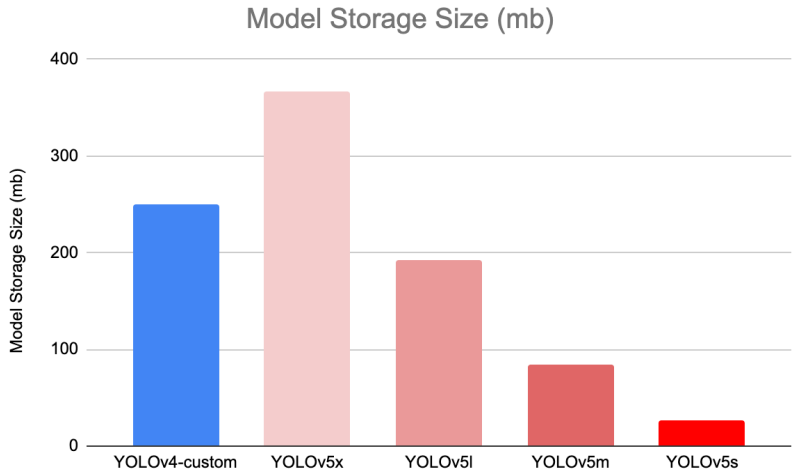


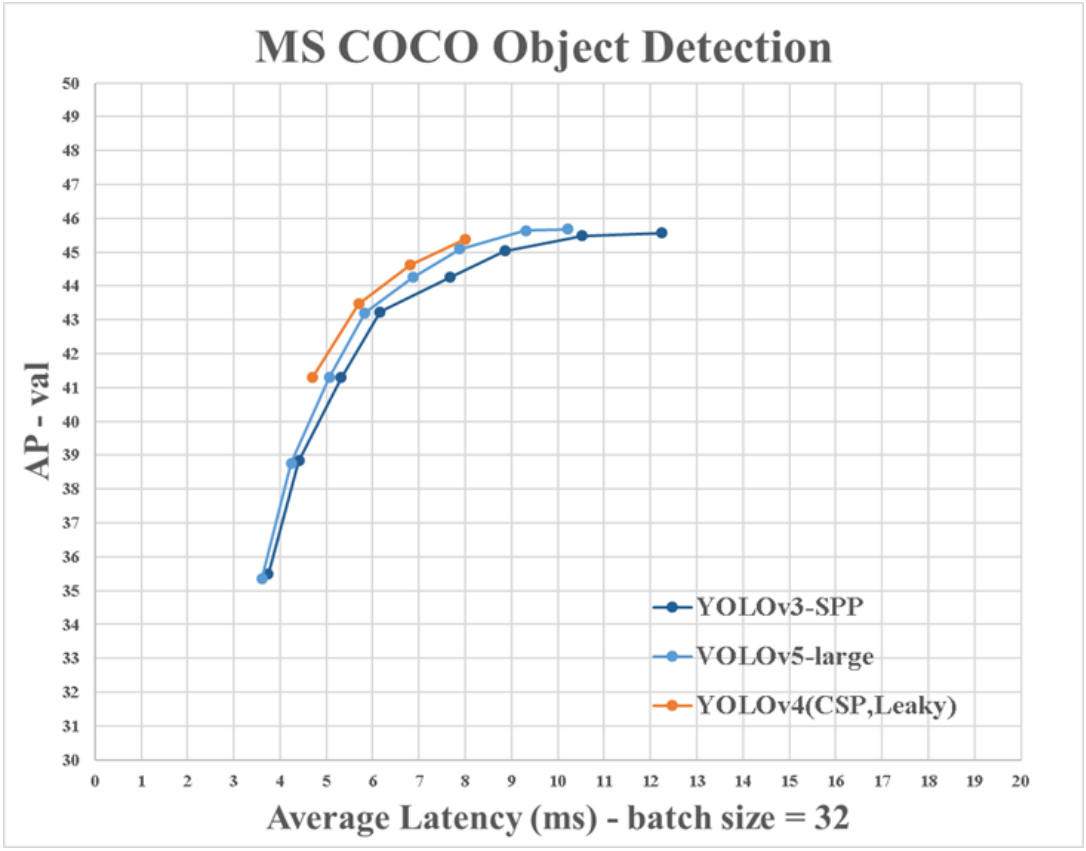


Object Detection based on YOLOv5
——Metric

SUSTech Southern University of Science and Technology

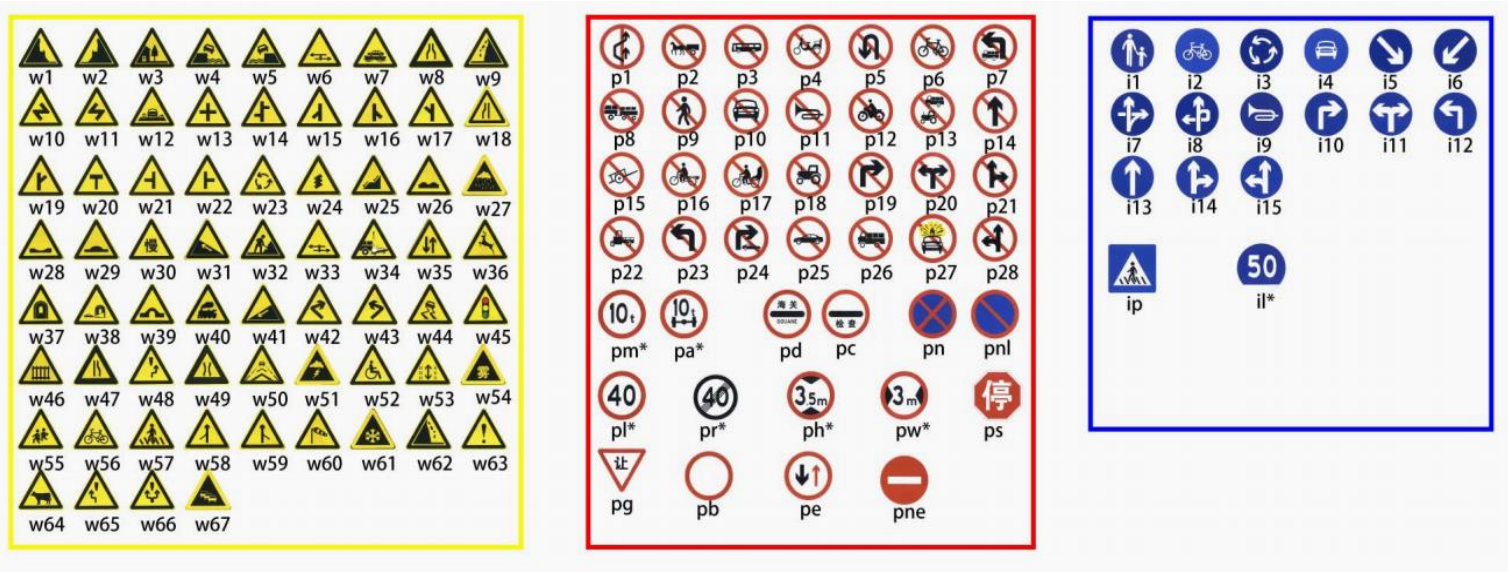
[COCO - Common Objects in Context (cocodataset.org)]





Object Detection based on YOLOv5
——Training Process: Dataset

SUSTech Southern University of Science and Technology



[German Traffic Sign]

[Chinese Traffic Sign Database]

[Tsinghua-Tencent 100K]

https://cg.cs.tsinghua.edu.cn/traffic-sign/

Object Detection based on YOLOv5
——Training Process: Parameters

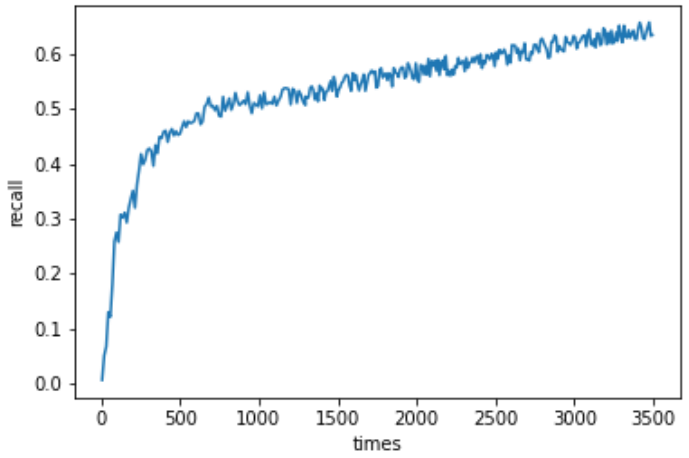
SUSTech Southern University of Science and Technology

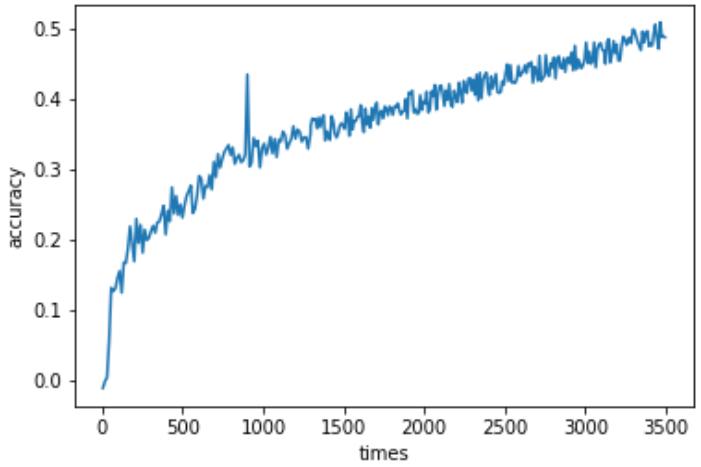
```yaml
nc: 151  # number of classes
depth_multiple: 0.33  # model depth multiple
width_multiple: 0.50  # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23]  # P3/8
  - [30,61, 62,45, 59,119]  # P4/16
  - [116,90, 156,198, 373,326]  # P5/32

# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, BottleneckCSP, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 9, BottleneckCSP, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, BottleneckCSP, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 1, SPP, [1024, [5, 9, 13]]],
   [-1, 3, BottleneckCSP, [1024, False]],  # 9
  ]
```

```yaml
# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],  # cat backbone P4
   [-1, 3, BottleneckCSP, [512, False]],  # 13

   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 4], 1, Concat, [1]],  # cat backbone P3
   [-1, 3, BottleneckCSP, [256, False]],  # 17 (P3/8-small)

   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 14], 1, Concat, [1]],  # cat head P4
   [-1, 3, BottleneckCSP, [512, False]],  # 20 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 10], 1, Concat, [1]],  # cat head P5
   [-1, 3, BottleneckCSP, [1024, False]],  # 23 (P5/32-large)

   [[17, 20, 23], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
  ]
```

Object Detection based on YOLOv5
——Result: Metrics

SUSTech Southern University of Science and Technology





200 epochs on [Tsinghua-Tencent 100K] dataset

	Recall	Accuracy
Fast R-CNN	0.56	0.5
YOLOv5	0.61	0.57

Object Detection based on YOLOv5
——Result

SUSTech Southern University of Science and Technology

Object Detection based on YOLOv5
——Udacity Self Driving Car

SUSTech Southern University of Science and Technology

biker, car, truck

Pedestrian

Traffic Light:
'traffic Light-Green’, 'traffic Light-Green Left’,
'traffic Light-Red', 'traffic Light-Red Left’,
'traffic Light-Yellow', 'traffic Light-Yellow Left'

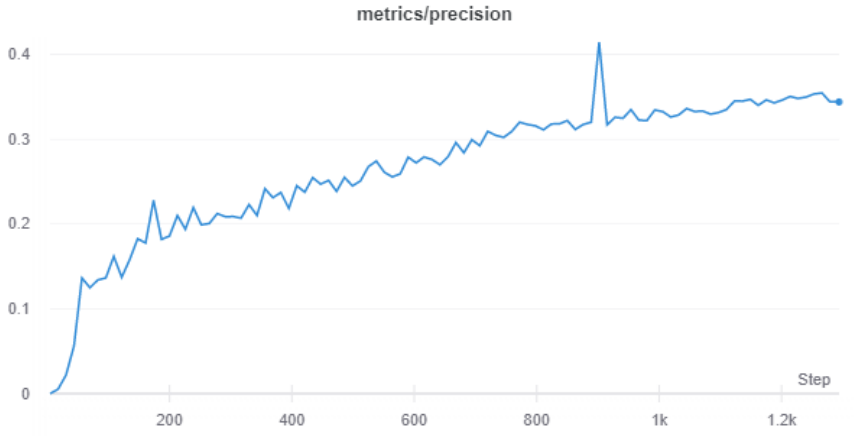
Udacity Self Driving Car

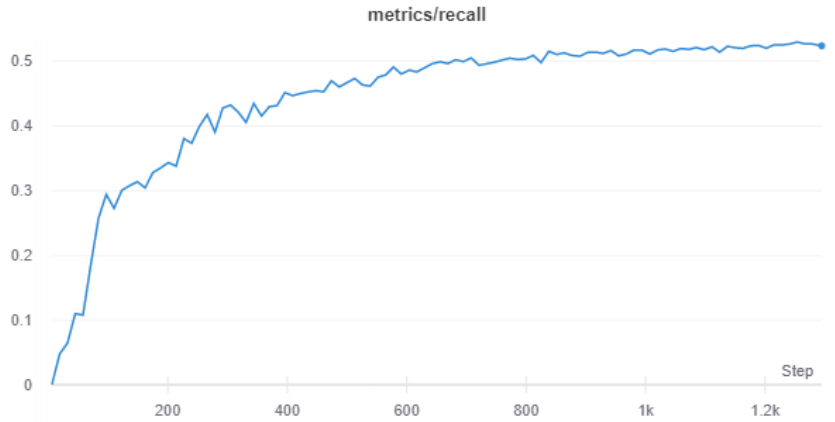
https://public.roboflow.com/object-detection/self-driving-car

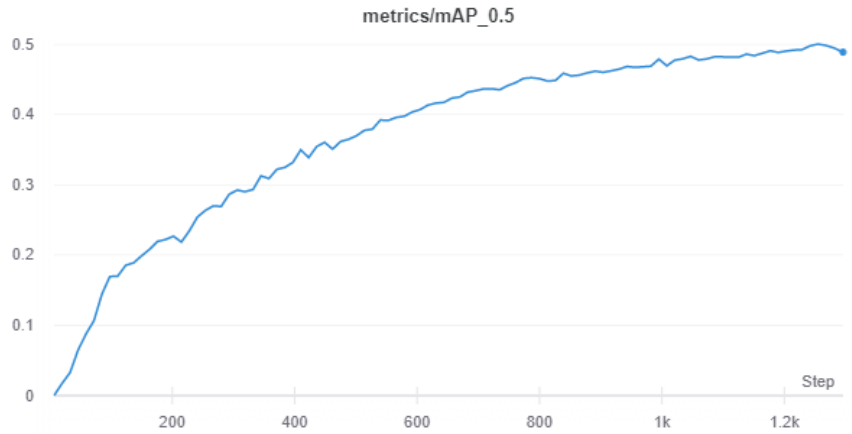
Object Detection based on YOLOv5
——Result: Metrics

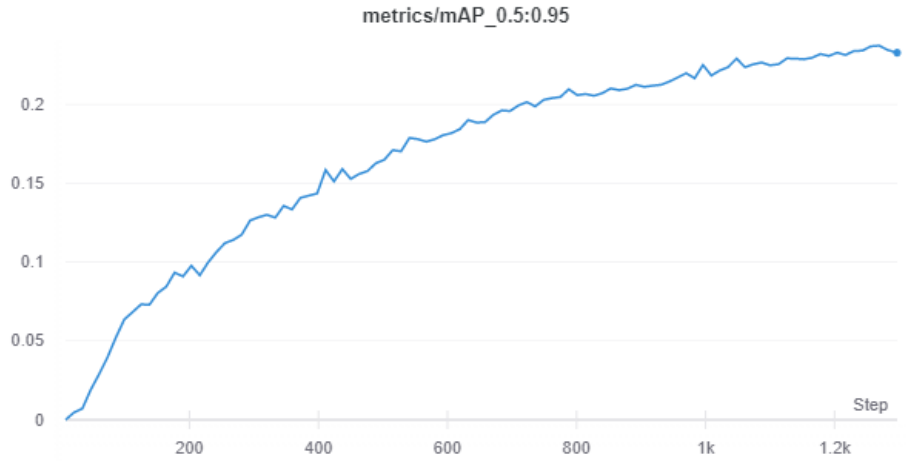
SUSTech Southern University of Science and Technology

100 epochs on [Udacity Self Driving Car] dataset







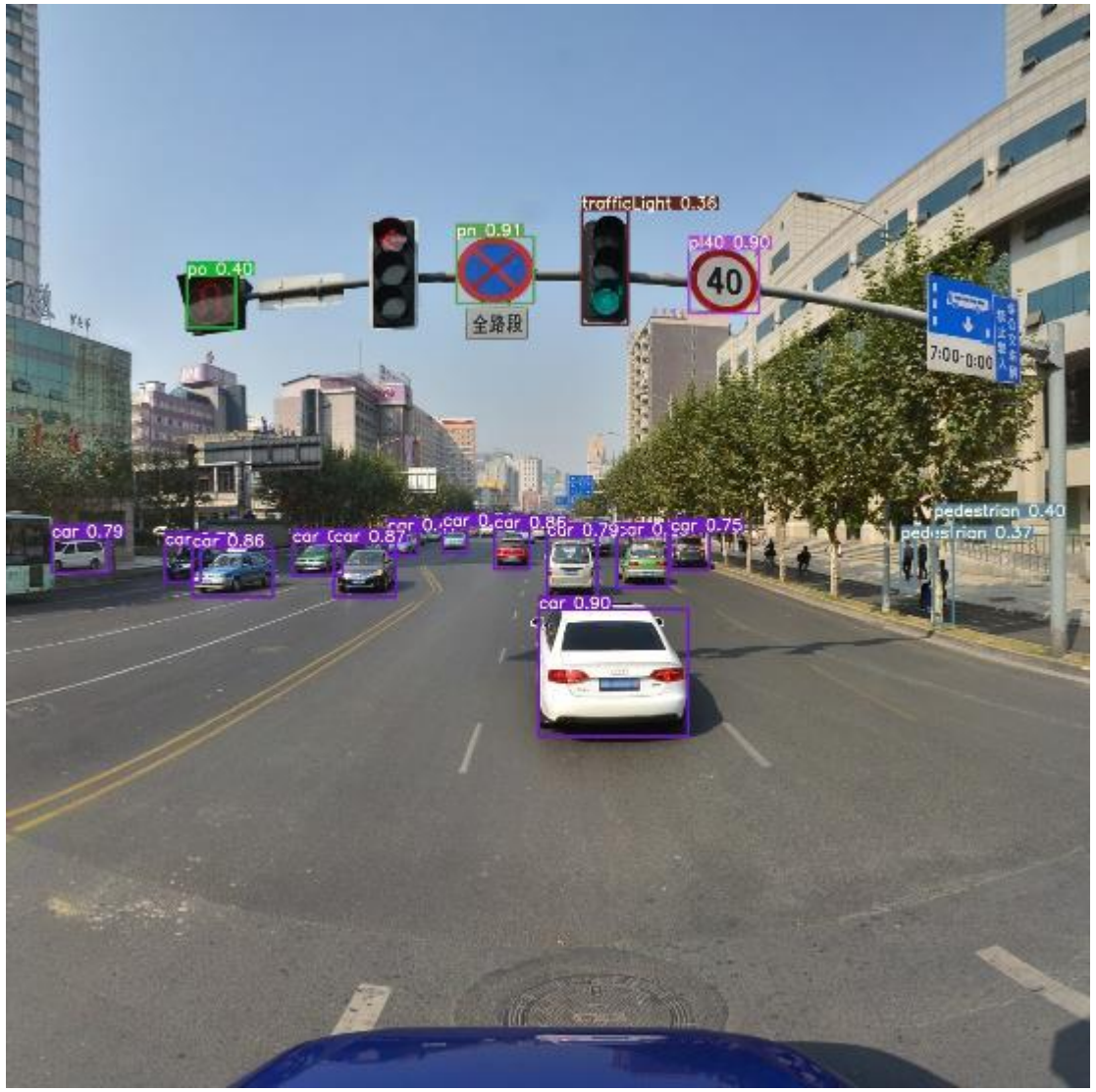


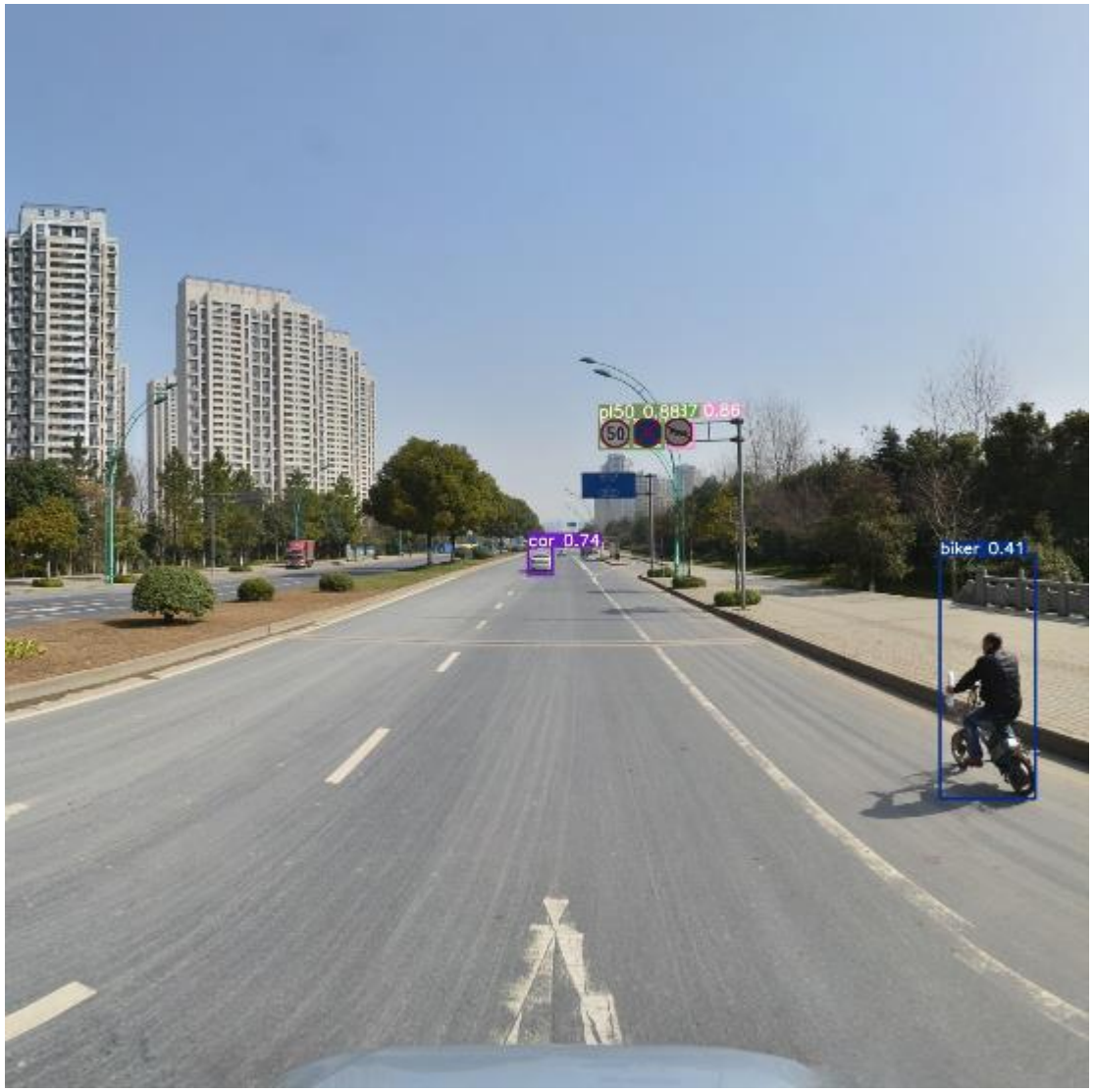
Object Detection based on YOLOv5
——Result

SUSTech Southern University of Science and Technology

Object Detection based on YOLOv5
——Result

SUSTech Southern University of Science and Technology





Object Detection based on YOLOv5 ——Summary

SUSTech Southern University of Science and Technology

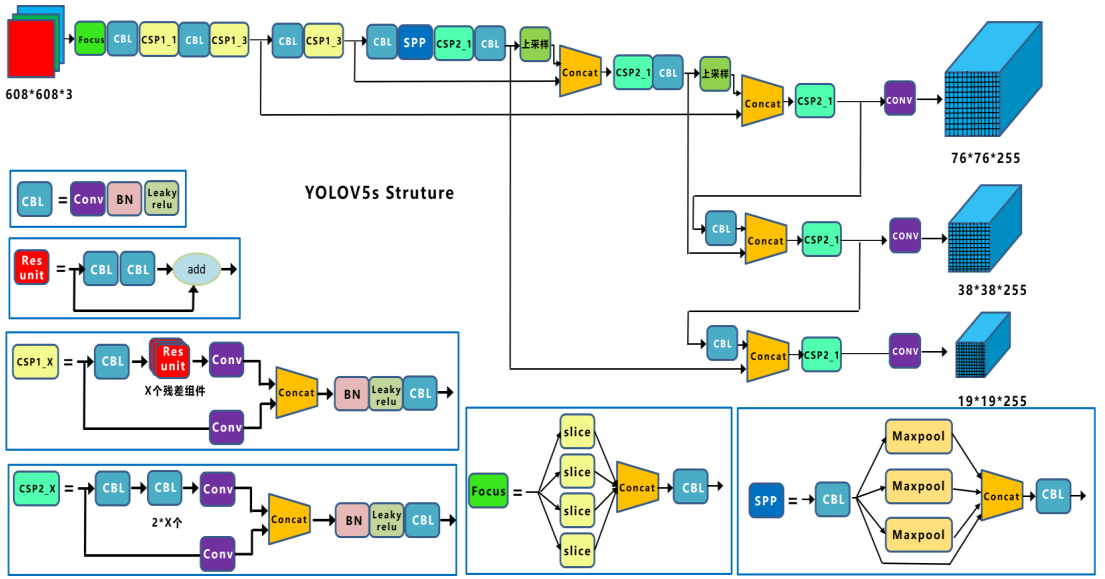
Data Augmentation

Auto Learning
Bounding Box Anchors

Structure: CSP PANET SPP

Cost Function

Activation Function



Not Bad Accuracy

140FPS

Mini Size

Lane Detection based on UFAST
——Introduction

SUSTech Southern University of Science and Technology

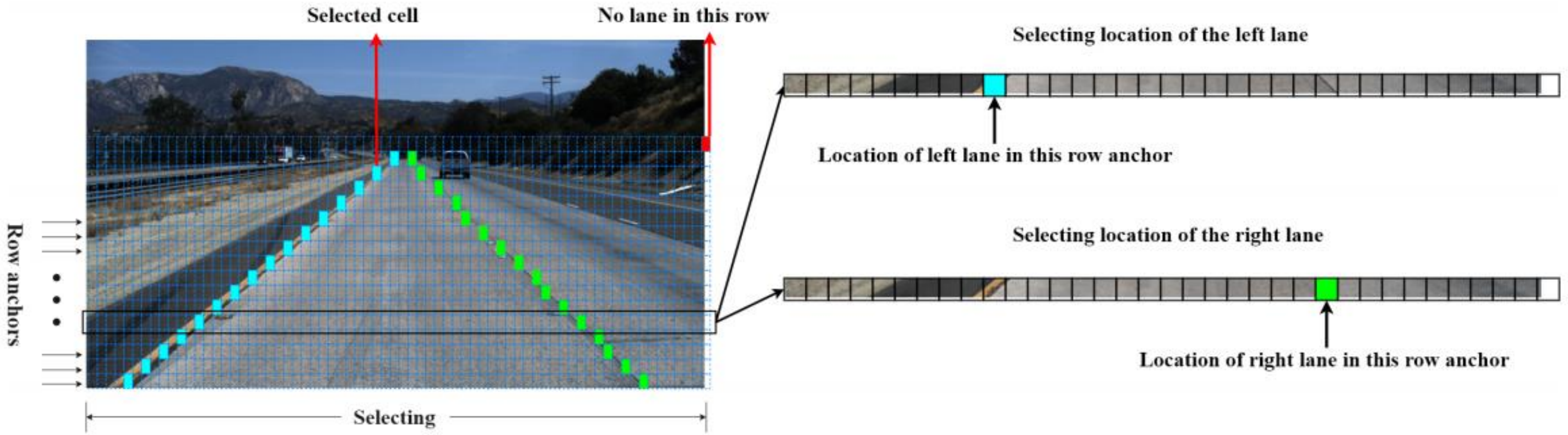
Lane detection enables vehicles to locate in the lane correctly, to observe the traffic rules stipulated in the lane, and to assist subsequent lane departure or trajectory planning decisions.

Traditional image processing methods have low robustness and it is easy to be affected by the environment. Most semantic segmentation methods based on deep learning classify each pixel of the image, which is difficult to achieve real-time performance due to the large computation.

Qin, Zequn et al. “Ultra Fast Structure-aware Deep Lane Detection.” ECCV (2020).

Lane Detection based on UFAST
——Proposed formulation

SUSTech Southern University of Science and Technology



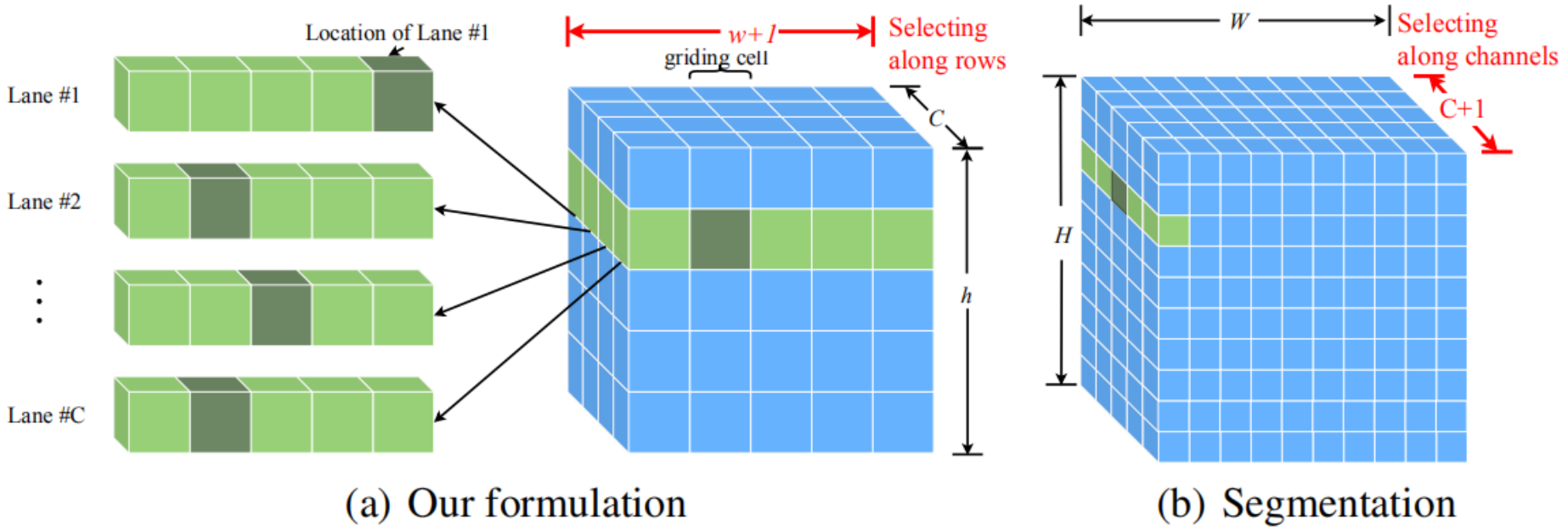
$$P_{i,j,:} = f^{ij}(X), \text{ s.t. } i \in [1, C], j \in [1, h]$$

Row anchors are the predefined row locations, and our formulation is defined as horizontally selecting on each of row anchor. On the right of the image, a background gridding cell is introduced to indicate no lane in this row.

Lane Detection based on UFAST
——Comparison with segmentation

SUSTech Southern University of Science and Technology

Our formulation is selecting locations (grids) on rows, while segmentation is classifying every pixel. The proposed formulation significantly reduces the computational cost.



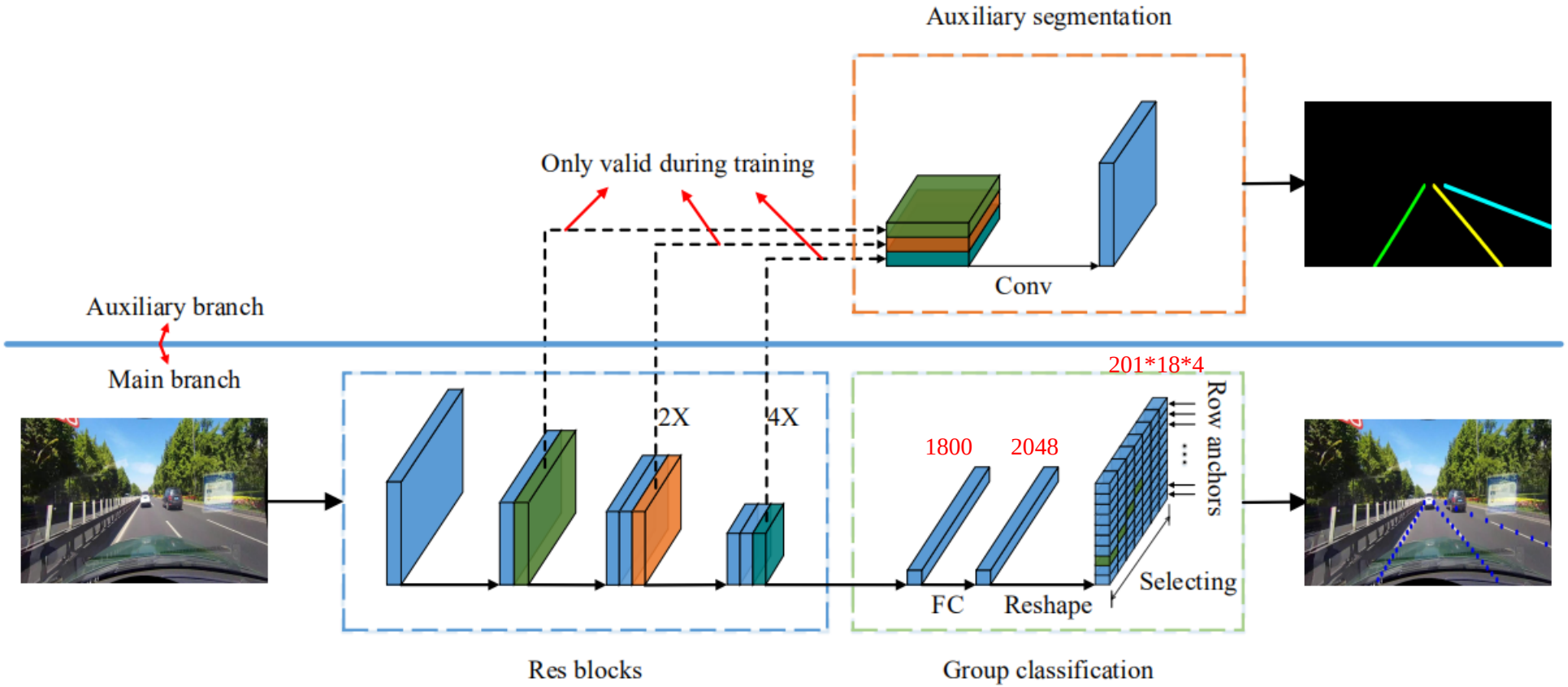
Culane Dataset:

For Row-Anchor: C*h*(w+1) = 4*18*(200+1) = 1e4

For Segmentation: (C+1)*W*H = 5*800*288 = 1e6

Lane Detection based on UFAST
——Framework

SUSTech Southern University of Science and Technology



Overall architecture

Lane Detection based on UFAST
——Loss function

SUSTech Southern University of Science and Technology

$$L_{total} = L_{cls} + \alpha L_{str} + \beta L_{seg},$$

$$L_{cls} = \sum_{i=1}^{C}\sum_{j=1}^{h} L_{CE}(P_{i,j,:}, T_{i,j,:}),$$

$$P_{i,j,:} = f^{ij}(X),\ \text{s.t.}\ \ i \in [1, C], j \in [1, h],$$

$$L_{str} = L_{sim} + \lambda L_{shp},$$

$$L_{sim} = \sum_{i=1}^{C}\sum_{j=1}^{h-1} \left\| P_{i,j,:} - P_{i,j+1,:} \right\|_1 ,$$

$$L_{shp} = \sum_{i=1}^{C}\sum_{j=1}^{h-2} \| (Loc_{i,j} - Loc_{i,j+1}) - (Loc_{i,j+1} - Loc_{i,j+2}) \|_1,$$

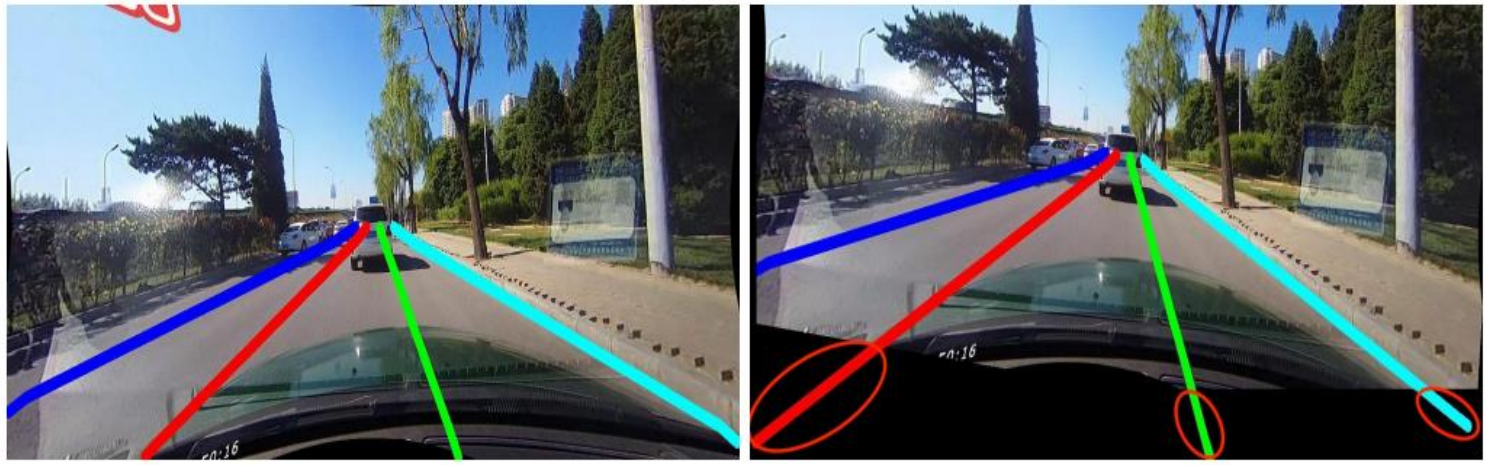
$$Prob_{i,j,:} = softmax(P_{i,j,1:w}),$$

$$Loc_{i,j} = \sum_{k=1}^{w} k \cdot Prob_{i,j,k}$$

Lane Detection based on UFAST ——Experiments

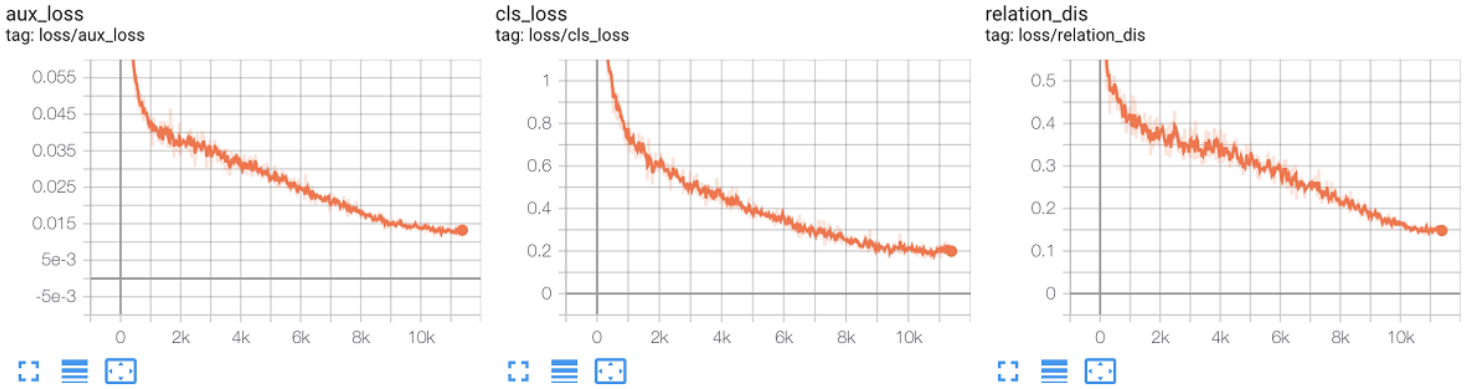
SUSTech Southern University of Science and Technology

Data augmentation

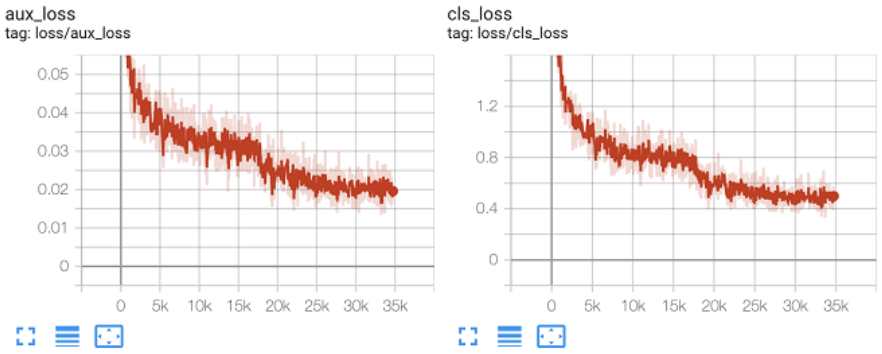


(a) Original anaotation (b) Augmentated result

100 epochs on TuSimple dataset



50 epochs on Culane dataset



Lane Detection based on UFAST
——Experiments and Visualization

SUSTech Southern University of Science and Technology

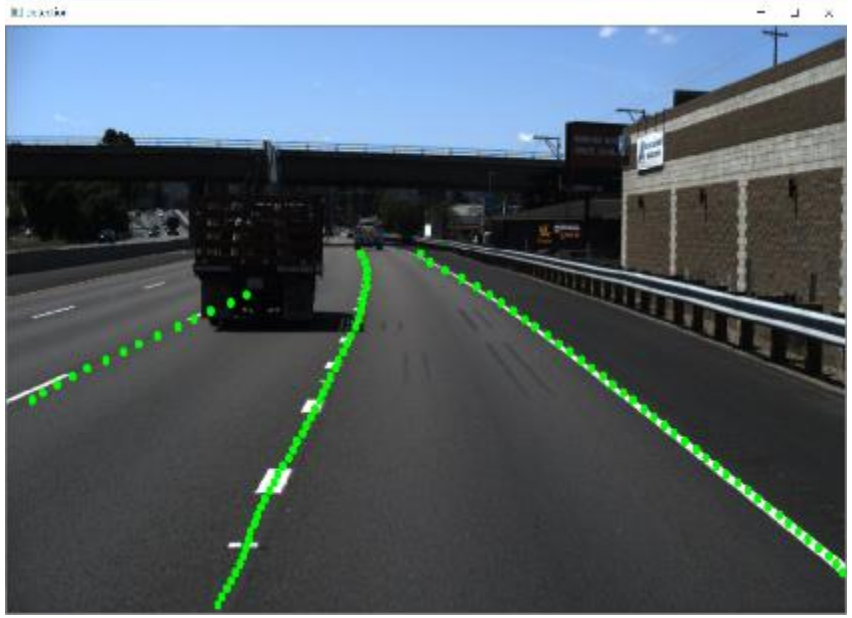
TuSimple

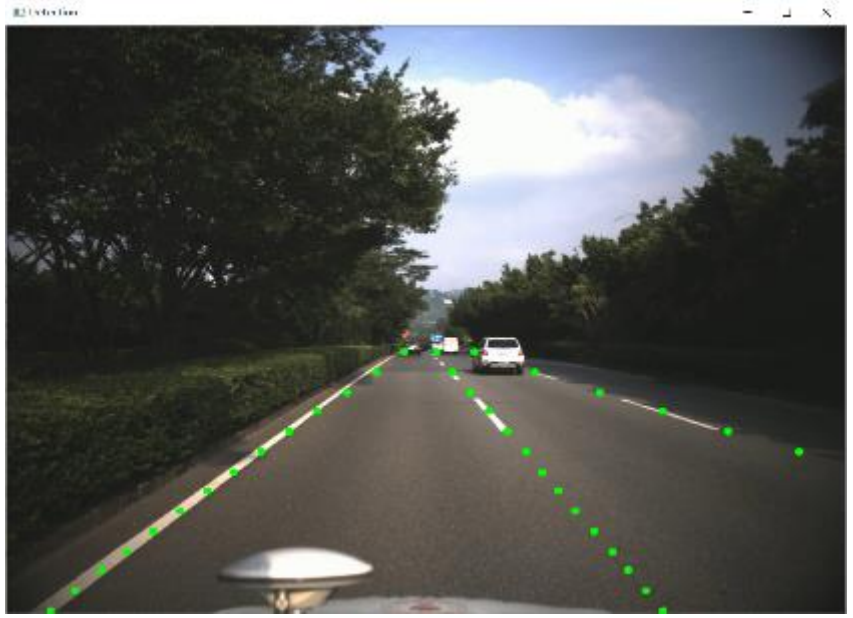
Method	Accuracy	Runtime(ms)	Multiple
Res18-Seg [3]	92.69	25.3	5.3x
Res34-Seg [3]	92.84	50.5	2.6x
LaneNet [21]	96.38	19.0	7.0x
EL-GAN [5]	96.39	>100	<1.3x
SCNN [22]	96.53	133.5	1.0x
SAD [9]	**96.64**	13.4	10.0x
Res34-Ours	96.06	5.9	22.6x
Res18-Ours	95.87	**3.2**	**41.7x**

Culane

Category	Res50-Seg[3]	SCNN[22]	FD-50[24]	Res34-SAD	SAD[9]	Res18-Ours	Res34-Ours
Normal	87.4	90.6	85.9	89.9	90.1	87.7	**90.7**
Crowded	64.1	69.7	63.6	68.5	68.8	66.0	**70.2**
Night	60.6	66.1	57.8	64.6	66.0	62.1	**66.7**
No-line	38.1	43.4	40.6	42.2	41.6	40.2	**44.4**
Shadow	60.7	66.9	59.9	67.7	65.9	62.8	**69.3**
Arrow	79.0	84.1	79.4	83.8	84.0	81.0	**85.7**
Dazzlelight	54.1	58.5	57.0	59.9	**60.2**	58.4	59.5
Curve	59.8	64.4	65.2	66.0	65.7	57.9	**69.5**
Crossroad	2505	1990	7013	1960	1998	**1743**	2037
Total	66.7	71.6	-	70.7	70.8	68.4	**72.3**
Runtime(ms)	-	133.5	-	50.5	13.4	3.1	5.7
Multiple	-	1.0x	-	2.6x	10.0x	43.0x	23.4x
FPS	-	7.5	-	19.8	74.6	322.5	175.4

TuSimple-Resnet18





Culane-Resnet18

Future work
——Lane Detection based on UFAST

SUSTech Southern University of Science and Technology

$$L_{str} = L_{sim} + \lambda L_{shp},$$

$$L_{sim} = \sum_{i=1}^{C} \sum_{j=1}^{h-1} \| P_{i,j,:} - P_{i,j+1,:} \|_1 ,$$

[0,1,0,0,0,0]
[0,1,0,0,0,0]
$L_{sim} = 0$

[0,0,1,0,0,0]
[0,1,0,0,0,0]

[0,0,0,0,1,0]
[0,1,0,0,0,0]
$L_{sim} = 2$

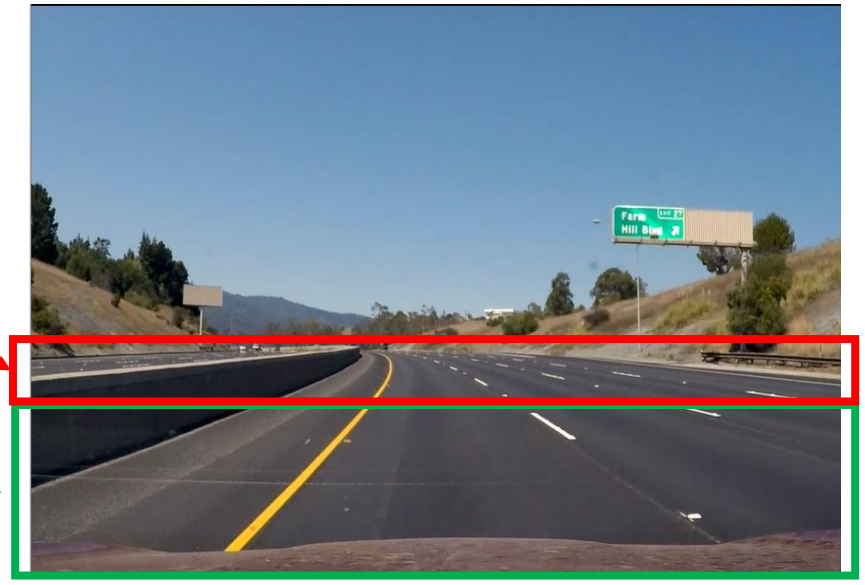
$$L_{shp} = \sum_{i=1}^{C} \sum_{j=1}^{h-2} \| (Loc_{i,j} - Loc_{i,j+1}) - (Loc_{i,j+1} - Loc_{i,j+2}) \|_1,$$

$$Loc_{i,j} = \sum_{k=1}^{w} k \cdot Prob_{i,j,k}$$

$$Prob_{i,j,:} = softmax(P_{i,j,1:w}),$$

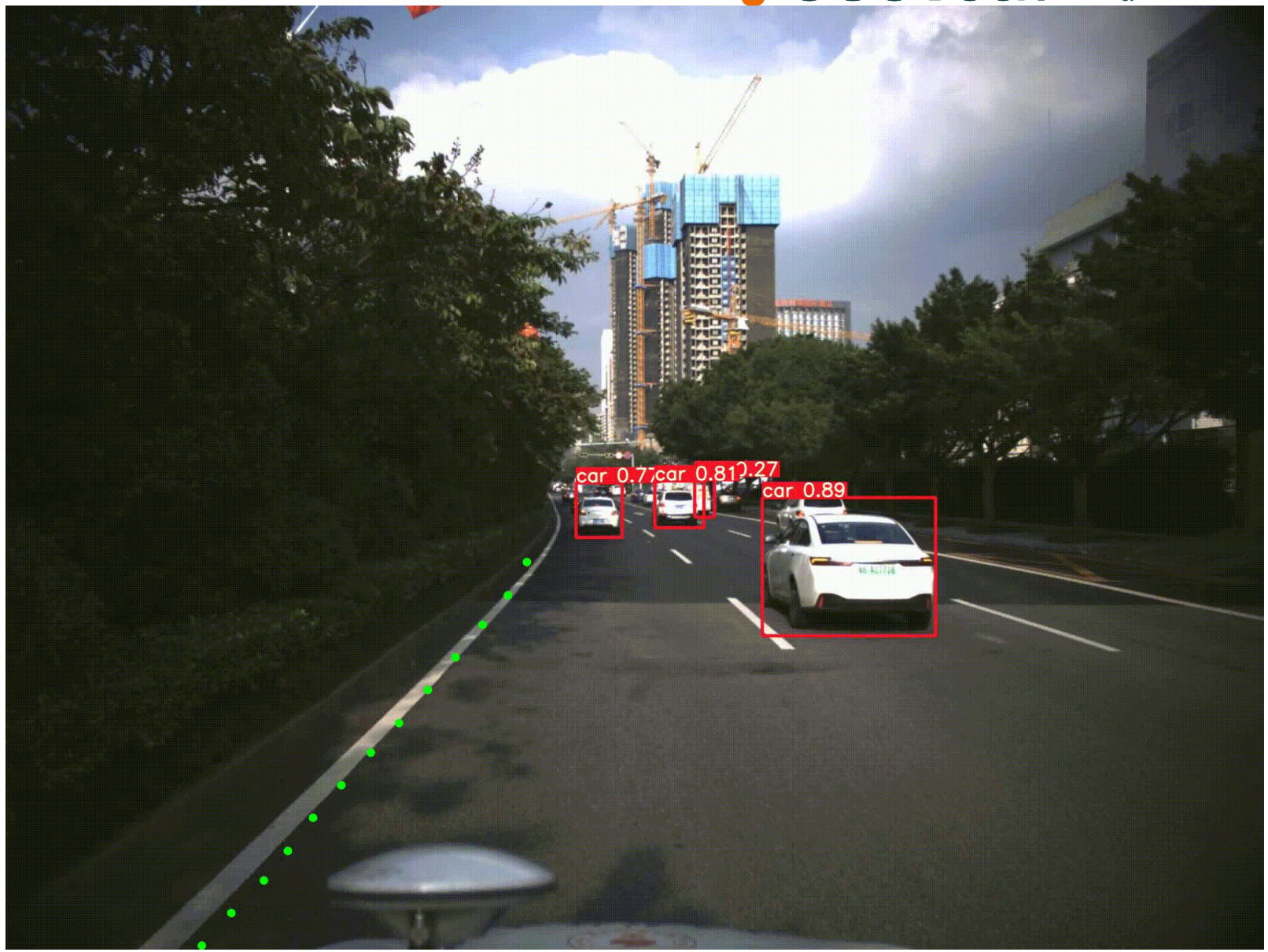
Adaptive structure loss

- Second order difference
- First order difference



Final Result

SUSTech Southern University of Science and Technology



SUSTech Southern University of Science and Technology

Thanks