ICPR2020
25th INTERNATIONAL CONFERENCE
ON PATTERN RECOGNITION
Milan, Italy 10 | 15 January 2021

Compression of YOLOv3 via Block-wise and Channel-wise Pruning for Real-time and Complicated Autonomous Driving Environment Sensing Applications

Jiaqi Li, Yanan Zhao, Li Gao and Feng Cui

Beijing Institute of Technology and

Beijing Smarter Eye Technology Co. Ltd

1 Introduction

Challenges:

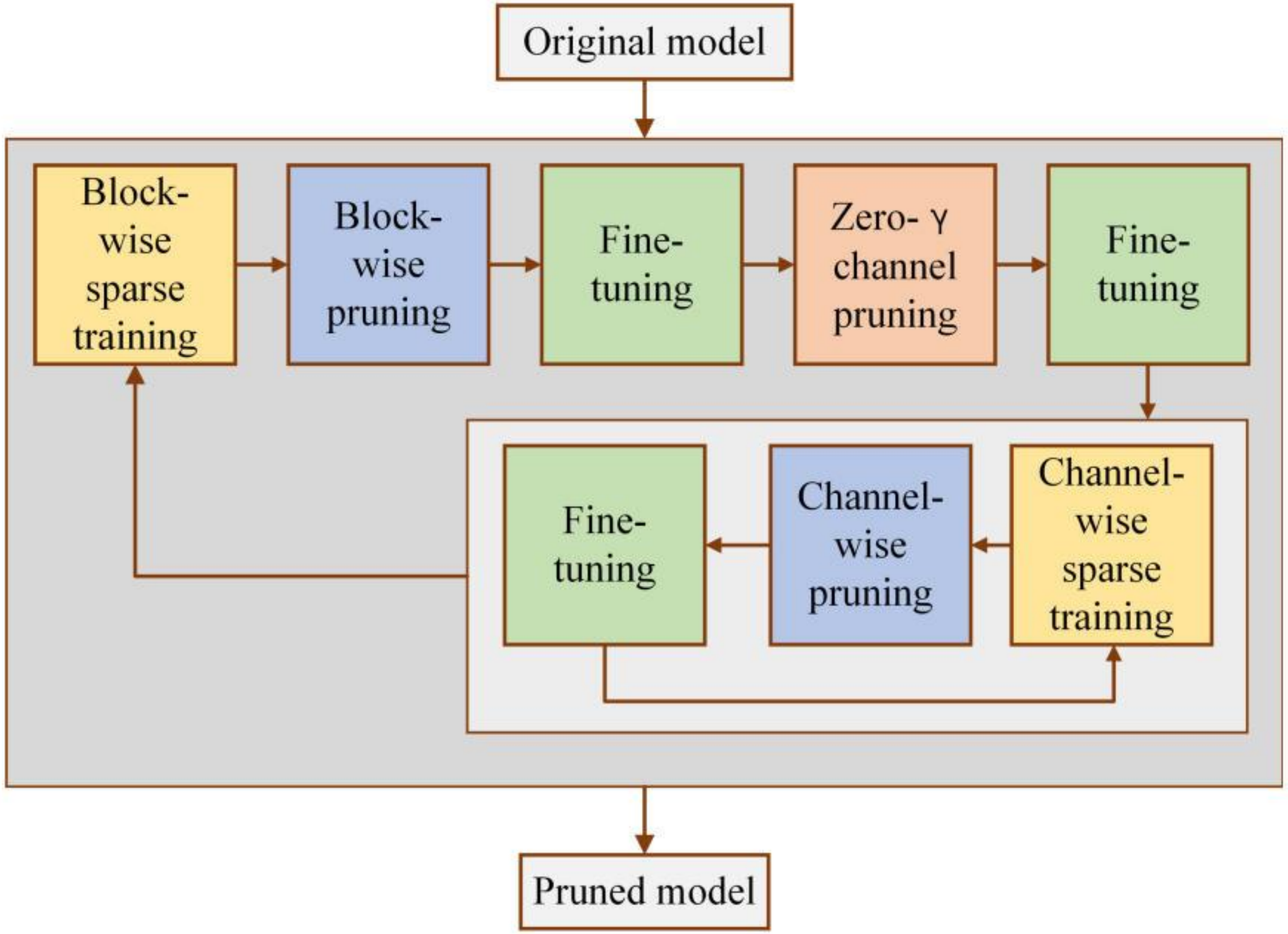
- the computational power of the object detectors is limited by the embedded devices on intelligent vehicles;
- the public datasets for autonomous driving are over-idealistic

Works:

- We propose a pipeline combining both block-wise pruning and channel-wise pruning to compress YOLOV3 model iteratively;
- We trained this model on our datasets which have more abundant and elaborate classes

2 Method

2.1 The pipeline for pruning YOLOV3 model

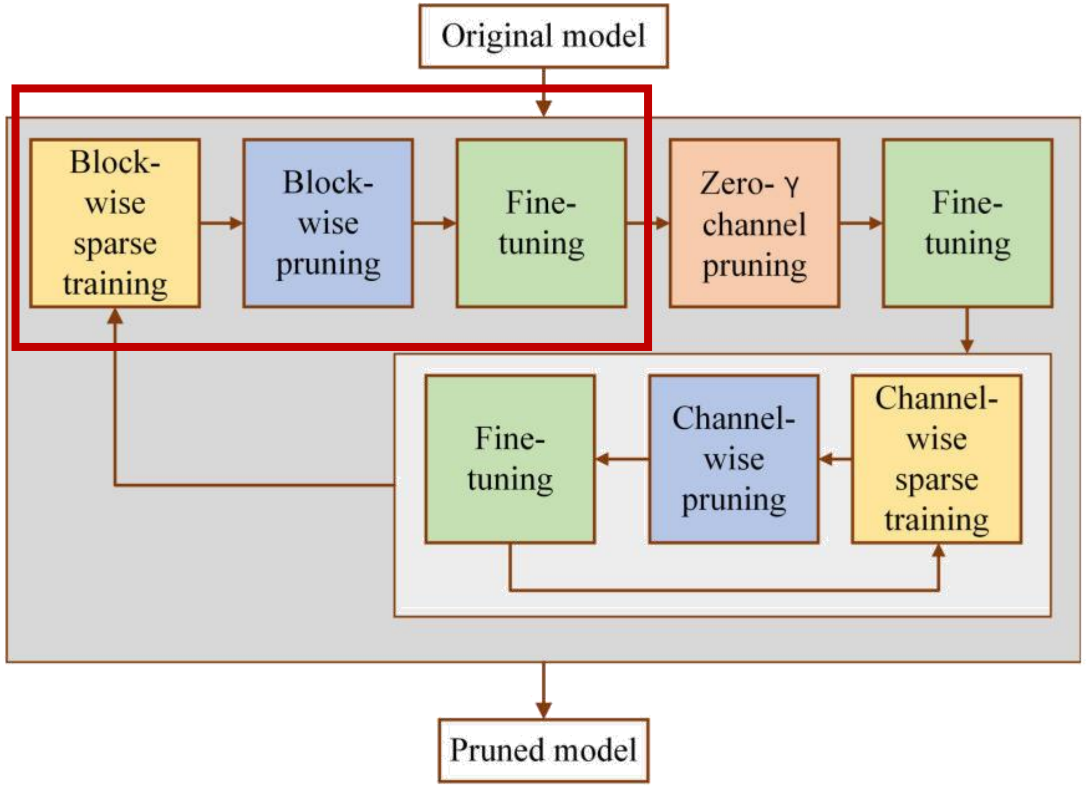


The block-wise pruning is performed iteratively and the channels are pruned iteratively after each block-wise pruning.

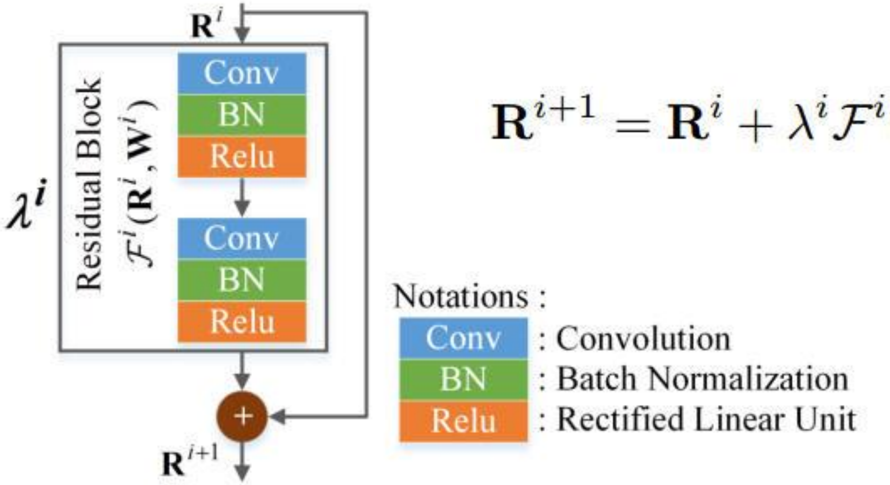
2 Method

2.1 Block-wise pruning

Sparsity training, Pruning and fine-tuning



Sparsity training:



$$\mathbf{R}^{i+1} = \mathbf{R}^i + \lambda^i \mathcal{F}^i(\mathbf{R}^i, \mathbf{W}^i)$$

Every residual block of the YOLOv3 model has two conv-bn-relu groups and a scale factor λ^i is added to multiply with the output of the residual block. The absolute value of λ^i represents the importance of the block.

$$\mathcal{L}_{bloreg} = \zeta \sum_{\lambda^i \in \Lambda} \left\| \lambda^i \right\|_1$$

L1 regularization → Sparse λ

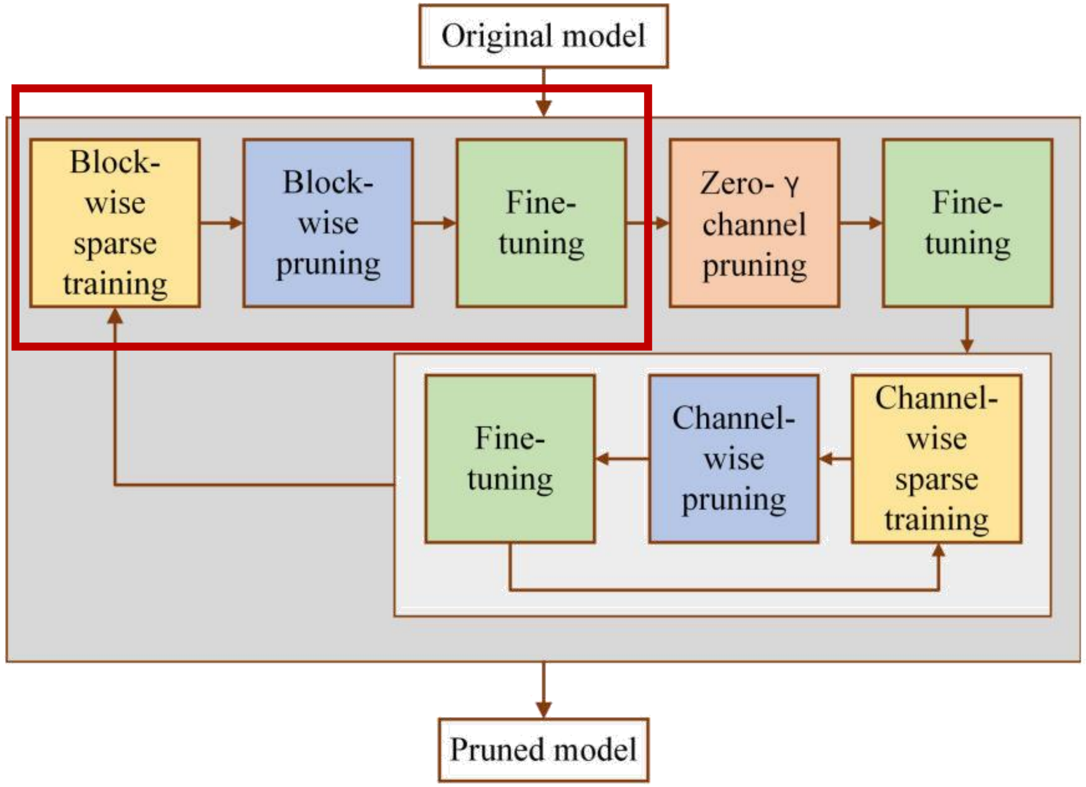
$$\text{Loss} = \mathcal{L}_{yolo} + \mathcal{L}_{bloreg}$$

FISTA

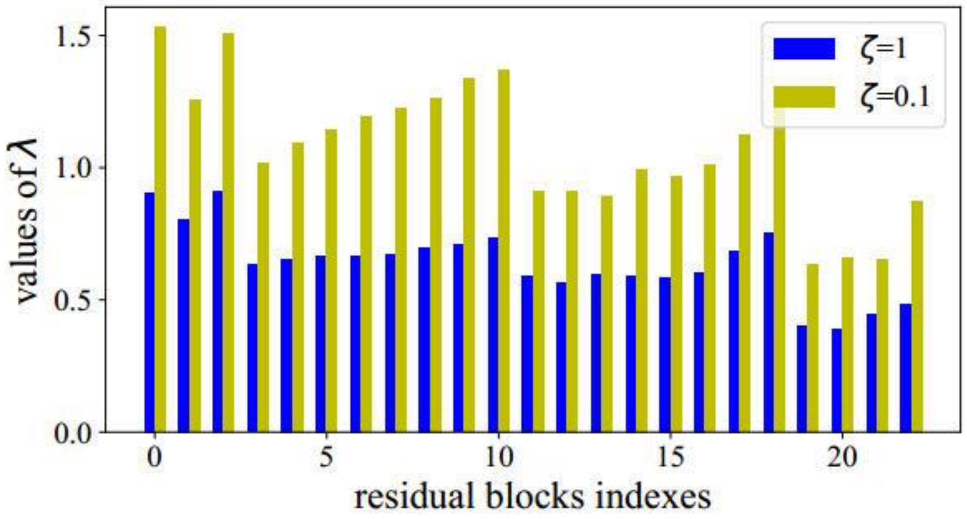
2 Method

2.1 Block-wise pruning

Sparsity training, Pruning and fine-tuning



Pruning and fine-tuning:



The distributions of λ with different ζ after sparsity training.

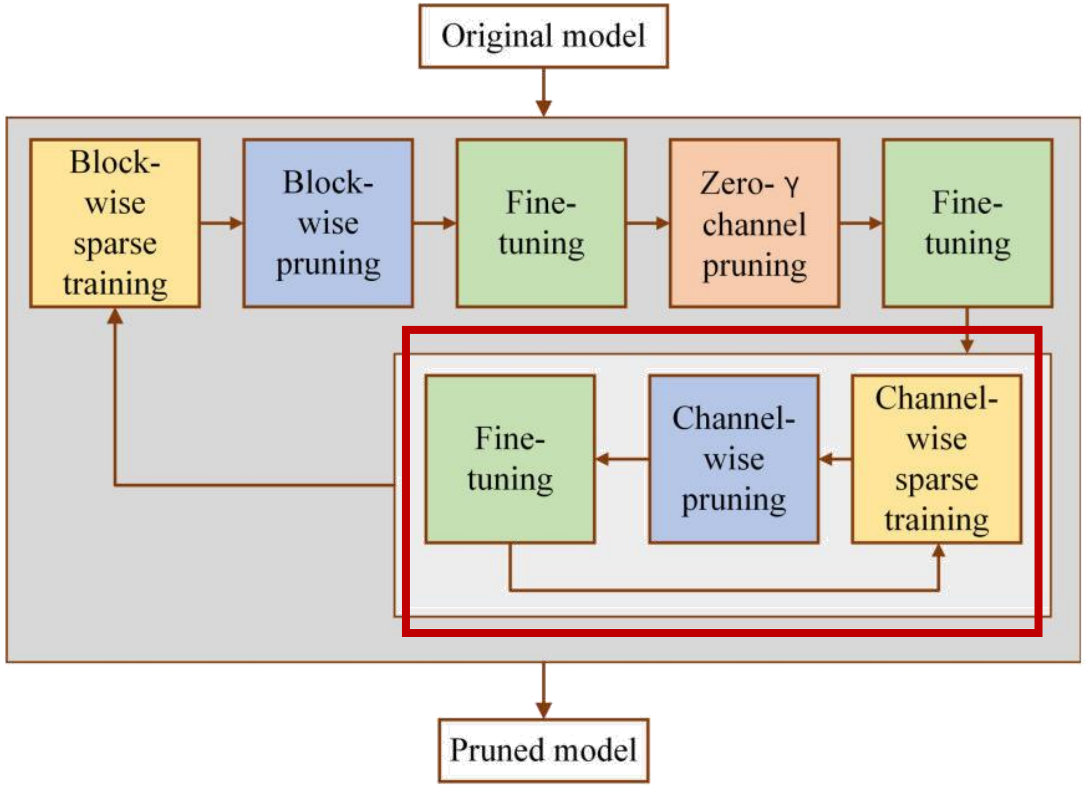
The importance of the residual blocks can be sorted according to $|\lambda^i|$. The residual blocks with smaller $|\lambda^i|$ can be removed entirely.

After pruning, the model should be fine-tuned.

2 Method

2.2 Channel-wise pruning

Sparsity training, Pruning and fine-tuning



Sparsity training:

$$b_{out}^{i,j} = \gamma^{i,j}\frac{b_{in}^{i,j}-\mu_{\mathcal{B}}^{i,j}}{\sqrt{\sigma_{\mathcal{B}}^{i,j}+\varepsilon}}+\beta^{i,j}$$

The absolute value $\left|\gamma^{i,j}\right|$ in each BN layer can reflect the importance of the channel.

$$\mathcal{L}_{chareg} = \xi \sum_{\gamma^{i,j}\in\Gamma} \left\|\gamma^{i,j}\right\|_1$$

L1 regularization ➡ Sparse γ

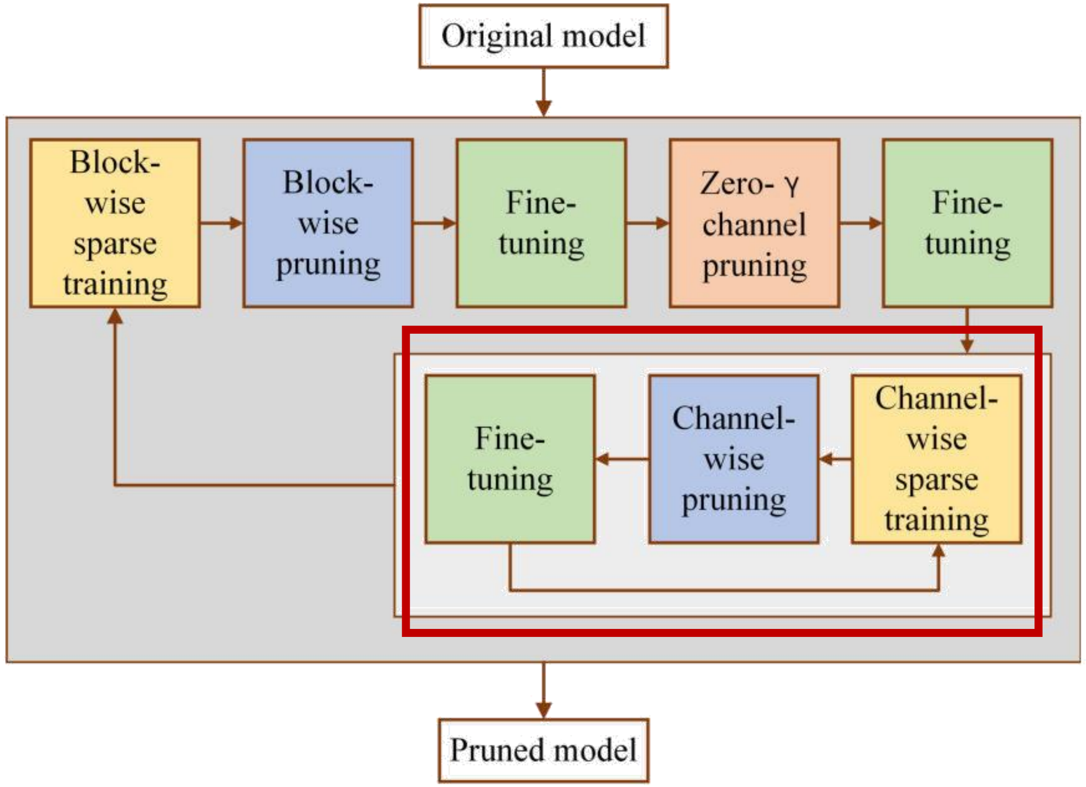
$$\text{Loss} = \mathcal{L}_{yolo} + \mathcal{L}_{chareg}$$

SGD

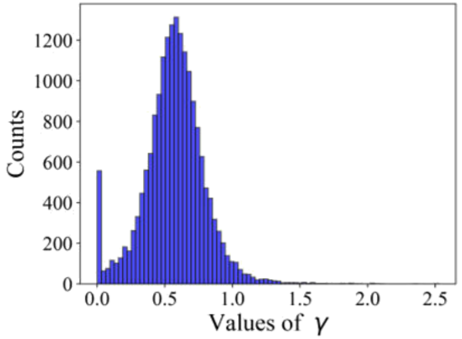
2 Method

2.2 Channel-wise pruning

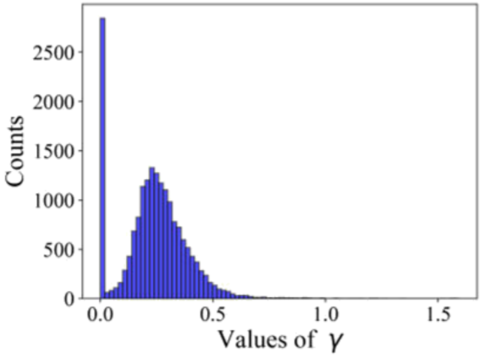
Sparsity training, Pruning and fine-tuning



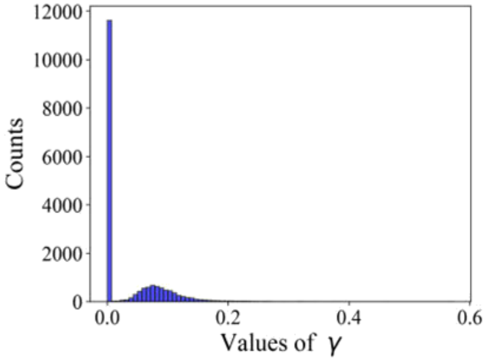
Pruning and fine-tuning:



(a)



(b)



(c)

Distributions of the scale factors γ in BN layers after channel-wise sparsity training with different ξ.

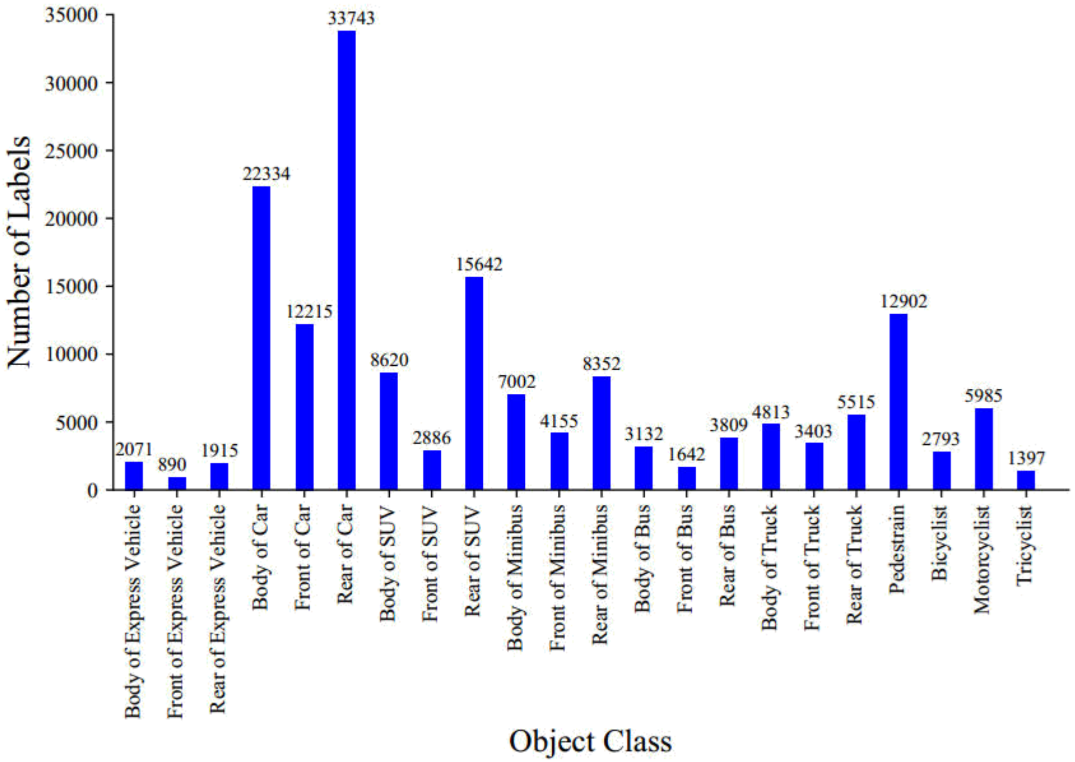
(a) $\xi = \mathbf{0.0001}$, (b) $\xi = \mathbf{0.001}$, (c) $\xi = \mathbf{0.01}$.

The less important channels of convolutional layers can be pruned according to the sorted sparse γ .

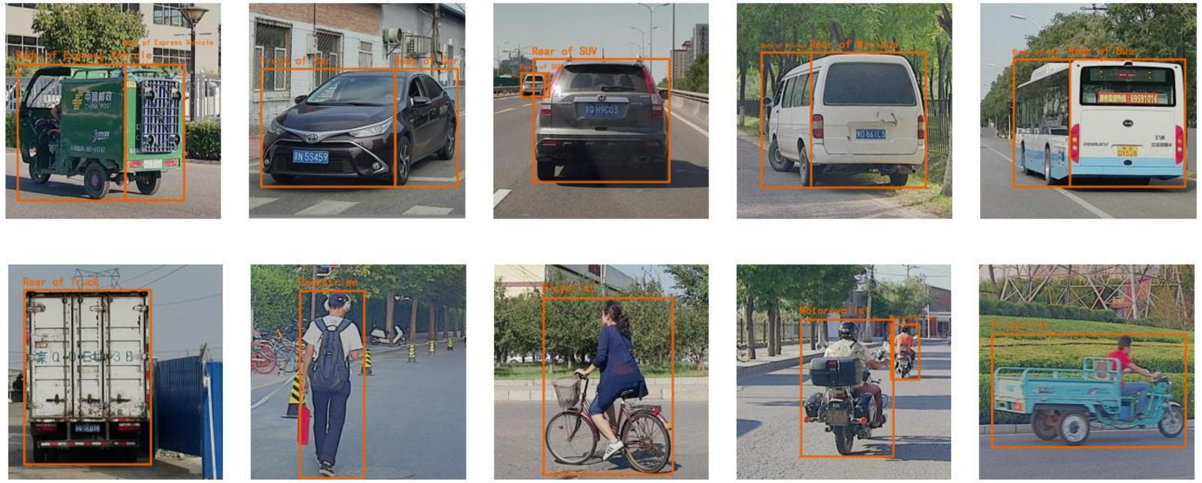
After pruning, the model should be fine-tuned.

3 Experiments and Results

3.1 Dataset



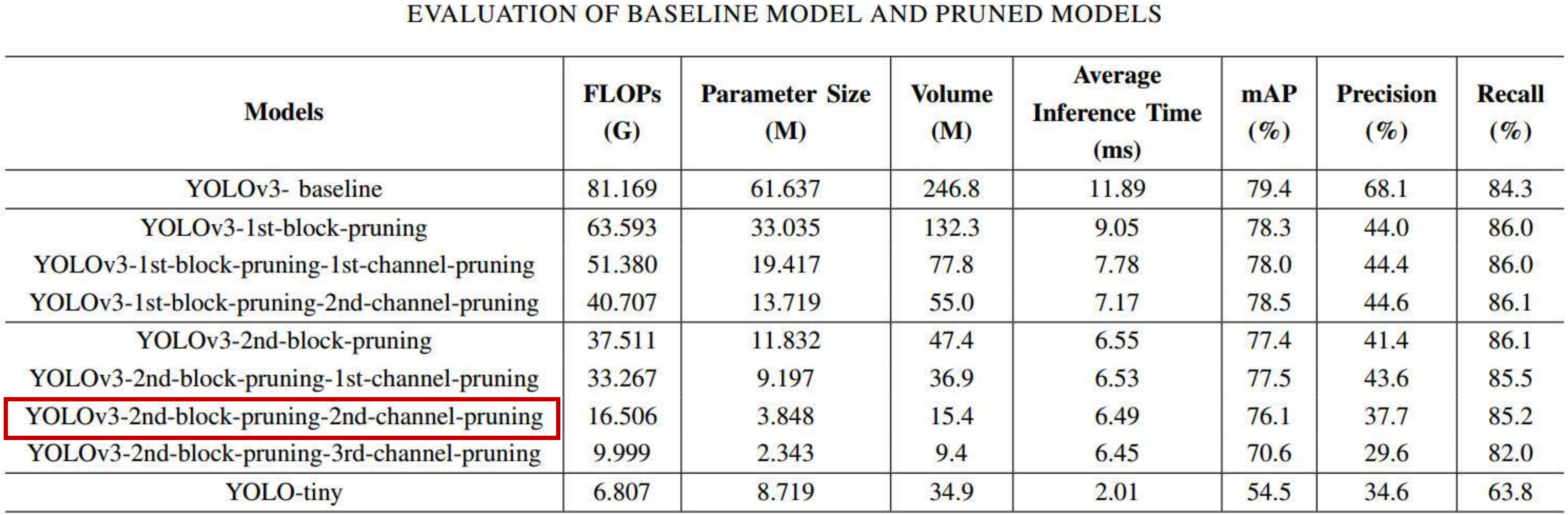
The classes in the datasets are defined as ‘Body of Express Vehicle’, ‘Front of Express Vehicle’, ‘Rear of Express Vehicle’, ‘Body of Car’, ‘Front of Car’, ‘Rear of Car’, ‘Body of SUV’, ‘Front of SUV’, ‘Rear of SUV’, ‘Body of Minibus’, ‘Front of Minibus’, ‘Rear of Minibus’, ‘Body of Bus’, ‘Front of Bus’, ‘Rear of Bus’, ‘Body of Truck’, ‘Front of Truck’, ‘Rear of Truck’, ‘Pedestrian’, ‘Bicyclist’, ‘Motorcyclist’ and ‘Tricyclist’.



There are 15,601 annotated static images including four kinds of scenes: freeway, urban road, suburb and residential area. Furthermore, the datasets also cover the scenes under poor illumination conditions such as the backlight scene.

3 Experiments and Results

3.2 experiments and results



EVALUATION OF BASELINE MODEL AND PRUNED MODELS

Models	FLOPs (G)	Parameter Size (M)	Volume (M)	Average Inference Time (ms)	mAP (%)	Precision (%)	Recall (%)
YOLOv3- baseline	81.169	61.637	246.8	11.89	79.4	68.1	84.3
YOLOv3-1st-block-pruning	63.593	33.035	132.3	9.05	78.3	44.0	86.0
YOLOv3-1st-block-pruning-1st-channel-pruning	51.380	19.417	77.8	7.78	78.0	44.4	86.0
YOLOv3-1st-block-pruning-2nd-channel-pruning	40.707	13.719	55.0	7.17	78.5	44.6	86.1
YOLOv3-2nd-block-pruning	37.511	11.832	47.4	6.55	77.4	41.4	86.1
YOLOv3-2nd-block-pruning-1st-channel-pruning	33.267	9.197	36.9	6.53	77.5	43.6	85.5
YOLOv3-2nd-block-pruning-2nd-channel-pruning	16.506	3.848	15.4	6.49	76.1	37.7	85.2
YOLOv3-2nd-block-pruning-3rd-channel-pruning	9.999	2.343	9.4	6.45	70.6	29.6	82.0
YOLO-tiny	6.807	8.719	34.9	2.01	54.5	34.6	63.8

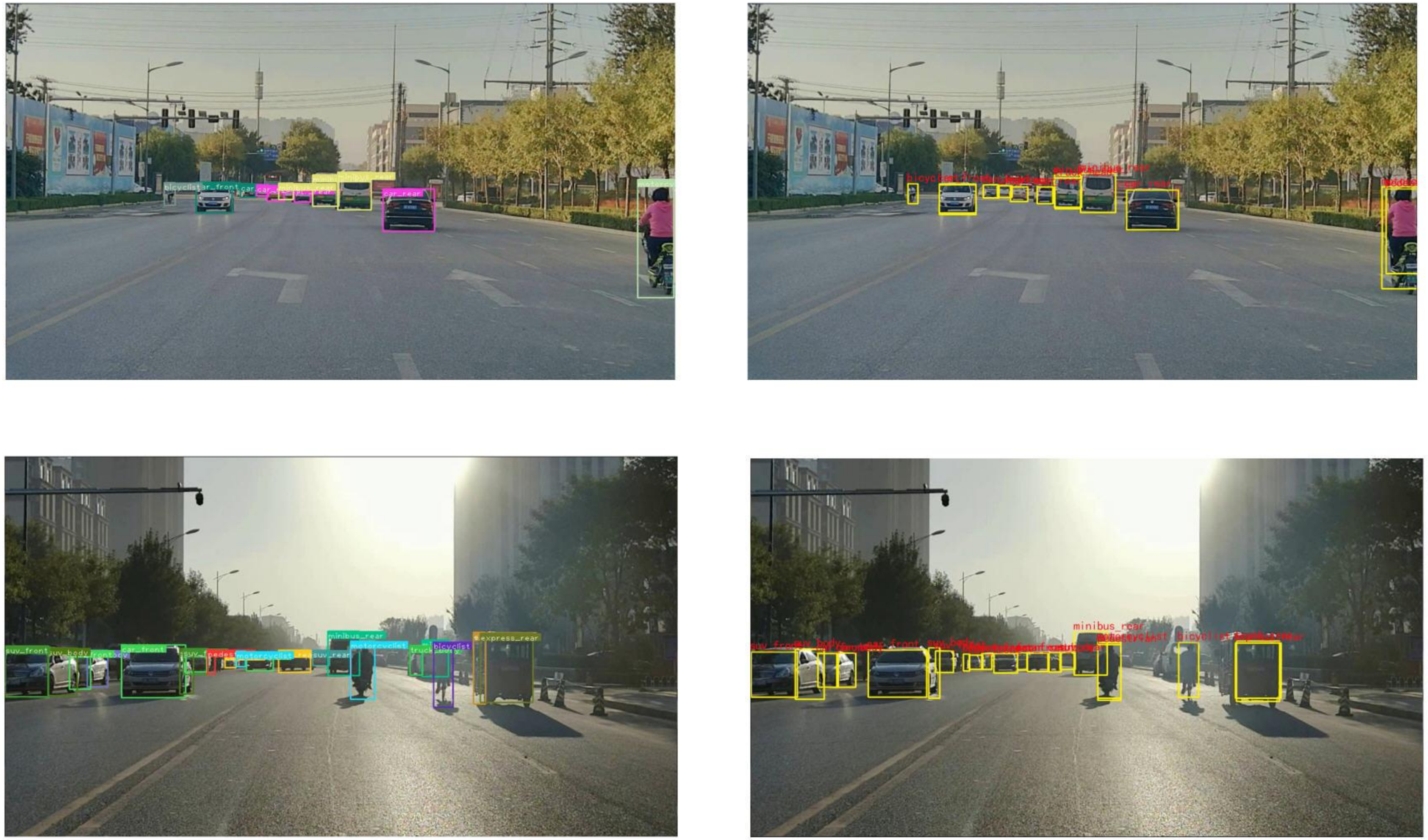
The performance on the validation set of all models during iterative pruning.

We choose YOLOv3-2nd-block-pruning-2nd-channel-pruning model as the final results of the experiments.

The final model save 79.7% FLOPs, reduce 93.8% parameter size, compress 93.8% model volumes as well as save 45.4% inference times, with only 4.16% mAP declines.

3 Experiments and Results

3.3 experiments and results



original model on server

pruned model on embedded device

The embedded device is a Xilinx® ZCU104 board with one B2304 core with 16 threads running at 330 MHz and DNNDK v3.0

The model can operate about 13 frames per second (FPS), which meets the needs of actual autonomous driving

Compared with the original model on the server, the mAP of the pruned model drops 4.53% of the mAP of the original model on the server

Thank You

Email: 3120180345@bit.edu.cn