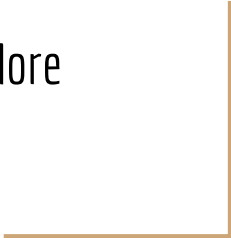




License Plate Recognition

Shishir M. Roy
M.Tech AI, IISc Bangalore



Dataset

- For initial experiments, we have chosen Chinese City Parking Dataset (CCPD).
- The dataset is divided into many sub-datasets, whose description is on the next slide, one of which is CCPD_Base which has 200k images.
- CCPD_Base is split into train-validation-test set in the ratio 5:1:4 and other sub-datasets are used as additional test sets.
- Each image in the dataset has only one License Plate (LP).
- Each LP number is comprised of a Chinese character, followed by a letter, and then five letters or numbers.

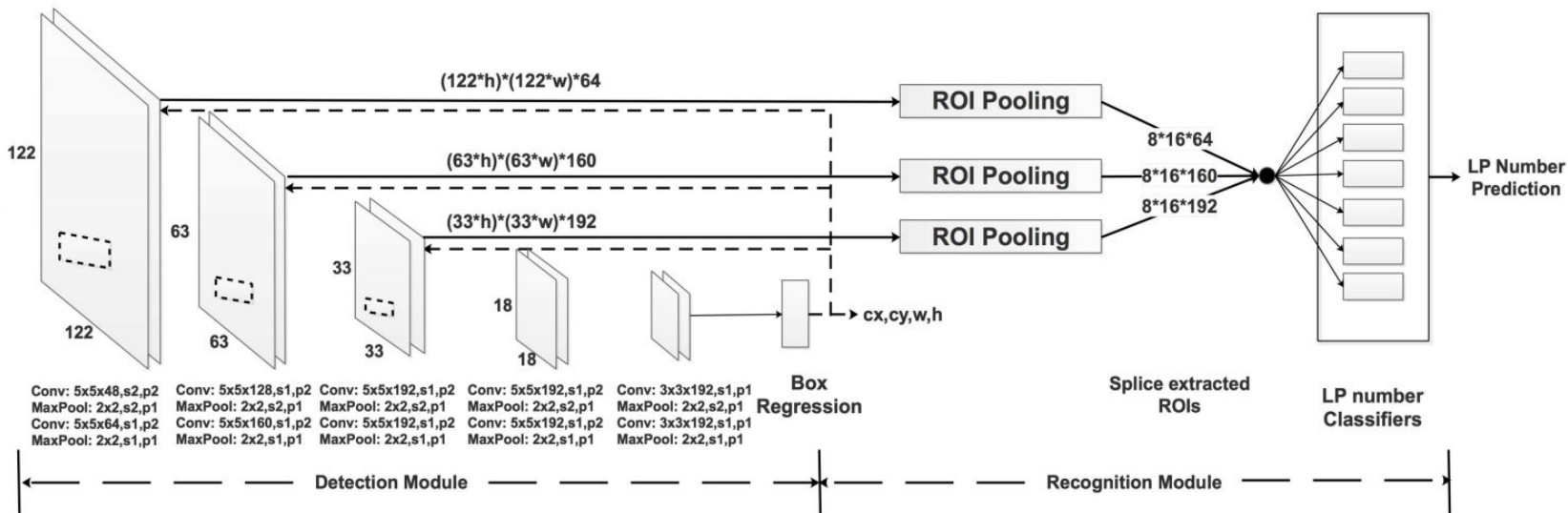
	Description
CCPD-Base	The only common feature of these photos is the inclusion of a license plate.
CCPD-DB	Illuminations on the LP area are dark, uneven or extremely bright.
CCPD-FN	The distance from the LP to the shooting location is relatively far or near.
CCPD-Rotate	Great horizontal tilt degree ($20^{\circ} \sim 50^{\circ}$) and the vertical tilt degree varies from -10° to 10° .
CCPD-Tilt	Great horizontal tilt degree ($15^{\circ} \sim 45^{\circ}$ degrees) and vertical tilt degree ($15^{\circ} \sim 45^{\circ}$).
CCPD-Blur	Blurry largely due to hand jitter while taking pictures.
CCPD-Weather	Images taken on a rainy day, snow day or fog day.
CCPD-Challenge	The most challenging images for LPDR to date.
CCPD-NP	Images of new cars without a LP.

Subset	CCPD_Base	CCPD_FN	CCPD_Tilt	CCPR_Weather
Number of Images	200k	21k	30k	10k
Subset	CCPD_DB	CCPD_Rotate	CCPD_Blur	CCPD_Challenge
Number of Images	10k	10k	20.5k	50k

Details of different sub-datasets in CCPD

Roadside Parking Net (RPNet)

RPNet Architecture



- Let **gb** be the ground truth bounding box and **pb** be the predicted bounding box.

$$L_{loc}(pb, gb) = \sum_N \sum_{m \in \{cx, cy, w, h\}} smooth_{L1}(pb^m - gb^m)$$

- Let **pn_i** ($1 \leq i \leq 7$) be the ground truth LP numbers and **pn_i** ($1 \leq i \leq 7$) be the predictions for LP numbers, each containing nc_i floats.

$$L_{cls}(pn, gn) = \sum_N \sum_{1 \leq i \leq 7} \{-pn_i[gn_i] + \log(\sum_{1 \leq j \leq nc_i} \exp(pn_i[j]))\}$$

- Model can be trained end to end by jointly optimizing both the losses.
- Moreover, pretraining the detection module helps the final model to converge faster.

- Since the detection and recognition modules share the feature maps, this architecture is faster compared to the usual two step solutions - detection and recognition.
- ROI Pooling -
 - Once the detection module predicts the bounding box, we can extract the feature maps from Region Of Interest (ROI) from the output of 2nd, 4th and 6th conv layers.
 - The ROI Pooling divides the ROI features into $h_{out} \times w_{out}$ grids and returns the max from those grids (perform max pooling).
- Number of Parameters in our implementation = 55, 073, 818.

Results

Sub-dataset	Paper Accuracy (300 epochs)	Our Accuracy (60 epochs)
CCPD_Base Test	98.5	95.17
CCPD_DB	96.9	55.35
CCPD_FN	94.3	50.81
CCPD_Rotate	90.8	73.97
CCPD_Tilt	92.5	67.54
CCPD_Weather	87.9	90.67
CCPD_Challenge	85.1	43.94
CCPD_Blur	-	47.17

	FPS	AP	Base(100k)	DB	FN	Rotate	Tilt	Weather	Challenge
Cascade classifier + HC	29	58.9	69.7	67.2	69.7	0.1	3.1	52.3	30.9
SSD300 + HC	35	95.2	98.3	96.6	95.9	88.4	91.5	87.3	83.8
YOLO9000 + HC	36	93.7	98.1	96.0	88.2	84.5	88.5	87.0	80.5
Faster-RCNN+ HC	13	92.8	97.2	94.4	90.9	82.9	87.3	85.5	76.3
TE2E	3	94.4	97.8	94.8	94.5	87.9	92.1	86.8	81.2
RPnet	61	95.5	98.5	96.9	94.3	90.8	92.5	87.9	85.1

Comparison of performance of various models on CCPD

- RPNet and TE2E models are capable of LP detection and recognition both, and can be trained end to end.
- All other models are two step ie one of the sub-models detect the LP and the other one Recognizes the LP characters.
- HC refers to Holistic CNN [2].

Stacked Attention Network (SAN)

ResNet 18 Architecture

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	$7 \times 7, 64$, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Layer 0

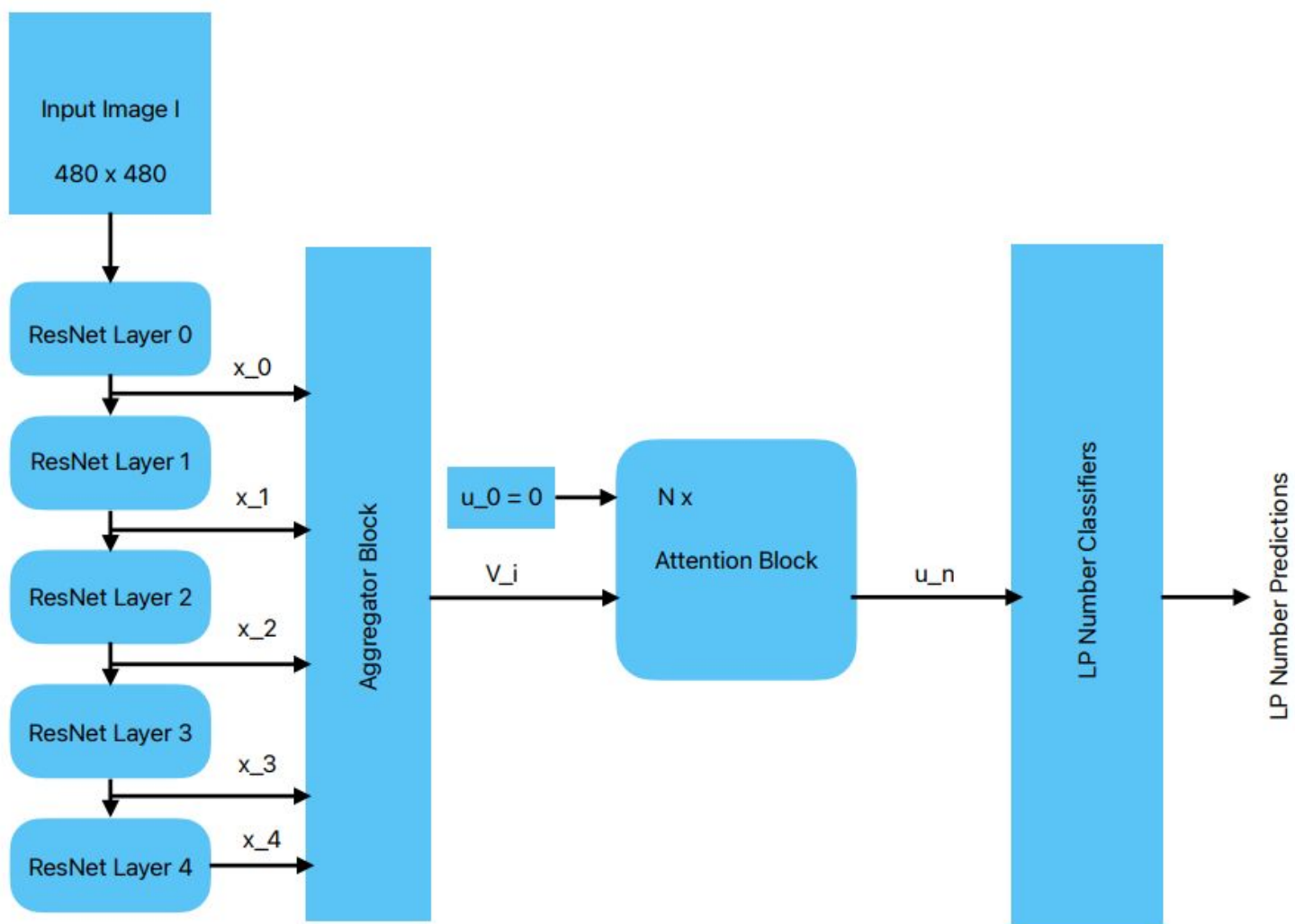
Layer 1

Layer 2

Layer 3

Layer 4

SAN Architecture



- Let $I \in \mathbb{R}^{3 \times 480 \times 480}$ be the image.
- Let $F_0(), F_1(), F_2(), F_3(), F_4()$ be the blocks of ResNet18.
- $x_0 = F_0(I), x_1 = F_1(x_0), x_2 = F_2(x_1), x_3 = F_3(x_2), x_4 = F_4(x_3)$.
- $x_0 \in \mathbb{R}^{64 \times 120 \times 120}, x_1 \in \mathbb{R}^{64 \times 120 \times 120}, x_2 \in \mathbb{R}^{128 \times 60 \times 60},$
 $x_3 \in \mathbb{R}^{256 \times 30 \times 30}, x_4 \in \mathbb{R}^{512 \times 15 \times 15}.$
- Define $A(z_1, z_2, pool)$ an aggregator function that takes $z_1, z_2 \in \mathbb{R}^{c \times h \times w}$ and returns $z \in \mathbb{R}^{2c \times \frac{h}{2} \times \frac{w}{2}}$ if $pool = \text{True}$, otherwise returns $z \in \mathbb{R}^{2c \times h \times w}.$
- After aggregating all x'_i s we get $X \in \mathbb{R}^{1024 \times 15 \times 15}.$

- We can treat the $X \in \mathbb{R}^{1024 \times 15 \times 15}$ as 15×15 regions of the image and 1024-sized feature vector for that region.
- Resize X such that $X \in \mathbb{R}^{1024 \times 255}$.
- Let $V_i = \tanh(W_1 \times X + b_1)$ be the embeddings from the image.
- Attention scores can be calculated as follows -

$$p_1 = \sigma(w_3 \times \tanh(W_2 \times V_i + b_2) + b_3)$$

- Now $u_1 = p_1 \times V_i$ is the vector that attends to the relevant regions of the image. This is the first level of attention.
- $W_1 \in \mathbb{R}^{1024 \times 1024}$, $W_2 \in \mathbb{R}^{1024 \times 1024}$, $w_3 \in \mathbb{R}^{1 \times 1024}$, $p_1 \in \mathbb{R}^{1 \times 255}$.

- This attention can be applied multiple times on top of one another, it's just we have to add the \mathbf{u}_i vector to all the image embeddings V_i .
- Feed the final \mathbf{u}_i to the classifiers to get the outputs.
- This is called Luong Attention [3] and has been successfully used for Image Captioning and Visual Question Answering [4] tasks.
- Number of Parameters in our implementation = 16, 326, 064.

Results

Sub-dataset	RPNet	SAN
CCPD_Base Test	95.17	97.17
CCPD_DB	55.35	35.35
CCPD_FN	50.81	30.81
CCPD_Rotate	73.97	56.97
CCPD_Tilt	67.54	47.54
CCPD_Weather	90.67	93.67
CCPD_Challenge	43.94	33.94
CCPD_Blur	47.17	27.17

Further Work - I

- Analysing the effect of varying -
 - Number of attention blocks.
 - Backbone model.
 - Layer outputs of backbone to be aggregated.

on the performance.

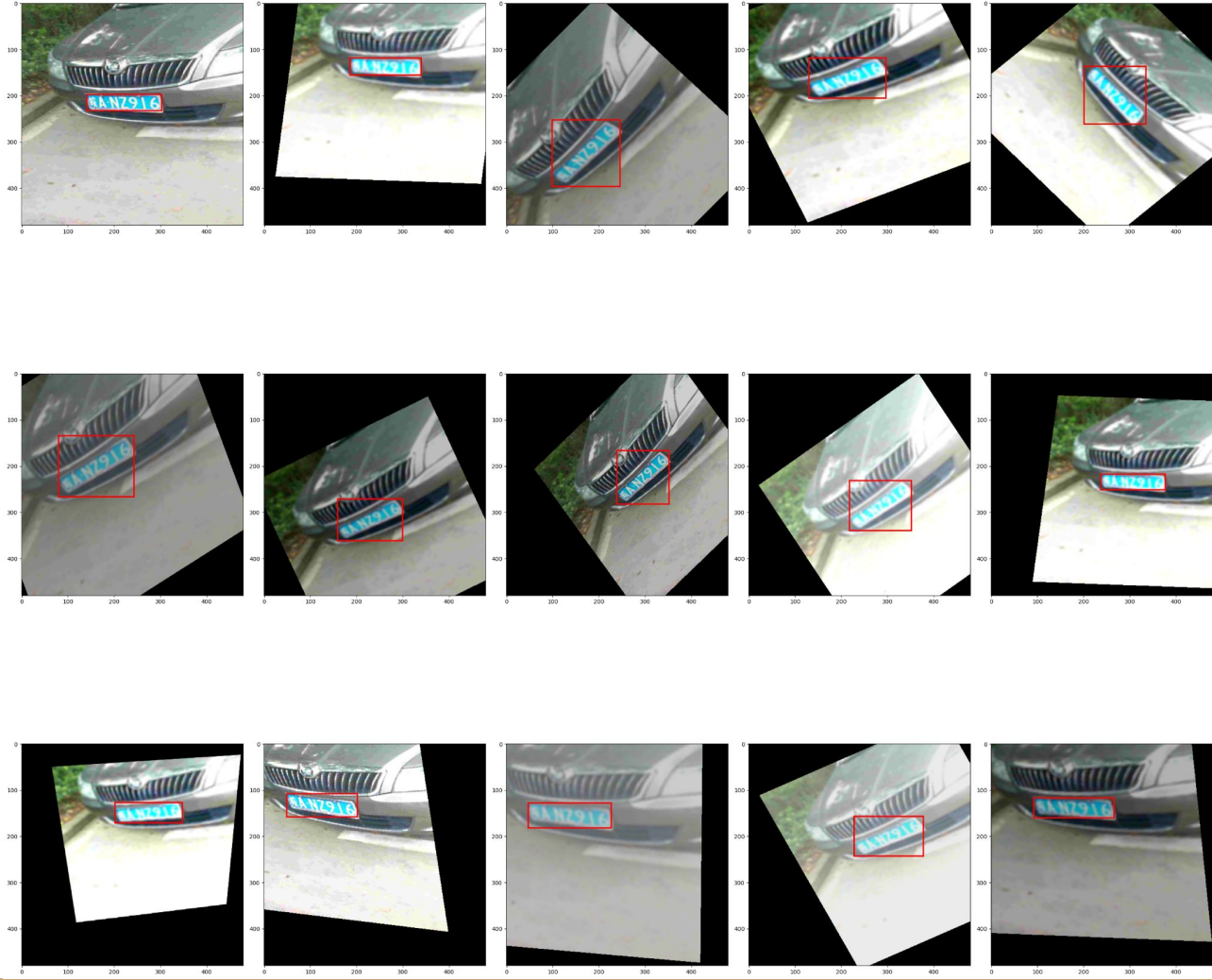
- Experiment with different Aggregator Functions.

Further Work - II

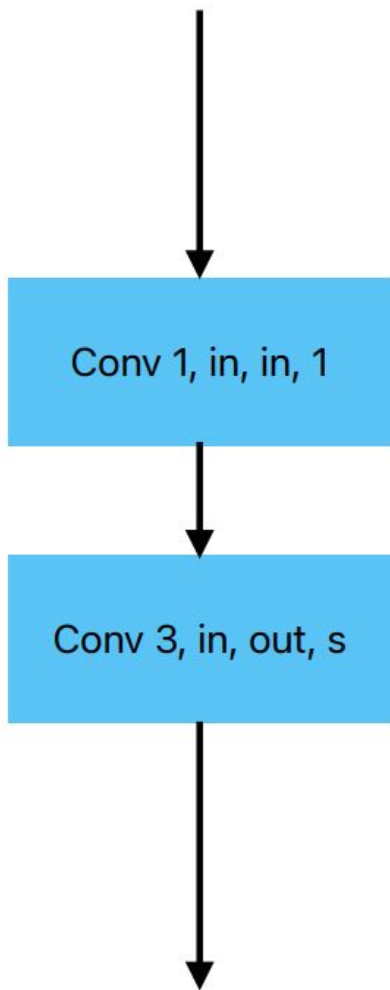
- In RPnet, approximately 90% of the parameters are concentrated in the box regressor and the LP classifiers.
- To optimize this, we can implement a Squeeze-and-Excitation (SE) block followed by a 1×1 convolution before the box regressor and LP classifiers.
- This approach will allow us to reduce the number of channels before the box regressor and LP classifiers, consequently decreasing the overall parameter count.
- As a result, we can allocate more output channels in the convolutional network, enabling it to learn richer and more diverse features.

RPNet V2

Data Augmentation Samples



Basic Block in, out, s



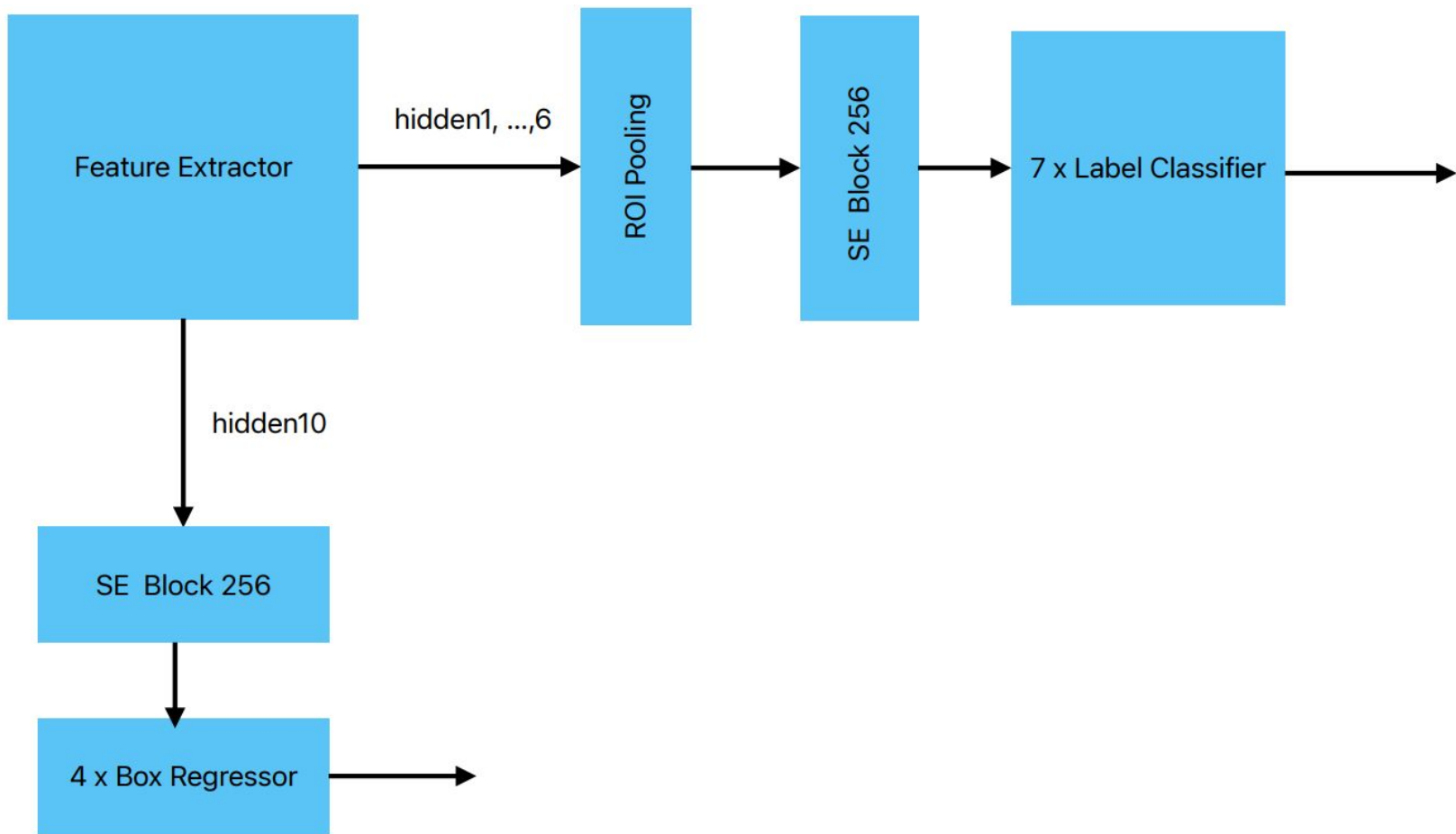
- Conv k, in, out, s

Where k = kernel size, in = in channels, out = out channels, s = stride

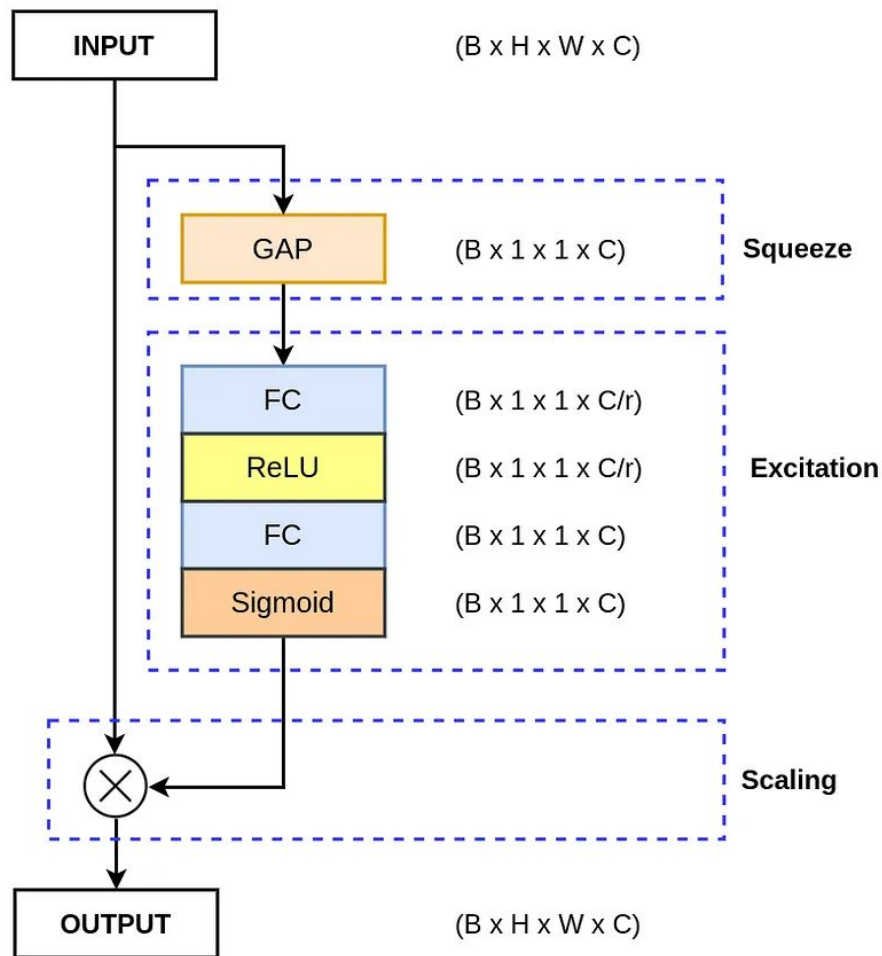
Feature Extractor

Layer Name	rpnet_v2
hidden0	Conv 7, 3, 64, 2
hidden1	Basic Block 64, 96, 2
hidden2	Basic Block 96, 96, 1
hidden3	Basic Block 96, 128, 2
hidden4	Basic Block 128, 128, 1
hidden5	Basic Block 128, 192, 2
hidden6	Basic Block 192, 192, 1
hidden7	Basic Block 192, 256, 2
hidden8	Basic Block 256, 384, 2
hidden9	Basic Block 384, 512, 2
hidden10	Basic Block 512, 768, 2

RPNet V2 Architecture



Squeeze and Excitation Block



- RpNet v2 has 11 Conv Layers and Squeeze and Excitation Blocks before the box regressor and label classifiers.
- Total number of parameters = 28, 617, 035.
- Number of parameters in box regressor = 526, 336.
- Number of parameters in label classifiers = 16, 899, 072.
- Images per second = 14.46 (CPU)
- Images per second = 121.08 (GPU A100)
- Memory Requirement for inference = 704 MB

Results

Sub-dataset	RPNet	RPNet v2
CCPD_Base Test	95.17	99.45
CCPD_DB	55.35	85.71
CCPD_FN	50.81	81.24
CCPD_Rotate	73.97	94.70
CCPD_Tilt	67.54	87.73
CCPD_Weather	90.67	97.32
CCPD_Challenge	43.94	83.23
CCPD_Blur	47.17	86.54

Some Output Samples

Original



Predicted



Some Output Samples

Original

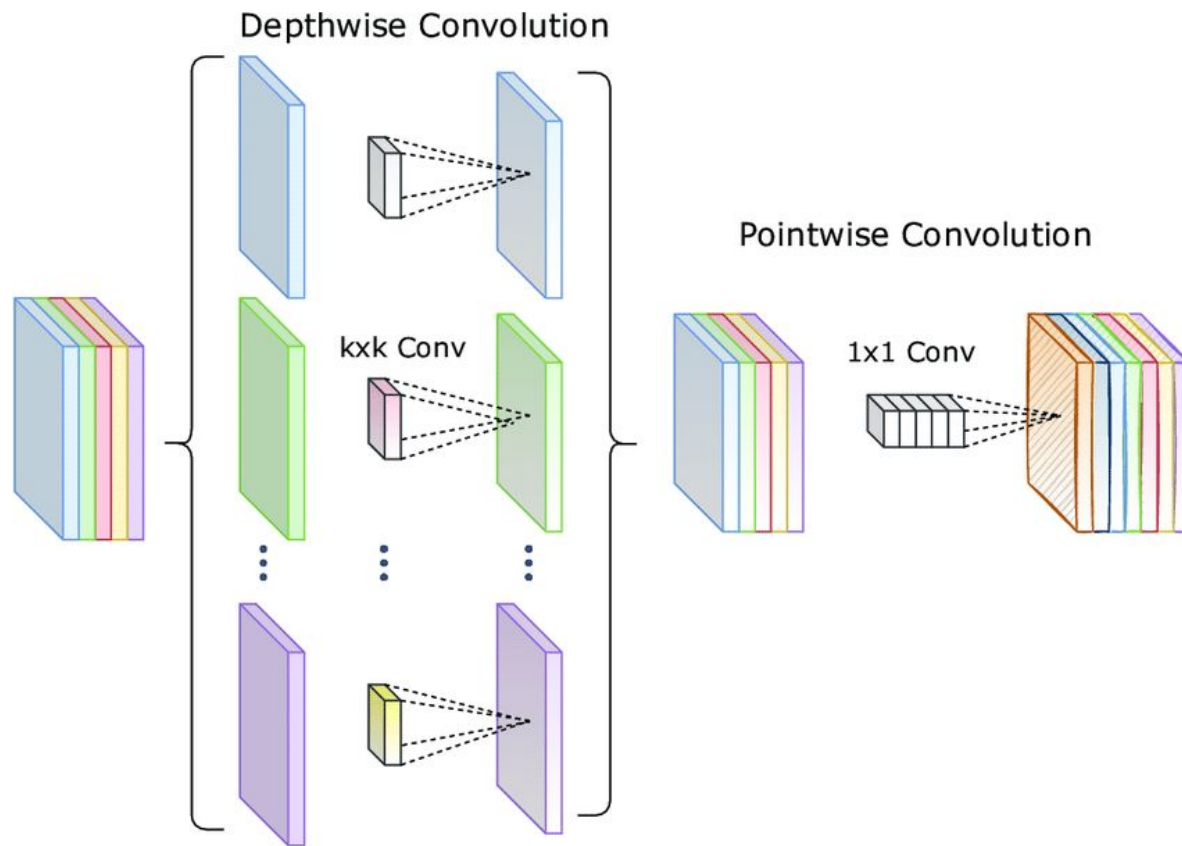


Predicted



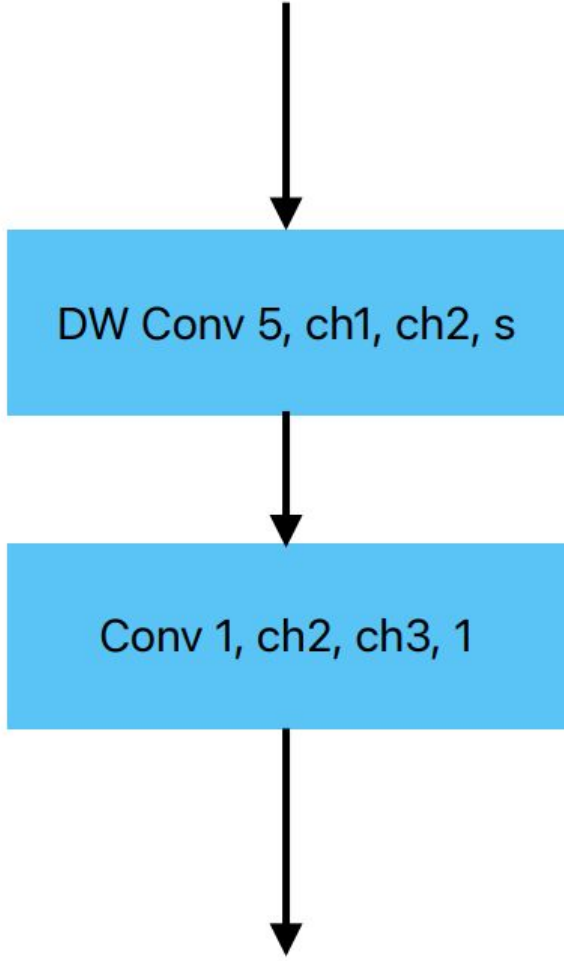
RPNet V3

Depthwise Separable Convolution

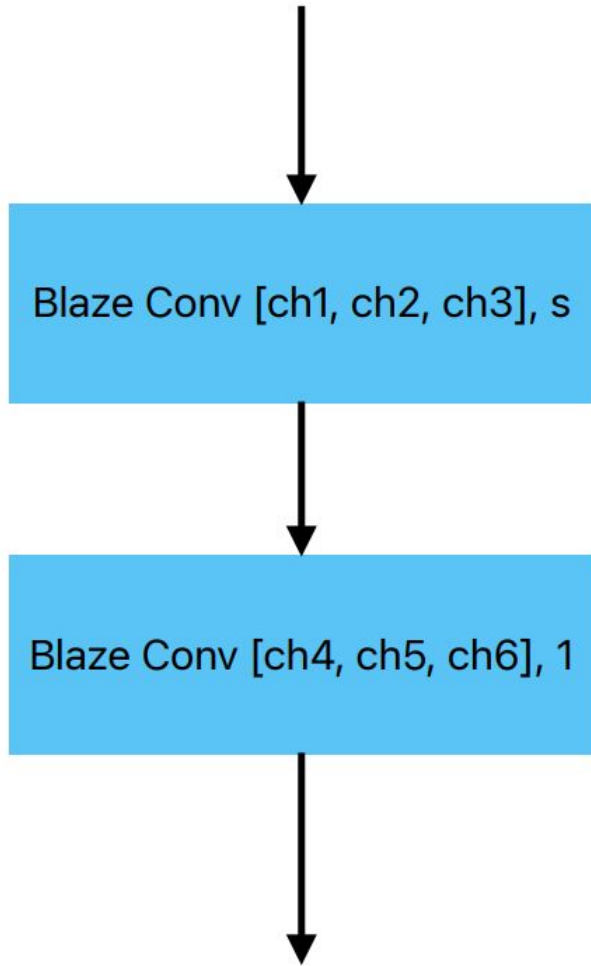


- Assume we have input of size $3 \times 16 \times 16$ and want the output to be $64 \times 16 \times 16$.
- By using vanilla convolution of kernel size 5×5 , we need -
 - $(16 \times 16 \times 64) \times (5 \times 5 \times 3) = 1,228,800$ multiplication operations.
 - $(16 \times 16 \times 64) \times (5 \times 5 \times 3 - 1) = 1,212,416$ addition operations.
 - $(3 \times 5 \times 5 \times 64) + (64) = 4,864$ parameters.
- By using depthwise convolution of kernel size 5×5 , we need -
 - $(16 \times 16 \times 3) \times (5 \times 5) = 19,200$ multiplication operations in depthwise convolution.
 - $(16 \times 16 \times 3) \times (5 \times 5 - 1) = 18,432$ addition operations in depthwise convolution.
 - $(16 \times 16 \times 64) \times (1 \times 1 \times 3) = 49,152$ multiplication operations in pointwise convolution.
 - $(16 \times 16 \times 64) \times (1 \times 1 \times 3 - 1) = 32,768$ addition operations in pointwise convolution.
 - $3 \times (1 \times 5 \times 5 \times 1 + 1) + (3 \times 1 \times 1 \times 64 + 64) = 334$ parameters.
- There is a reduction of -
 - 24x in number of addition operations.
 - 19x in number of multiplication operations.
 - 15x in number of parameters.

Blaze Conv [ch1, ch2, ch3], s



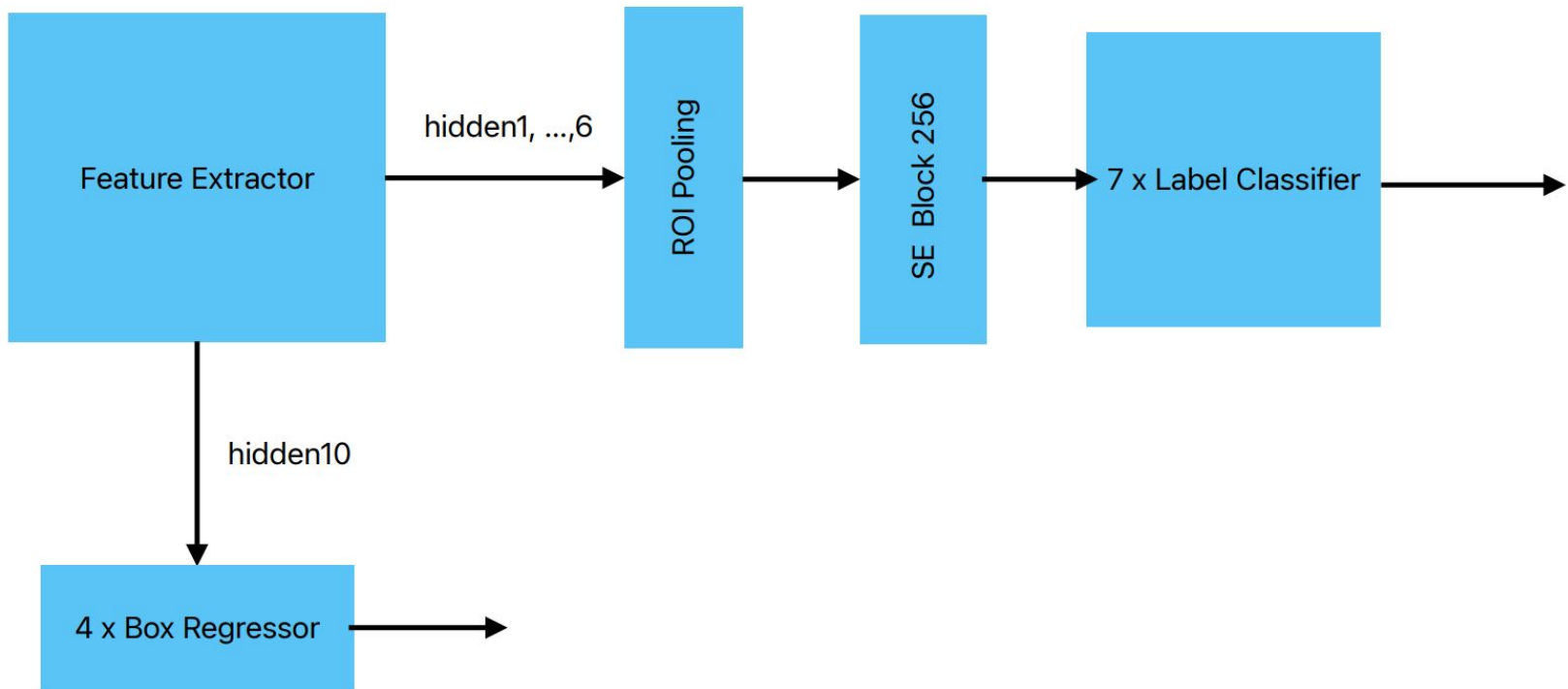
Blaze Block [ch1, ..., ch6], s



Feature Extractor

Layer Name	rpnet_v3
hidden0	Conv 7, 3, 64, 2
hidden1	BlazeBlock [64, 64, 32, 32, 32, 64], 2
hidden2	BlazeBlock [64, 64, 32, 32, 32, 96], 1
hidden3	BlazeBlock [96, 96, 48, 48, 48, 96], 2
hidden4	BlazeBlock [96, 96, 48, 48, 48, 128], 1
hidden5	BlazeBlock [128, 128, 64, 64, 64, 128], 2
hidden6	BlazeBlock [128, 128, 64, 64, 64, 192], 1
hidden7	BlazeBlock [192, 192, 96, 96, 96, 192], 2
hidden8	BlazeBlock [192, 192, 96, 96, 96, 256], 2
hidden9	BlazeBlock [256, 256, 128, 128, 128, 256], 2
hidden10	BlazeBlock [256, 256, 128, 128, 128, 384], 2

RPNET V3 Architecture



- RpNet v3 has 11 Conv Layers and Squeeze and Excitation Blocks before the label classifiers.
- Total number of parameters = 19, 759, 659.
- Number of parameters in box regressor = 1, 050, 624.
- Number of parameters in label classifiers = 16, 899, 072.

Results

Sub-dataset	RPNet v2	RPNet v3
CCPD_Base Test	99.45	97.53
CCPD_DB	85.71	65.55
CCPD_FN	81.24	54.22
CCPD_Rotate	94.70	80.55
CCPD_Tilt	87.73	69.97
CCPD_Weather	97.32	94.25
CCPD_Challenge	83.23	61.05
CCPD_Blur	86.54	64.75

RPNet V4

- The models that we have seen so far predicts the LP numbers in single step, balancing performance and efficiency.
- Using larger images increases computation time, affecting model efficiency.
- Using smaller images may result in a loss of information, impacting LP recognition accuracy.
- With this in mind, we can separate the detection and recognition such that we perform LP detection on a smaller image and LP recognition on a larger crop of LP.

RPNET V4 - Detector

Name	Input	Layer/Block
hidden0	$3 \times 384 \times 238$	Conv 7, 3, 48, 2
hidden1	$48 \times 192 \times 119$	Basic Block 48, 64, 2
hidden2	$64 \times 96 \times 60$	Basic Block 64, 96, 2
hidden3	$96 \times 48 \times 30$	Basic Block 96, 128, 2
hidden4	$128 \times 24 \times 15$	Basic Block 128, 192, 2
hidden5	$192 \times 12 \times 8$	Basic Block 192, 256, 2
hidden6	$256 \times 6 \times 4$	Basic Block 256, 384, 2
box regressor	$384 \times 3 \times 2$	-

RPNET V4 - Recognizer

Name	Input	Layer/Block
hidden0	$3 \times 64 \times 256$	Conv 7, 3, 64, 2
hidden1	$64 \times 32 \times 128$	Basic Block 64, 96, 2
hidden2	$96 \times 16 \times 64$	Basic Block 96, 128, 2
hidden3	$128 \times 8 \times 32$	Basic Block 128, 256, 1
hidden4	$256 \times 8 \times 32$	Conv (8, 7), 256, 256, (8, 1)
lp classifier	$256 \times 1 \times 32$	-

- Number of parameters in detector = 3, 071, 348.
- Number of parameters in recognizer = 8, 490, 094.
- Images per second = 135.13 (GPU RTX 3090)
- Memory Requirement for inference = 497 MB.

Results

Sub-dataset	RPNet v2	RPNet v3	RPNet v4
CCPD_Base Test	99.45	97.53	99.40
CCPD_DB	85.71	65.55	86.75
CCPD_FN	81.24	54.22	87.07
CCPD_Rotate	94.70	80.55	93.56
CCPD_Tilt	87.73	69.97	86.52
CCPD_Weather	97.32	94.25	97.90
CCPD_Challenge	83.23	61.05	84.07
CCPD_Blur	86.54	64.75	87.07