温州肯恩大学

Hi

Wenzhou-Kean University

中美合作 全英文教学

Image Processing with Machine Learning

Gaurav Gupta & Shen Zheng

温州肯恩大学
WENZHOU-KEAN UNIVERSITY

溫州 – 上海 動車 3 小時，航程 1 小時
溫州 – 杭州 動車 2.5 小時
溫州 – 台灣 直航航程 1 小時
溫州 – 香港 直航航程 2 小時
溫州 – 廣州 直航航程 2 小時


温州城市夜景


雁荡山—国家5A级旅游区，史称中国“东南第一山”


南麂岛——国家级自然保护区


楠溪江——国家4A级旅游区、世界地质公园

建校历史

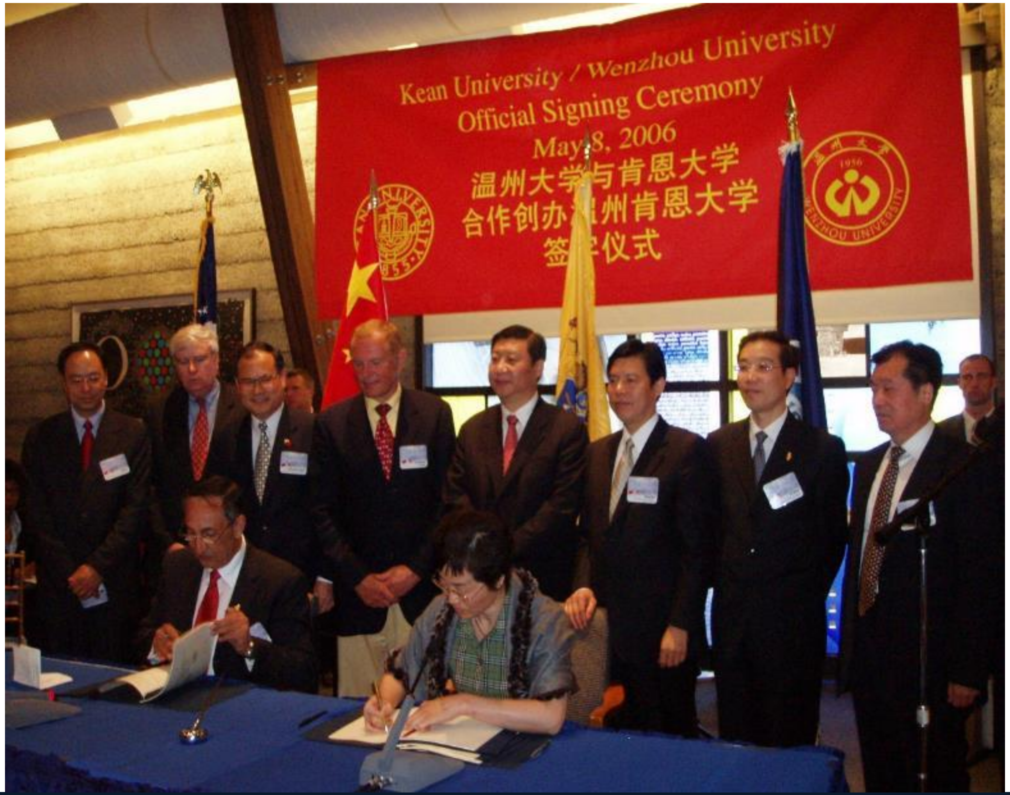
温州肯恩大学
WENZHOU-KEAN UNIVERSITY

2006年5月8日
中美双方签约创办温州肯恩大学

2011年11月16日
教育部批准筹建温州肯恩大学

2012年7月
以“中美合作肯恩项目”开始招生

2014年3月31日
教育部正式批准设立温州肯恩大学



本科专业设置

温州肯恩大学
WENZHOU-KEAN UNIVERSITY

商务与公共管理学院 College of Business and Public Management		
金融学（国际金融方向）	会计学（国际会计方向）	国际商务
市场营销（国际方向）	管理科学（国际供应链和信息管理方向）	经济学

人文学院 College of Liberal Arts	
心理学	英语
传播学	

理工学院 College of Science and Technology		
计算机科学与技术	化学	环境科学
数学与应用数学（数据分析方向）	生物科学（细胞与分子方向）	

建筑与设计学院 College of Architecture and Design	
建筑学	* 视觉传达设计
* 产品设计	* 环境设计

带“*”的专业为美术类专业

第一学期无条件自由转专业政策（美术类专业、建筑学例外）

MATH Department

Dr. Gaurav Gupta
Data Analytics
Image Processing

Dr. Chun-Te lee
Applied Mathematics
Data Analytics

Dr. Sangeet Srivastava
Mathematical Modeling
Data Analytics

Dr. Adrees Ahmad
Applied Mathematics

Dr. Eduardo Rodriguez
Risk Analytics

Dr. Puneet Rana
Computational Fluid Mechanics

Dr. Imen Hassairi
Stochastic Analysis

Dr. Seyedali Ahamadian
Hosseini
Computational Fluid Mechanics

Research Focus

WKU Math Department

温州肯恩大学
WENZHOU-KEAN UNIVERSITY

- ✓ *Data Analytics*
  - ✓*Computer Vision*
  - ✓*Statistical Learning*
  - ✓*Natural Language Processing*
  - ✓*Risk analytics and strategic intelligence*
  - ✓*Computational Fluid Dynamics*

WKU Center for Excellence in Data Analytics

Introduction

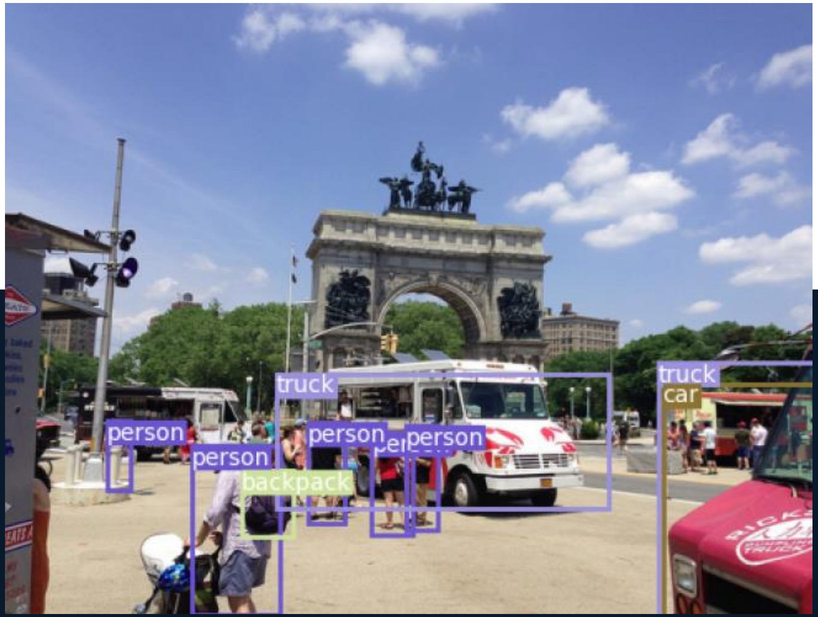
Machine Learning

- ✓ *Help us to find out meaningful predictive patterns from large amount of complex data.*
- ✓ *The benefits are outstanding, but it may not always succeed.*
- ✓ *The machine learning process is a bit tricky and challenging.*

Introduction

Image Processing with Machine Learning

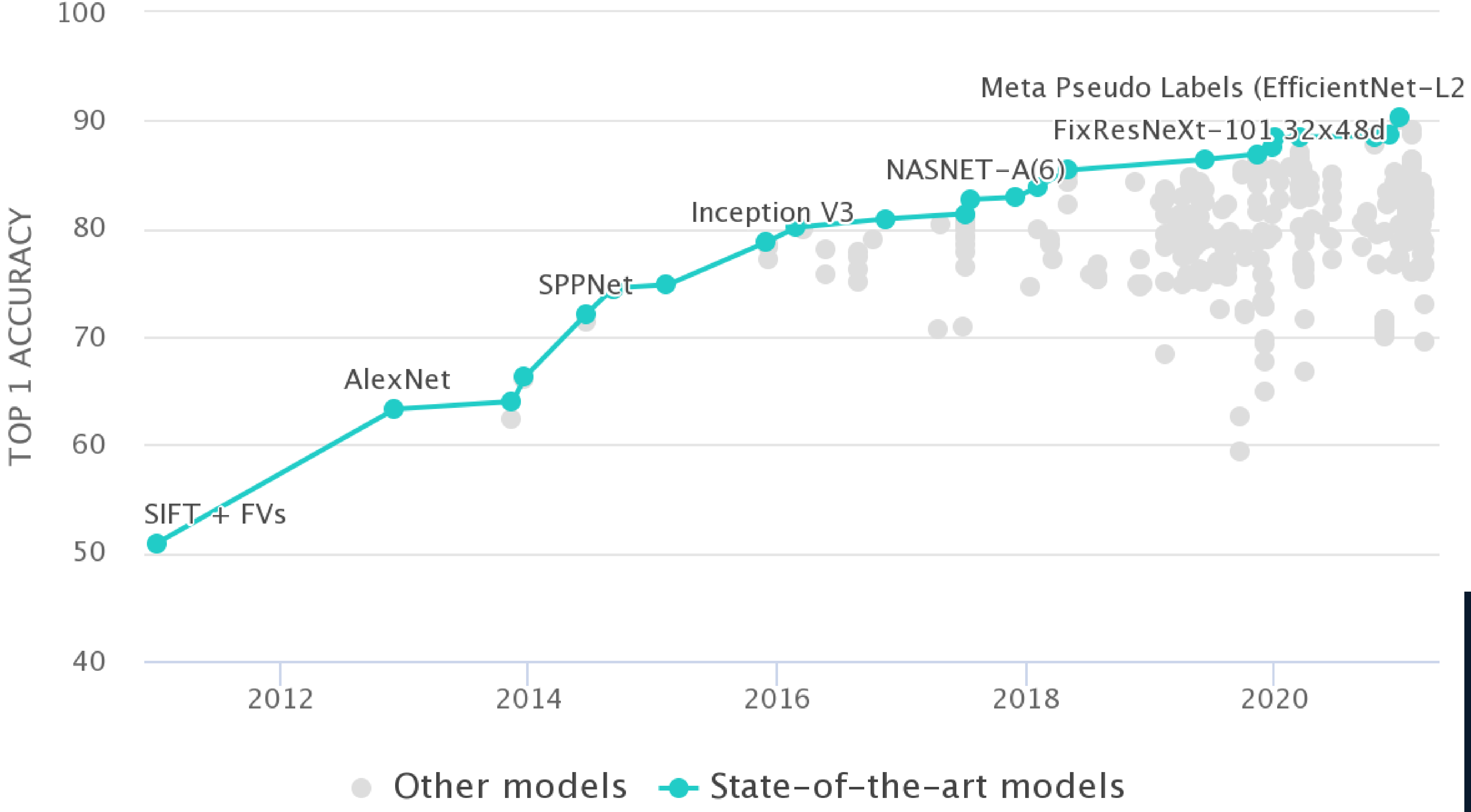
- ✓ *Dealing with large amounts of images, to find needed information*
- ✓ *Manual annotation is very costly.*
- ✓ *Machine learning can do real wonders in the domain of image processing.*



Convolutional Neural Networks

Most popular deep learning model

CNNs are everywhere!

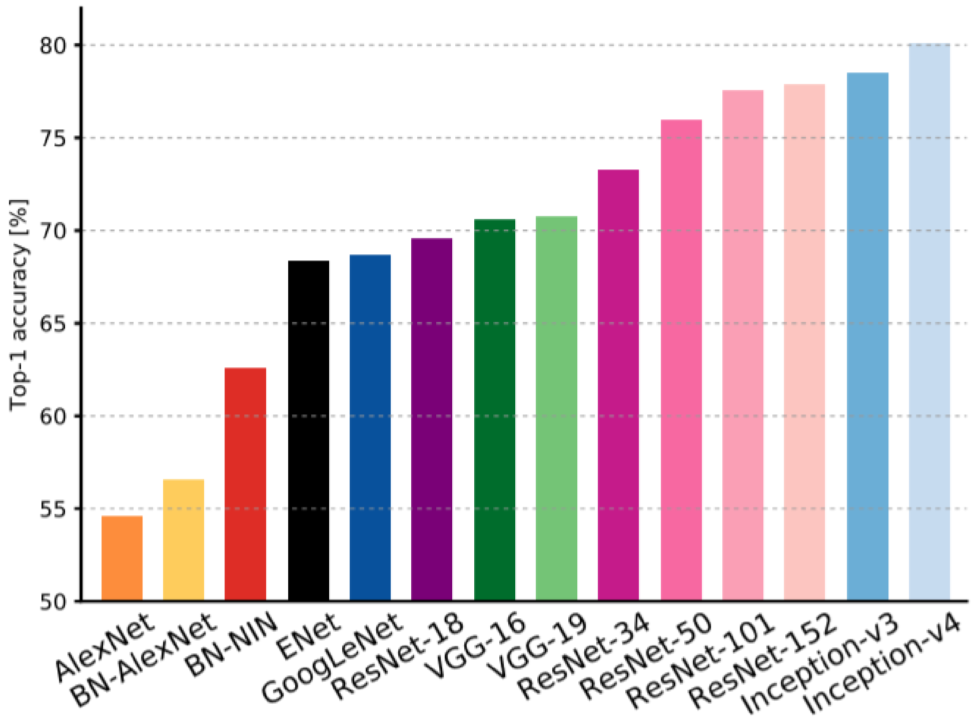


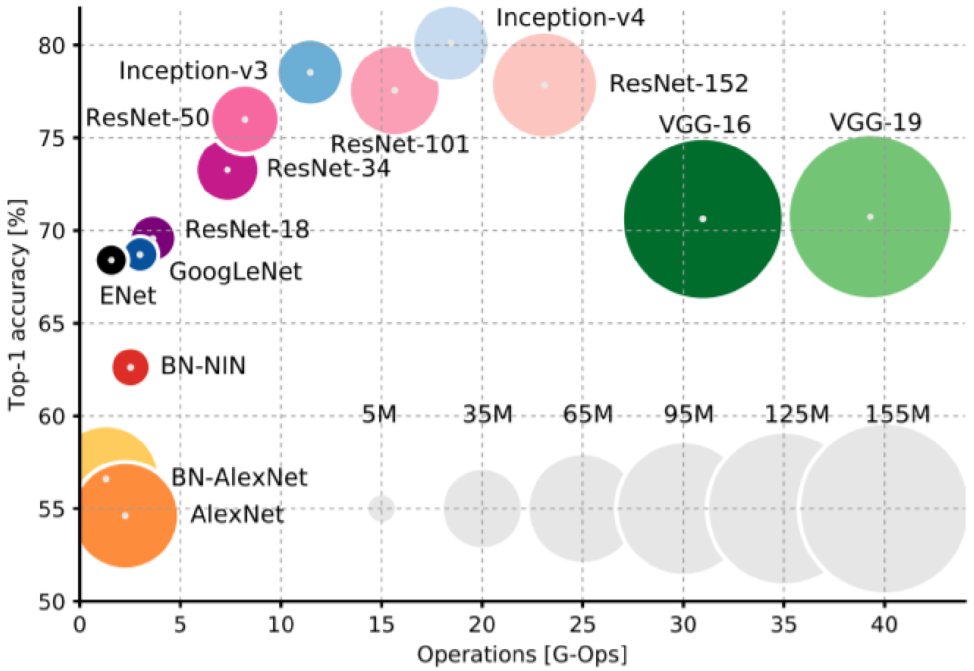
Ref: https://paperswithcode.com/sota/image-classification-on-imagenet

Problems

Most popular deep learning model

SLOW COMPUTATION!





Refernece: Alfredo Canziani, Adam Paszke, Eugenio Culurciello, 2017

Talk Focus

Focus 1: Efficient Ensemble Sparse Convolutional Neural Networks with Dynamic Batch Size

Focus 2: Deblur-YOLO: Real-Time Object Detection with Efficient Blind Motion Deblurring

Focus 3: SAPNet: Segmentation-Aware Progressive Network for Single Image Deraining

Focus 1

温州肯恩大学
WENZHOU-KEAN UNIVERSITY

Efficient Ensemble Sparse Convolutional Neural Networks with Dynamic Batch Size

- **Network Pruning & Convolutional accelerator**

-> FFT Conv. (Mathieu, 2013)

-> Winograd Conv. Operation (Winograd, 1980; Lavin, 2015)

-> Pruning & Retraining (Liu, 2016)

-> Replace Conv. with Winograd Conv. Layers (Li, 2017)

-> Move ReLU into Winograd domain (Liu, 2018)

- **Activation Functions (not in this pre.)**

Refernece: Alfredo Canziani, Adam Paszke, Eugenio Culurciello, 2017

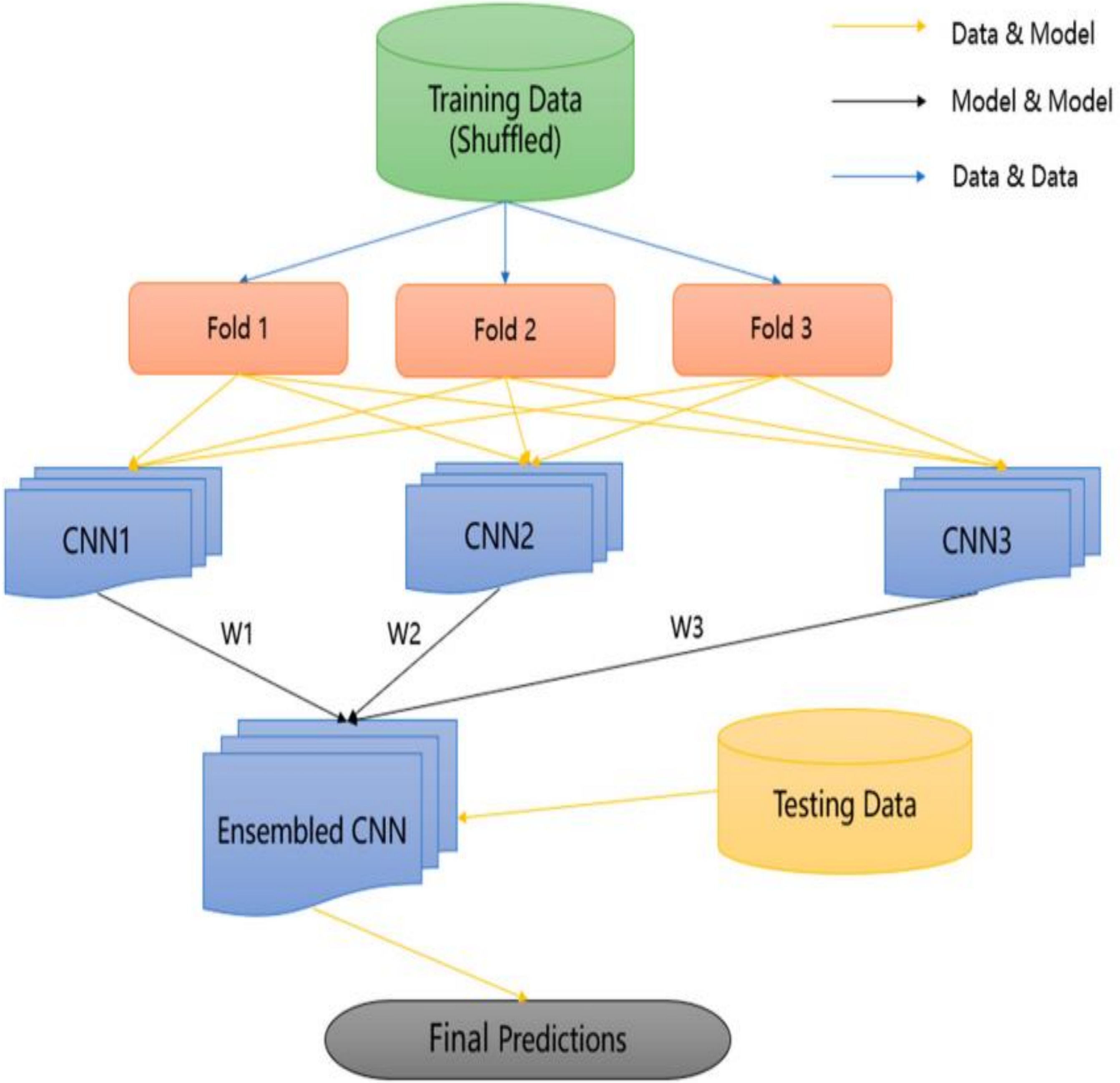
Batch Sizes

- ✓Large Batch Size: Generalization Gap ” (Krizhevsky, 2014)
- ✓Sharp Minima “ (Keskar, 2016)
- ✓Generalization Gap”: Insufficient updates (Hoffer, 2016)
- ✓Increasing the Learning Rate & Momentum (Smith, 2017)
- ✓Learning Rate Warm Up (Goyal, 2017)

Methodology

Our Solutions

Weighted Average Stacking + Network Pruning + Winograd-ReLU Convolution + Dynamic Batch size Algorithm



Methodology

Our Solutions

Algorithm 1. Mini-Batch SGDM with Dynamic Batch Size.

Require: Learning rate η, batch size B, momentum coefficient m, numbers of steps T, number of data points N, loss function $f(\theta)$.

1: **for** $t \in [1, T]$ **do**
2: $\quad B_{min} = B_0$
3: $\quad B = \text{Round\_\&\_Clip}\left(\frac{\eta(1-m_0)}{\eta_0(1-m)} B_0, B_{\min}, B_{\max}\right)$
4: $\quad B = Stepwise(B)$
5: $\quad g_t = \frac{1}{B} \sum_{i=1}^{B} \nabla f(\theta_i)$
6: $\quad v_t = m v_{t-1} + \eta g_t$
7: $\quad \theta_t = \theta_{t-1} - v_t$
8: **end for**
9: **return** B, θ_t

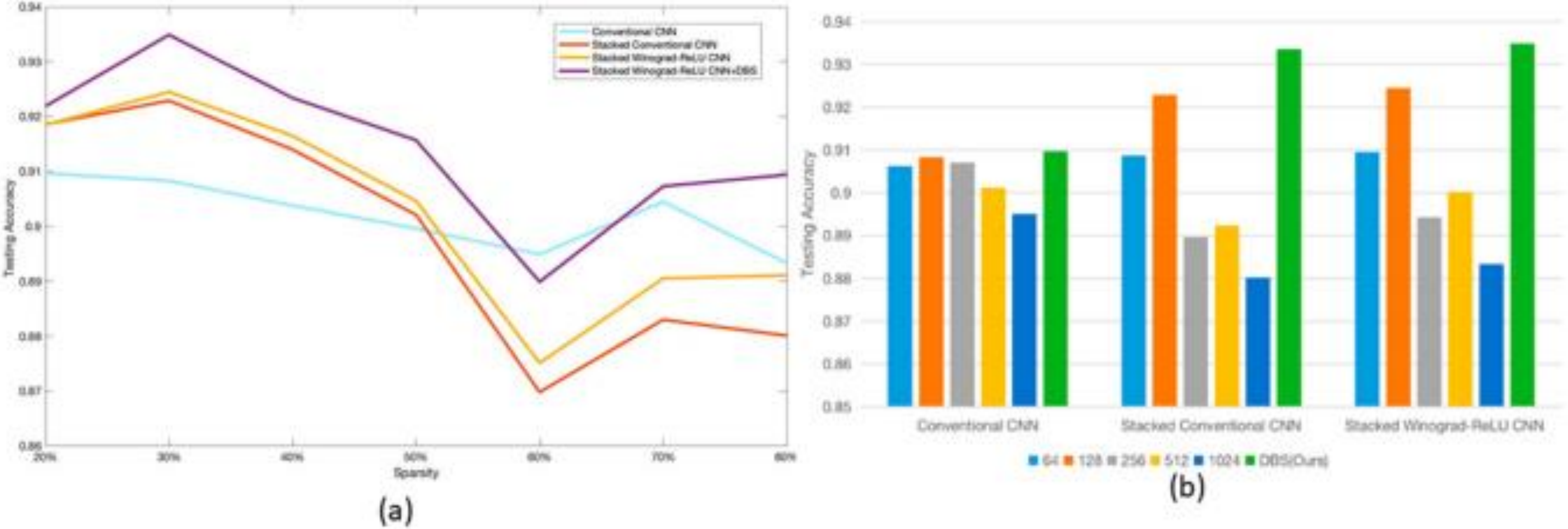
Model Development

温州肯恩大学 WENZHOU-KEAN UNIVERSITY

- Warm Up the Learning Rate gradually from 0.01 to 0.02, for the beginning 10% of the total epochs.
- Increase the learning rate by a multiplier of 2 every n epoch until validation accuracy falls, keeping momentum coefficient fixed. Linearly Scale the batch size to the learning rate.
- Increase the momentum coefficient, keeping learning rate fixed. Scale the batch size to momentum coefficient.
- Stop the above action until reaching maximum batch size, which is determined by three restrictions: GPU memory limits, non-decreasing validation accuracy and linear scaling rule constraints ($B \ll N/10$)
- If validation accuracy does not improve for five consecutive epochs, decrease the learning rate by a multiplier of 0.1.

Experiments

AlexNet (krizhevsky, 2012)+ FASHION-MNIST (Xiao, 2017)



(a) Testing accuracy for different AlexNet models on FASHION-MNIST with different batch sizes (b) Testing accuracy for different AlexNet models on FASHION-MNIST with different batch sizes

Experiments

AlexNet + FASHION-MNIST

Computational speed for different AlexNet models on FASHION-MNIST at Different sparsity

Sparsity	C-CNN		SC-CNN		SWR-CNN		SWR-CNN + DBS (ours)	
	Time	Speed	Time	Speed	Time	Speed	Time	Speed
20%	12	1.42x	37	0.46	17	1.00x	11	1.55x
30%	12	1.42x	37	0.46	17	1.00x	11	1.55x
40%	12	1.42x	36	0.47	16	1.06x	11	1.55x
50%	11	1.55x	37	0.46	17	1.00x	11	1.55x
60%	11	1.55x	37	0.46	16	1.06x	10	1.7x
70%	11	1.55x	36	0.47	16	1.06x	10	1.7x
80%	11	1.55x	37	0.46	16	1.06x	10	1.7x
Overall	11.42	1.49x	36.71	0.46x	16.43	1.03x	10.57	1.61x

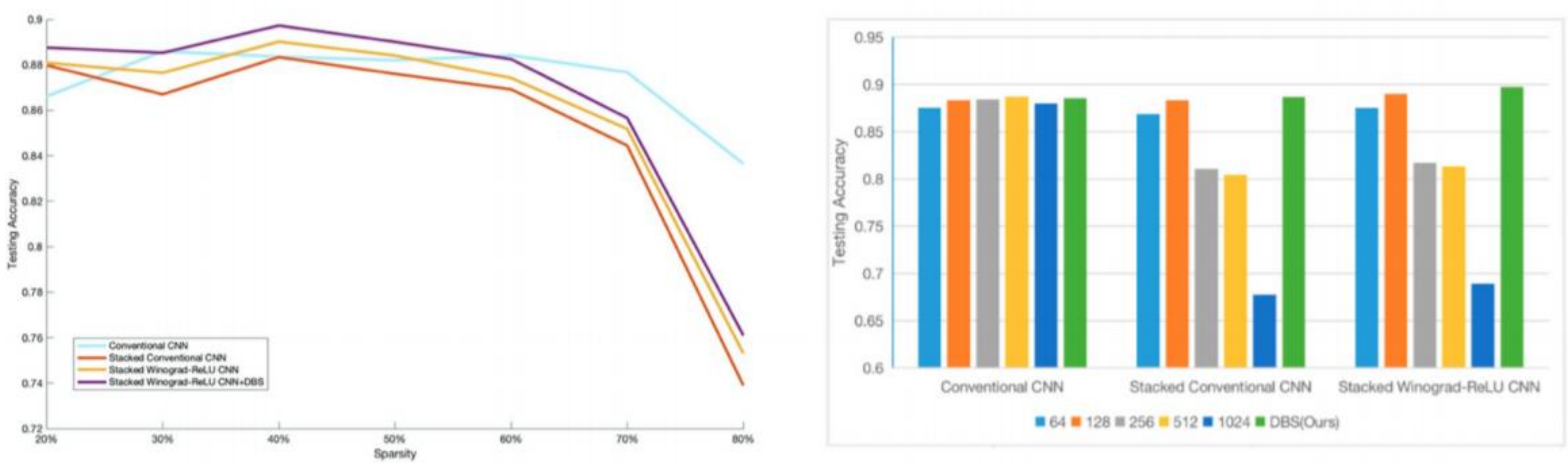
Computational speed for different AlexNet models on FASHION-MNIST with different batch sizes

Batch size	C-CNN		SC-CNN		SWR-CNN	
	Time	Speed	Time	Speed	Time	Speed
64	20	0.85x	53	0.32x	24	0.71x
128	12	1.42x	37	0.46x	17	1.00x
256	7	2.43x	28	0.61x	13	1.31x
512	5	3.4x	22	0.77x	10	1.70x
1024	3	5.67x	21	0.81x	8	2.13x
DBS(Ours)	6	2.83x	24	0.71x	11	1.55x

Experiments

温州肯恩大学
WENZHOU-KEAN UNIVERSITY

VGG (Simonyan, 2014) + CIFAR10 (Krizhevsky, 2009)



(a) (b)

(a) Testing accuracy for different VGG models on CIFAR-10 at different sparsity
(b) Testing accuracy for different VGG models on CIFAR-10 with different batch sizes

Experiments

VGG + CIFAR10

Computational speed for different VGG models on CIFAR-10 at different sparsity

Sparsity	C-CNN		SC-CNN		SWR-CNN		SWR-CNN +DBS (ours)	
	Time	Speed	Time	Speed	Time	Speed	Time	Speed
20%	19	1.37x	41	0.63x	18	1.44x	15	1.73x
30%	19	1.37x	41	0.63x	18	1.44x	14	1.86x
40%	18	1.44x	40	0.65x	18	1.44x	14	1.86x
50%	18	1.44x	40	0.65x	18	1.44x	14	1.86x
60%	18	1.44x	40	0.65x	18	1.44x	13	2.00x
70%	18	1.44x	39	0.67x	17	1.53x	13	2.00x
80%	18	1.44x	39	0.67x	17	1.53x	12	2.17x
Overall	18.3	1.42x	40.0	0.65x	17.71	1.47x	13.57	1.92x

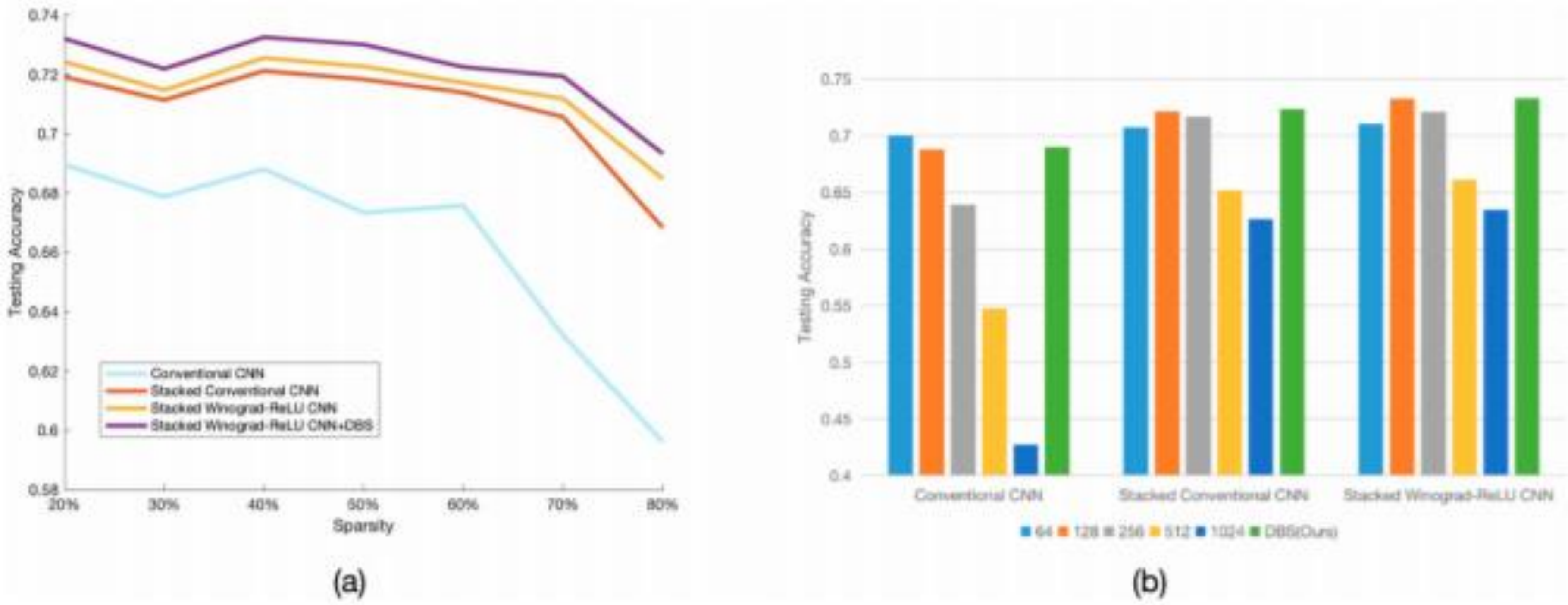
Computational speed for different VGG models on CIFAR-10 with different batch sizes

Batch size	C-CNN		SC-CNN		SWR-CNN	
	Time	Speed	Time	Speed	Time	Speed
64	28	0.93x	62	0.42x	28	0.93x
128	18	1.44x	40	0.65x	18	1.44x
256	13	2.00x	32	0.81x	15	1.73x
512	11	2.36x	28	0.93x	13	2.00x
1024	9	2.89x	27	0.96x	12	2.17x
DBS(Ours)	13	2.00x	29	0.90x	14	1.86x

Experiments

温州肯恩大学 WENZHOU-KEAN UNIVERSITY

ResNet (He, 2015) + CIFAR100 (Krizhevsky, 2009)



(a) Testing accuracy for different ResNet models on CIFAR-100 at different sparsity (b) Testing accuracy for different ResNet models on CIFAR-100 with different batch sizes

Experiments

ResNet + CIFAR100

温州肯恩大学 WENZHOU-KEAN UNIVERSITY

Sparsity	C-CNN		SC-CNN		SWR-CNN		SWR-CNN +DBS (ours)	
	Time	Speed	Time	Speed	Time	Speed	Time	Speed
20%	28	1.93x	51	1.06x	25	2.16x	21	2.57x
30%	28	1.93x	51	1.06x	25	2.16x	21	2.57x
40%	29	1.86x	49	1.10x	24	2.25x	20	2.7x
50%	28	1.93x	53	1.02x	26	2.08x	22	2.45x
60%	29	1.86x	53	1.02x	26	2.08x	22	2.45x
70%	30	1.8x	51	1.06x	25	2.16x	21	2.57x
80%	28	1.93x	49	1.1x	24	2.25x	20	2.7x
Overall	28.57	1.89x	51	1.06x	25	2.16x	21	2.57x

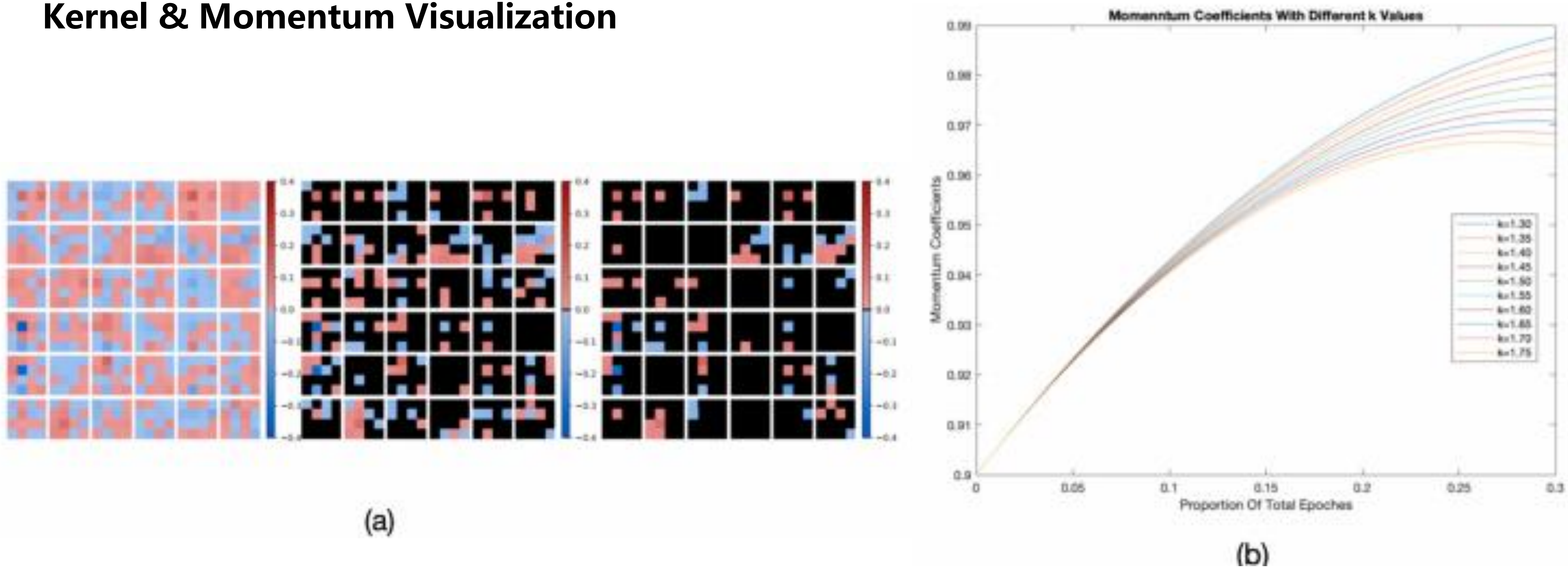
Batch size	C-CNN		SC-CNN		SWR-CNN	
	Time	Speed	Time	Speed	Time	Speed
64	49	1.10x	90	0.60x	53	1.02x
128	29	1.86x	49	1.10x	20	2.70x
256	20	2.70x	34	1.59x	14	3.86x
512	14	3.86x	25	2.16x	11	4.91x
1024	12	4.50x	22	2.45x	10	5.40x
DBS(Ours)	18	3.00x	29	1.86x	13	4.15x

Table 6. Computational speed for different ResNet models on CIFAR-100 with different batch sizes

Table 5. Computational speed for different ResNet models on CIFAR-100 at different sparsity

Experiments

Kernel & Momentum Visualization



(a) Kernels of Layer 2 from Winograd-ReLU ResNet-32 Model with dynamic batch size at different pruning sparsity (Left 0, Middle 60%, Right 80%) (b) Increase of momentum coefficient with different k values.

Findings

温州肯恩大学
WENZHOU-KEAN UNIVERSITY

Efficient Ensemble Sparse Convolutional Neural Networks with Dynamic Batch Size

- ✓ Efficient Convolutional Neural Network
- ✓ Weighted Average Stacking
- ✓ Winograd-ReLU Convolution + Pruning
- ✓ Dynamic Batch Size
  - ✓ Increase learning rate
  - ✓ Increase momentum coefficient
  - ✓ Scale the batch size

Promising Results

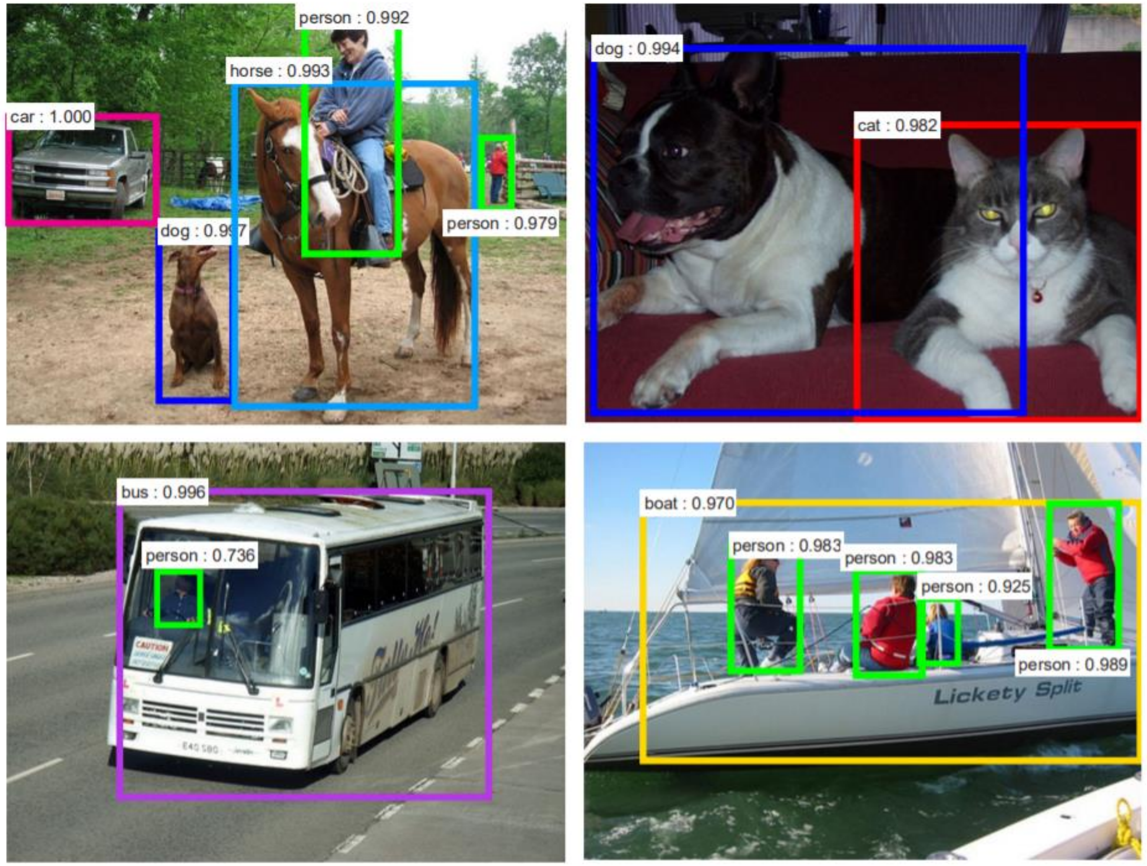
✓Fashion-MNIST	1.55x & 2.66%
✓CIFAR-10	2.86x & 1.37%
✓CIFAR-100	4.15x & 4.48%

Focus 2

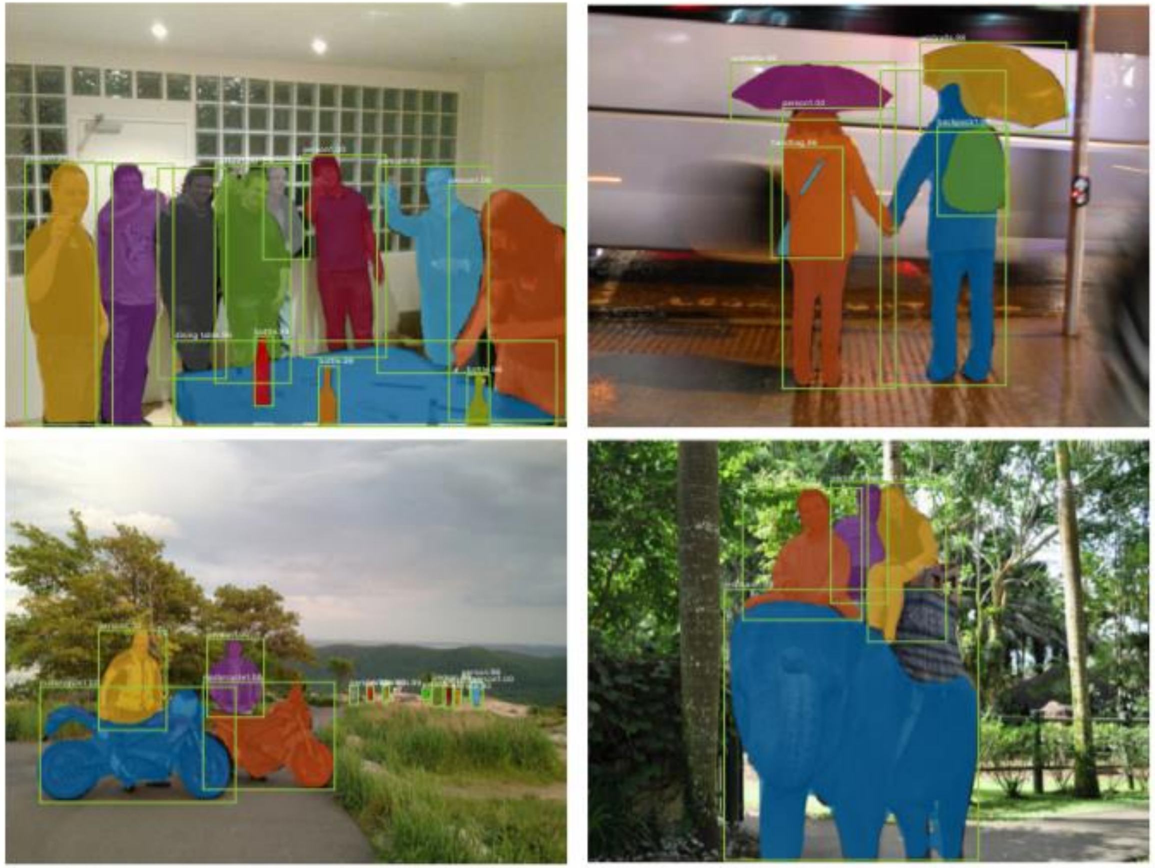
温州肯恩大学
WENZHOU-KEAN UNIVERSITY

Deblur-YOLO: Real-Time Object Detection with Efficient Blind Motion Deblurring

Object Detector are AWESOME!



Ren et.al. Faster R-CNN (2016)



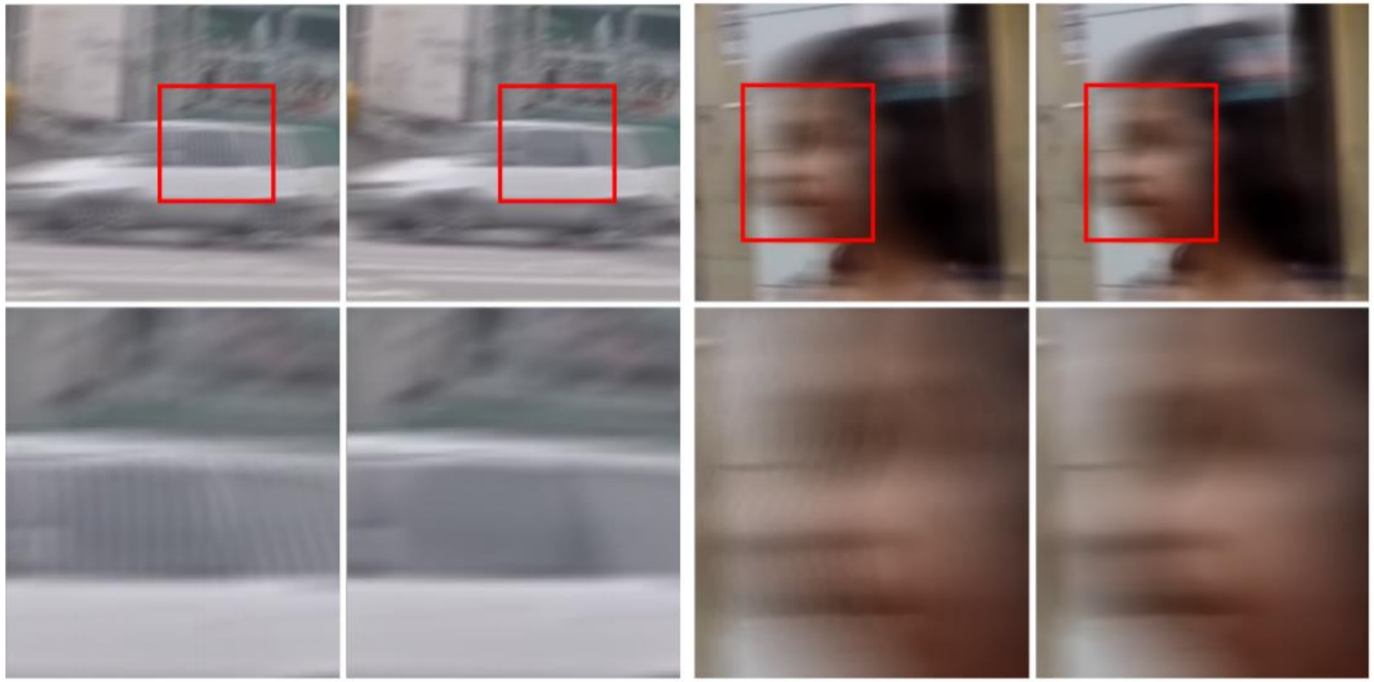
He et.al. Mask R-CNN (2018)

Problem

Real-Time Object Detection /Motion Deblurring

Real-World Situations ?

- Vehicle movement
- Camera Shake
- Poor Weather



Kupyn et.al. DeblurGANv2 (2019)

Problem...

温州肯恩大学
WENZHOU-KEAN UNIVERSITY

Real-Time Object Detection /Motion Deblurring

AWESOME"   **ONLY** at Clean Images but Suffer from Image Degradation

(a) Clean Image (b) Blurred Image (c) Deblur-YOLO

Sample Detection Result. Deblur-YOLO makes blur robust object detecton at a densely populated image from COCO 2014. Left: Yolov3 at clean image. Middle: Yolov3 at Blurred Image. Right: Deblur-YOLO at Blurred Image

Existing Solutions

- Non-Blind Deblurring
  - Unnatural l0 sparse representation (Xu, 2013)
  - Edge-based kernel estimation + Patch priors (Sun, 2013)

- Blind Deblurring
  - Non-uniform motion blur kernel estimation (Sun, 2015)
  - Fourier coefficient of deconvolutional kernel (Chakrabarti, 2016)
  - DeepDeblur (Nah, 2017), SRN-DeblurNet (Tao, 2018)
  - DeblurGANv1&v2 (Kupyn, 2018&2019)

Problems

VERY SLOW => Unsuitable for real-world tasks

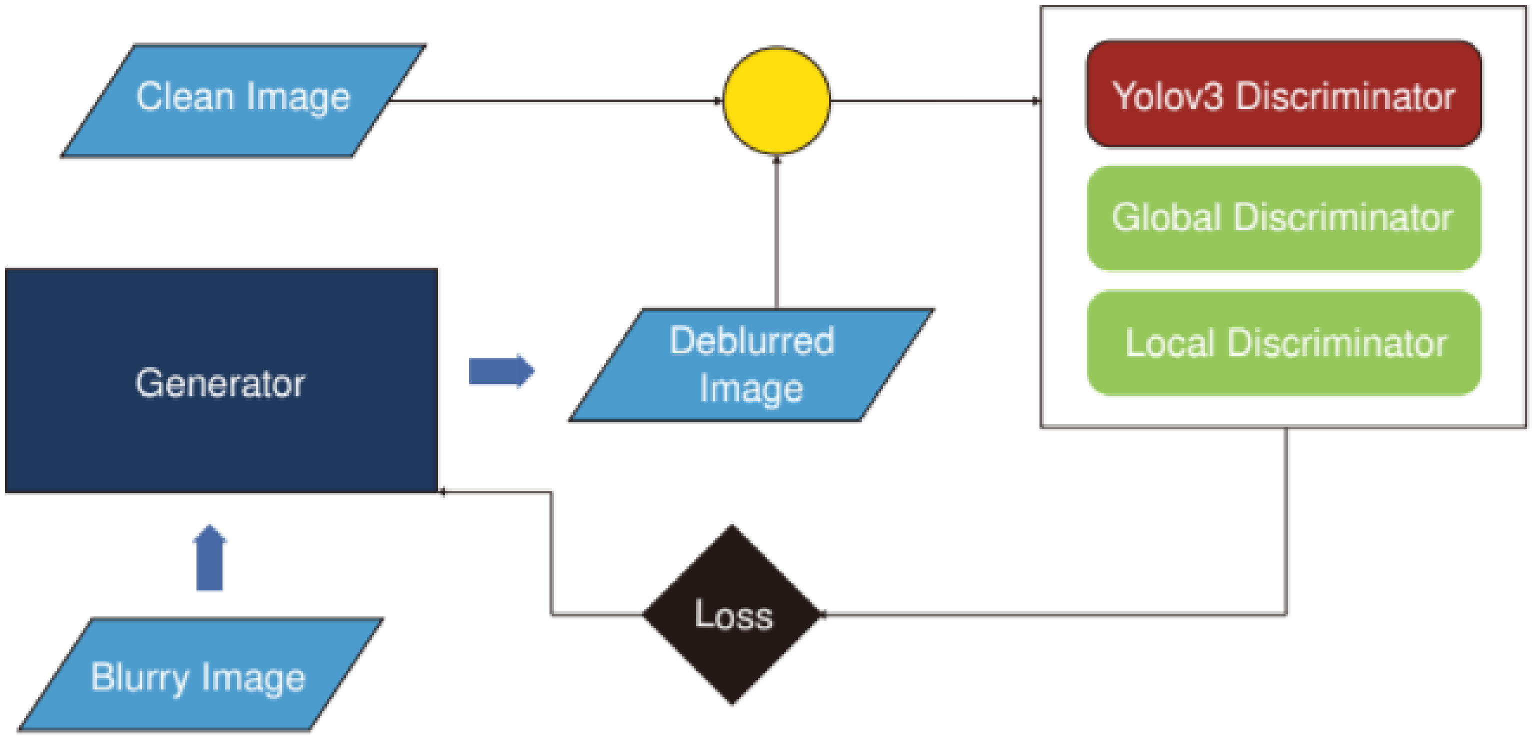
温州肯恩大学
WENZHOU-KEAN UNIVERSITY

Deblurring Performance at COCO 2014

	Time	Params	PSNR	SPSNR	SSIM
Blur Image	None	None	21.02	116.51	0.701
DeepDeblur	1.5495	47.4	24.86	105.08	0.823
DynamicDeblur	1.5247	47.8	**27.19**	113.20	**0.873**
SRN	0.3790	86.9	24.61	99.92	0.815
DeblurGANv2(I-R)	0.1589	233.0	20.29	108.45	0.687
DeblurGANv2(M)	**0.0769**	**12.8**	20.34	124.94	0.687
Deblur-Yolo	0.0772	12.9	23.94	**131.39**	0.817

Our Solution

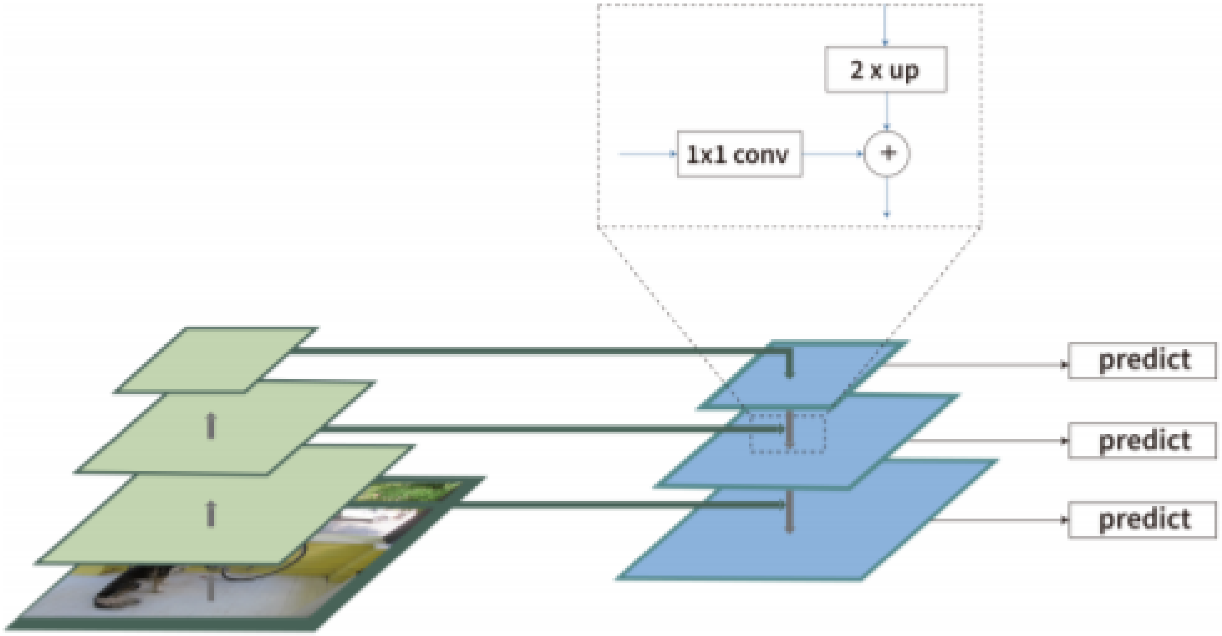
Deblur-YOLO Work Flow



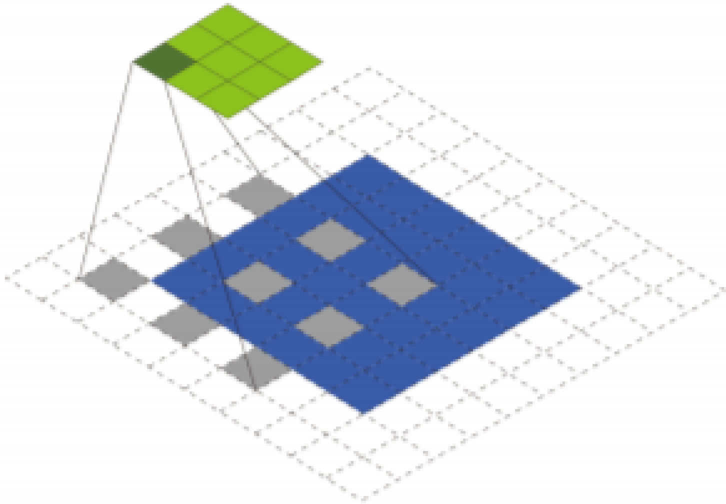
Model WorkFlow Design. Deblur-YOLO is a Generative Adversarial Network (GAN) with one generator and a group of discriminators.

Our Solution

Deblur-YOLO Generator Architecture



Generator



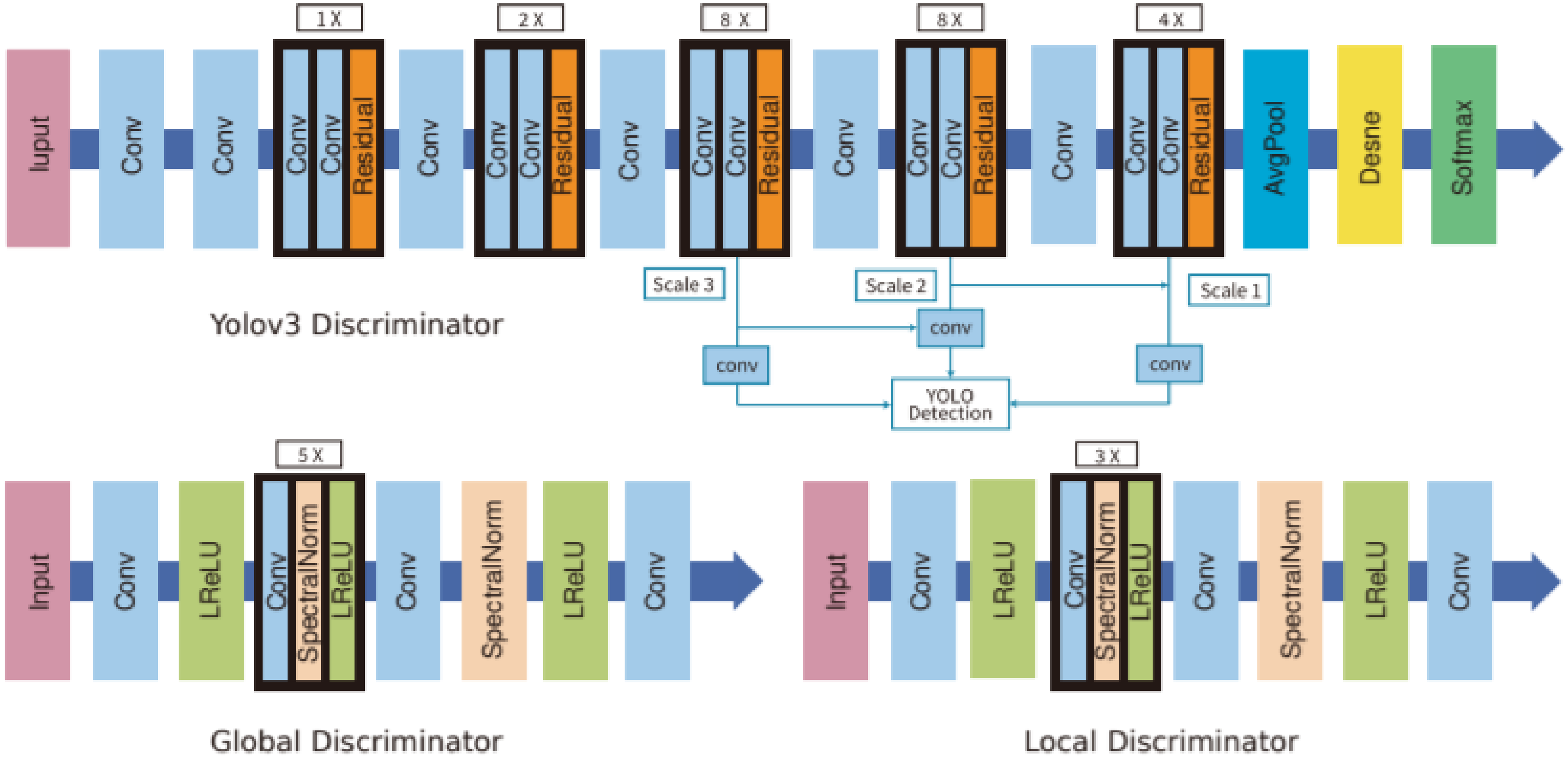
Dilated Convolution

Generator Architecture. Left: Generator building blocks with convolution and upsampling operations. Right: Dilated Convolution layers with stride 1, padding 2, dilation 2, kernel size 3 and filter number 128. We use blue, light green and dark green for dilated convolution blocks, vanilla convolution blocks and kernels, respectively. Each Convolution block consists of a convolution layer, a normalization layer and a ReLU [15] activation layer.

Our Solution

温州肯恩大学
WENZHOU-KEAN UNIVERSITY

Deblur-YOLO Discriminator Architecture



Discriminator Architecture. Up: Detection Discriminator. Lower Left: Global-Scale Discriminator. Lower Right: Local-Scale Discriminator. For Yolov3 Discriminator, we use "Conv" for convolution blocks and "Residual" for residual blocks [29]. For Global and Local Discriminator, we use "Conv" for convolution layers and "LReLU" for Leaky ReLU [30] activation layers. Each convolution layers have stride 2, padding 2 and kernel size 4.

Our Solution

Deblur-YOLO Loss Function

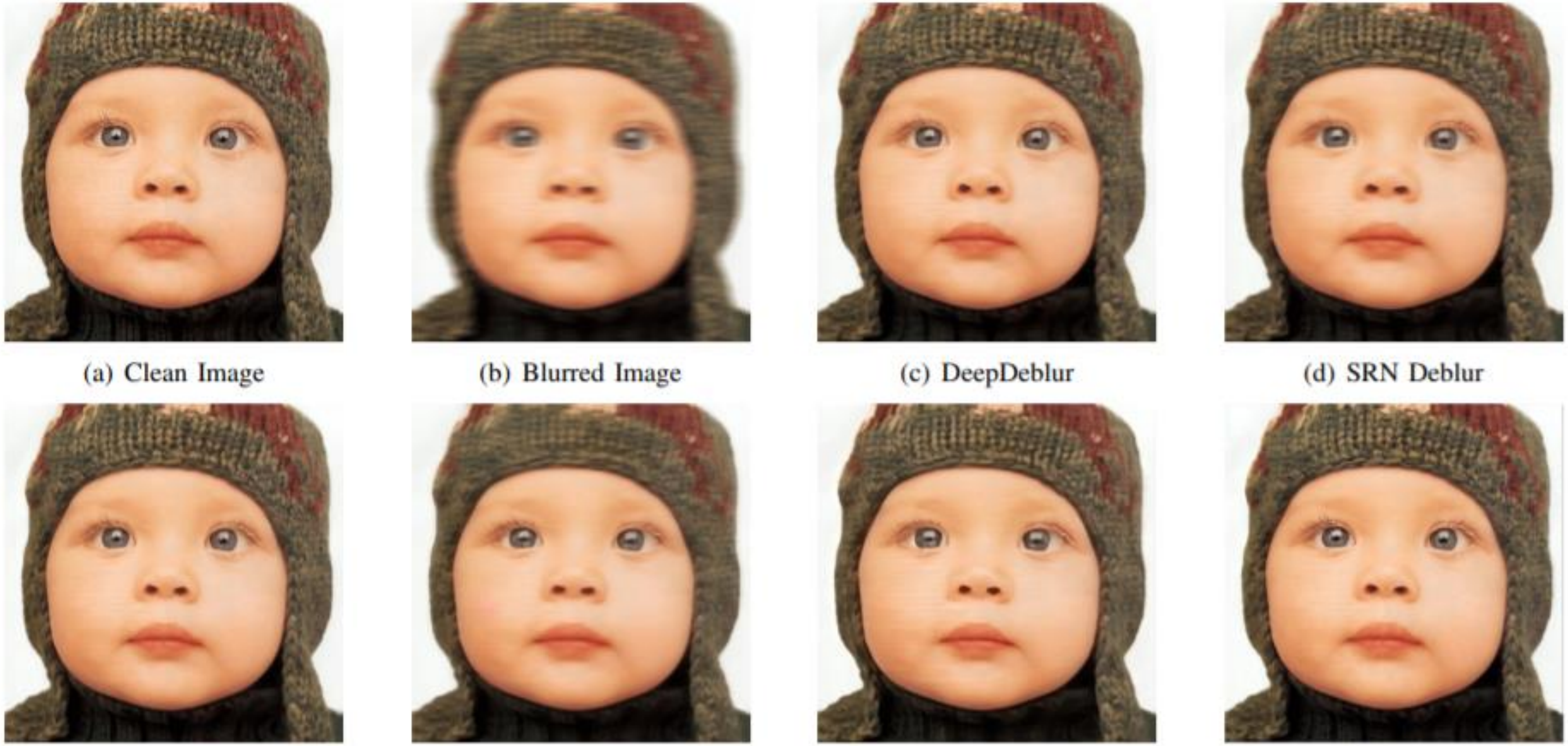
$$\mathrm{L_G} = 0.5 * \mathrm{L_C} + 0.006 * \mathrm{L_P} + 0.01 * \mathrm{L_{A_G}} + 0.1 * \mathrm{L_D}$$

$$\begin{aligned}\mathrm{L_{A_G}} = {} & \mathbb{E}_{z\sim p_z(z)}[(G(z) - \mathbb{E}_{x\sim p_{\text{data}}(x)}G(x) - 1)^2] \\ & + \mathbb{E}_{x\sim p_{\text{data}}(x)}[(G(x) - \mathbb{E}_{z\sim p_z(z)}G(z) + 1)^2]\end{aligned}$$

$$\begin{aligned}\mathrm{L_{A_D}} = {} & \mathbb{E}_{x\sim p_{\text{data}}(x)}[(D(x) - \mathbb{E}_{z\sim p_z(z)}D(G(z)) - 1)^2] \\ & + \mathbb{E}_{z\sim p_z(z)}[(D(G(z)) - \mathbb{E}_{x\sim p_{\text{data}}(x)}D(x) + 1)^2]\end{aligned}$$

Experiments

Qualitative Result at Set5



(a) Clean Image (b) Blurred Image (c) DeepDeblur (d) SRN Deblur

(e) DynamicDeblur (f) DeblurGANv2(I-R) (g) DeblurGANv2(M) (h) Deblur-YOLO

: Deblurring Result Comparison at Baby Picture from Set 5

Experiments

Qualitative Result at Set14

(a) Clean Image (b) Blurred Image (c) DeepDeblur (d) SRN Deblur

(e) DynamicDeblur (f) DeblurGANv2(I-R) (g) DeblurGANv2(M) (h) Deblur-YOLO

Deblurring Result Comparison at Lenna Picture from Set14

Experiments

Qualitative Result at COCO 2014

(a) clean Image (b) Blurred Image (c) DeepDeblur
(d) SRN Deblur (e) DynamicDeblur (f) DeblurGANv2(I-R)
(g) DeblurGANv2(M) (h) Deblur-YOLO

Deblurring and Detection Result Comparison at COCO 2014

Experiments

Quantitative Results

mAP Score at COCO 2014

	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Clean Image	58.5	73.7	46.9	44.4	40.6	42.3	75.2	58.5	73.0	39.8	52.3	41.4	73.2	77.5	61.7	69.1	42.4	59.9	52.1	75.4	69.9
Blur Image	29.7	43.3	16.9	15.3	14.8	10.4	51.8	34.5	40.4	13.5	12.6	26.9	31.7	30.9	28.2	42.8	19.0	23.7	33.7	57.6	45.4
DeepDeblur	51.7	64.9	36.9	35.2	35.3	32.2	73.1	53.7	70.3	33.9	40.8	40.1	59.6	68.1	51.9	65.3	35.2	46.8	48.5	72.8	68.4
DynamicDeblur	**56.0**	**70.6**	**43.2**	**41.4**	**41.4**	**36.8**	**75.3**	**57.1**	**72.6**	**36.6**	**45.8**	40.0	**68.2**	**72.7**	**56.3**	**67.0**	**39.8**	**58.4**	49.9	**75.3**	**71.7**
SRN	52.3	70.1	38.2	35.8	35.8	31.8	71.9	53.4	69.2	33.1	39.4	39.8	63.4	66.6	53.5	64.1	35.1	51.7	48.0	**75.3**	69.0
DeblurGANv2(I-R)	42.0	55.0	28.6	26.6	30.2	24.9	61.4	44.9	53.5	27.5	35.4	32.4	47.4	53.7	39.6	51.8	24.8	41.2	39.2	65.2	55.9
DeblurGANv2(M)	40.8	52.2	27.4	25.0	28.9	24.3	61.0	44.3	53.7	25.9	31.7	30.5	45.2	49.4	39.2	50.8	25.0	38.6	40.6	66.0	56.8
Deblur-Yolo	47.5	55.5	33.8	30.0	37.7	29.7	67.7	51.1	62.6	31.2	39.5	**41.2**	51.4	54.7	44.9	56.1	33.6	53.9	**50.2**	72.8	52.2

Deblurring Performance at Set 5 & Set 14

		Blur Image	DeepDeblur	DynamicDeblur	SRN	DeblurGANv2(I-R)	DeblurGANv2(M)	Deblur-Yolo
Set 5	PSNR	24.20	28.36	29.10	28.07	26.64	27.06	**29.39**
	SPSNR	113.79	104.80	113.99	98.77	103.74	122.66	**128.40**
	SSIM	0.66	0.81	0.85	0.80	0.74	0.77	**0.88**
Set 14	PSNR	23.12	26.65	27.35	25.90	25.95	25.03	**27.85**
	SPSNR	119.26	111.00	115.09	111.58	116.26	**128.30**	121.70
	SSIM	0.55	0.69	0.73	0.67	0.68	0.65	**0.75**

Findings

Deblur-YOLO

✓Efficient, Detection-Driven, One-Stage

✓Generator + Multi-Scale Discriminator + Detection Discriminator

✓Blind motion deblurring + Object Detection

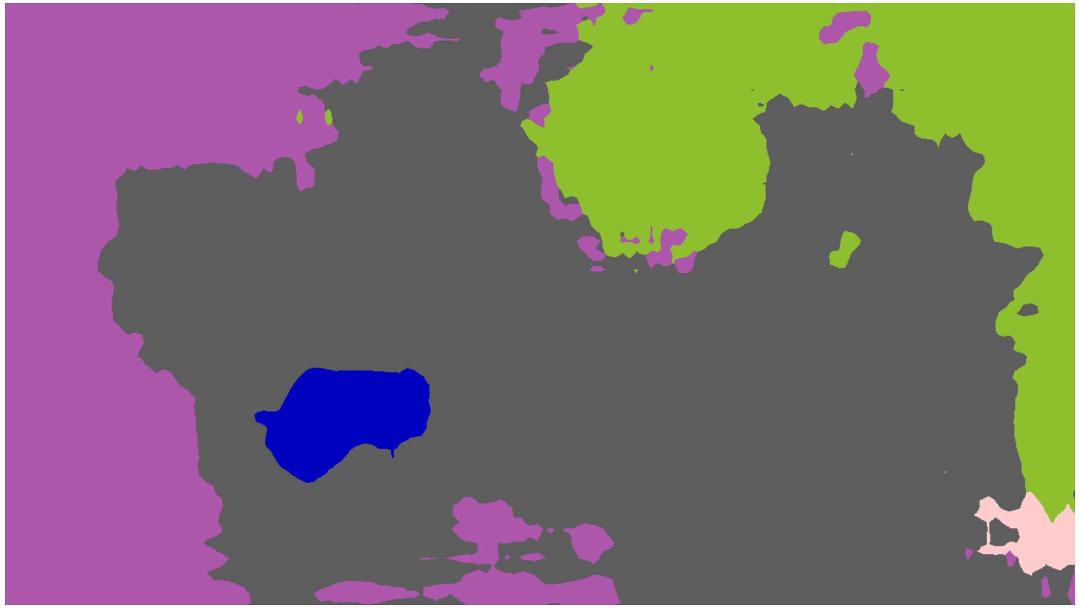
✓Smooth Peak Signal-to-Noise Ratio (SPSNR)

✓Promising Results on COCO2014, Set5 and Set14

Focus 3

SAPNet: Segmentation-Aware Progressive Network for Single Image Deraining

- Rain: Severely Degrade High Level Vision Tasks



Li et.al. (2019)

Existing Solution

温州肯恩大学
WENZHOU-KEAN UNIVERSITY

SAPNet: Segmentation-Aware Progressive Network for Single Image Deraining

- Traditional Deraining Methods (Before 2017)
  - Image Decomposition (Kang, 2011)
  - Sparse Representation (Luo, 2015)
  - Gaussian Mixture Model (Li, 2016)

- Deep Deraining Learning Methods (After 2017)
  - Convolutional Neural Network (Fu,2017; Yang,2017; Li,2018, Ren,2019; Jiang,2020)
  - Generative Adversarial Network (Qian,2018; Li,2019)

Questions

- Is image restoration always beneficial for high-level vision tasks?

  Probably **NO**! (Haris, 2018; Pei, 2018; Li, 2019)

- Low-Level + High-Level separately?

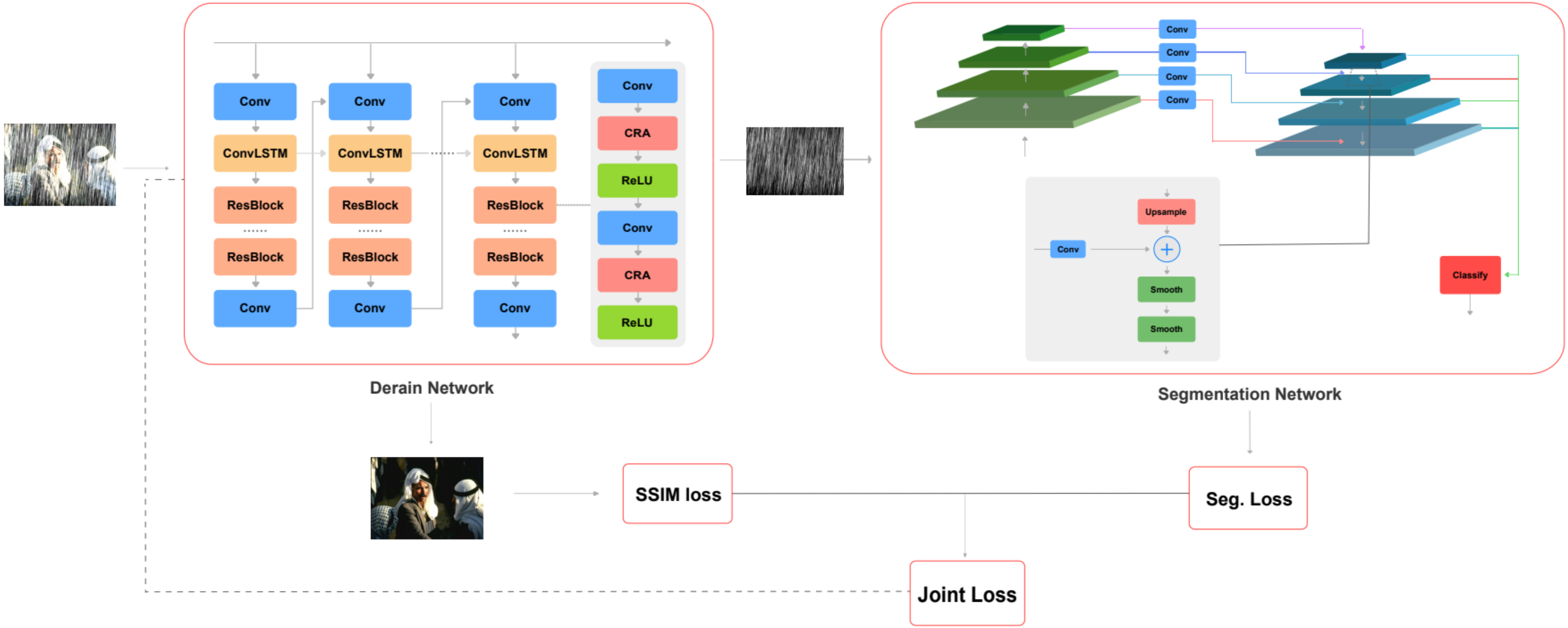
  Restoration Outcome -> **NOT** directly helpful

- Joint Training? (Liu, 2017; Fan, 2018; Haris, 2018)

  Expensive Annotations

Our Solution

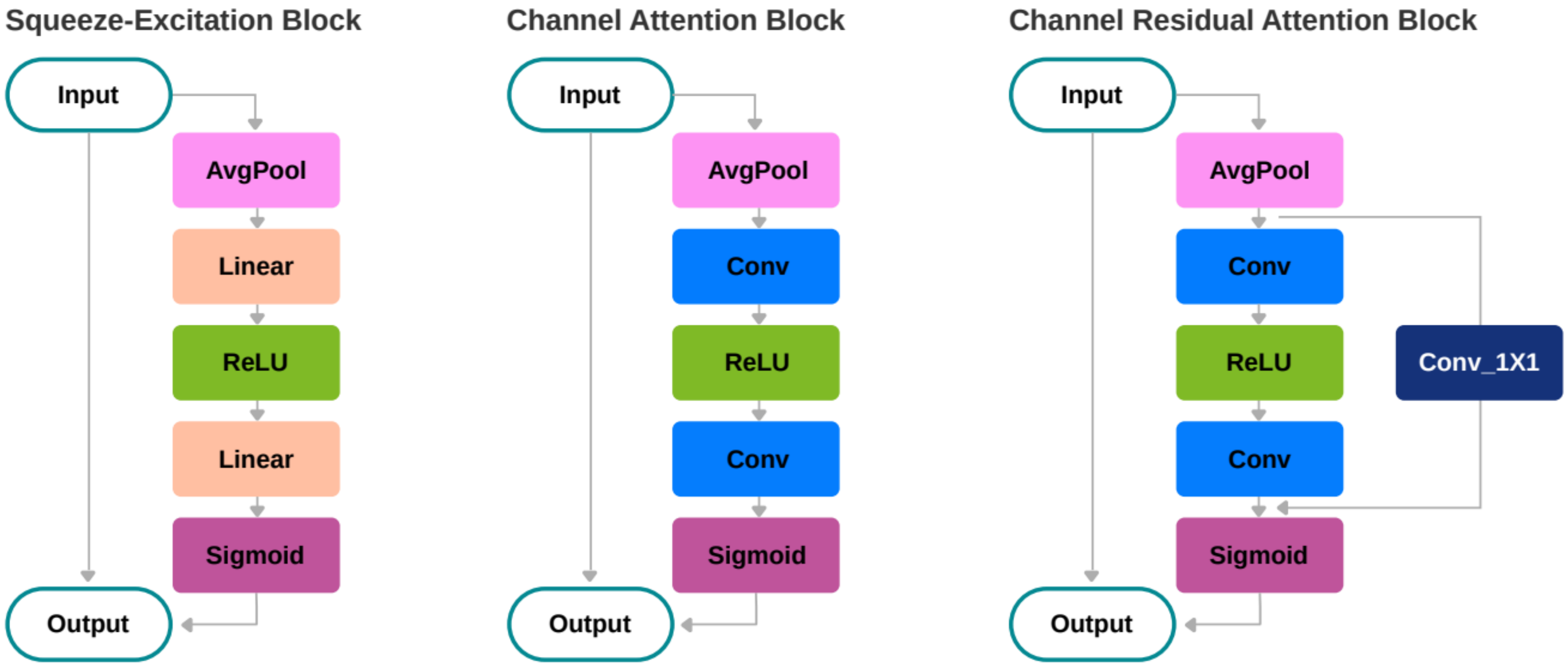
Model Architecture



Model architecture for SAPNet. SAPNet joins a derain network for supervised rain removal and a segmentation network for unsupervised rain streak segmentation. The derain network's negative SSIM loss merges with the segmentation network's segmentation loss. We use "Conv" for convolution blocks, "ConvLSTM" for convolutional LSTM, "ResBlock" for residual blocks and "CRA" for channel residual attention blocks in Fig. 3.

Our Solution

Channel Residual Attention Block



Different Building Blocks for image deraining. Left: Squeeze-Excitation [15] Block. Middle: Channel Attention [46] Block. Right: Channel Residual Attention Block (ours). We use "AvgPool" for adaptive average pooling and "Conv 1×1" for 1×1 convolution.

Our Solution

Loss Function

$$\mathcal{L}_{\text{content}} = -\operatorname{SSIM}\left(\mathbf{x}^D, \mathbf{x}^G\right)$$

$$\mathcal{L}_{\text{seg}} = \frac{1}{XY} \sum_{1 \le i \le X, 1 \le j \le Y} -\alpha \left(1 - p_{i,j}\right)^{\gamma} \log p_{i,j}$$

$$\mathcal{L}_{\text{edge}} = \sqrt{\left(\operatorname{Canny}\left(\mathbf{x}^D\right) - \operatorname{Canny}\left(\mathbf{x}^G\right)\right)^2 + \varepsilon^2}$$

$$\mathcal{L} = \lambda_1 \times \mathcal{L}_{content} + \lambda_2 \times \mathcal{L}_{seg} + \lambda_3 \times \mathcal{L}_{edge}$$

Experiments

Ablation Study

	Model1	Model2	Model3	Model4
URSS		✓	✓	✓
Canny			✓	✓
CRA				✓
PSNR	28.05	28.11	28.16	29.03
SSIM	0.883	0.886	0.884	0.894
Inf. Time	0.129	0.131	0.130	0.150

Table 2. Ablation Study on Different Components

	SAPNet-SE	SAPNet-CA	SAPNet-CRA
PSNR	28.74	28.89	29.03
SSIM	0.890	0.892	0.894
# Params	170243	170243	180483
Inf. Time	0.166	0.149	0.150

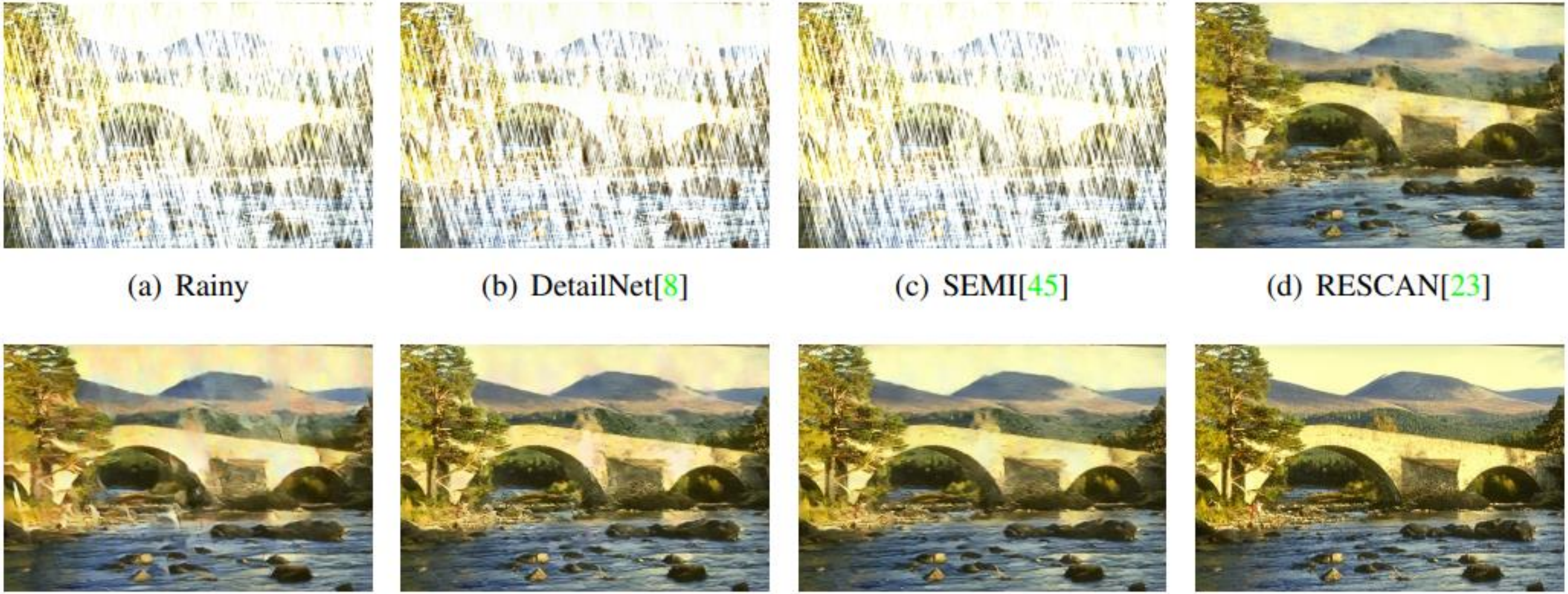
Table 3. Ablation Study on Different Building Blocks.

(a) Rainy (b) SAPNet-Conv (c) SAPNet-SE (d) SAPNet-CA (e) SAPNet-CRA

. Visual Comparison for Ablation Study

Experiments

Synthetic Rain Images (Qualitative)



(a) Rainy (b) DetailNet[8] (c) SEMI[45] (d) RESCAN[23]

(e) MSPFN[16] (f) PreNet[39] (g) SAPNet (Ours) (h) GroundTruth

Visual Comparison on Synthetic Images (Rain100H [50])

Experiments

Synthetic Rain Images (Quantitative)

	DetailNet[8]	RESCAN[23]	PreNet[39]	MSPFN[16]	SAPNet (ours)
# Params	57,369	149,823	168,963	15,823,424	180,483
Inf. Time (481 × 321)	0.161	0.498	0.128	0.327	0.150
Inf. Time (512 × 512)	0.528	0.865	0.177	0.555	0.203

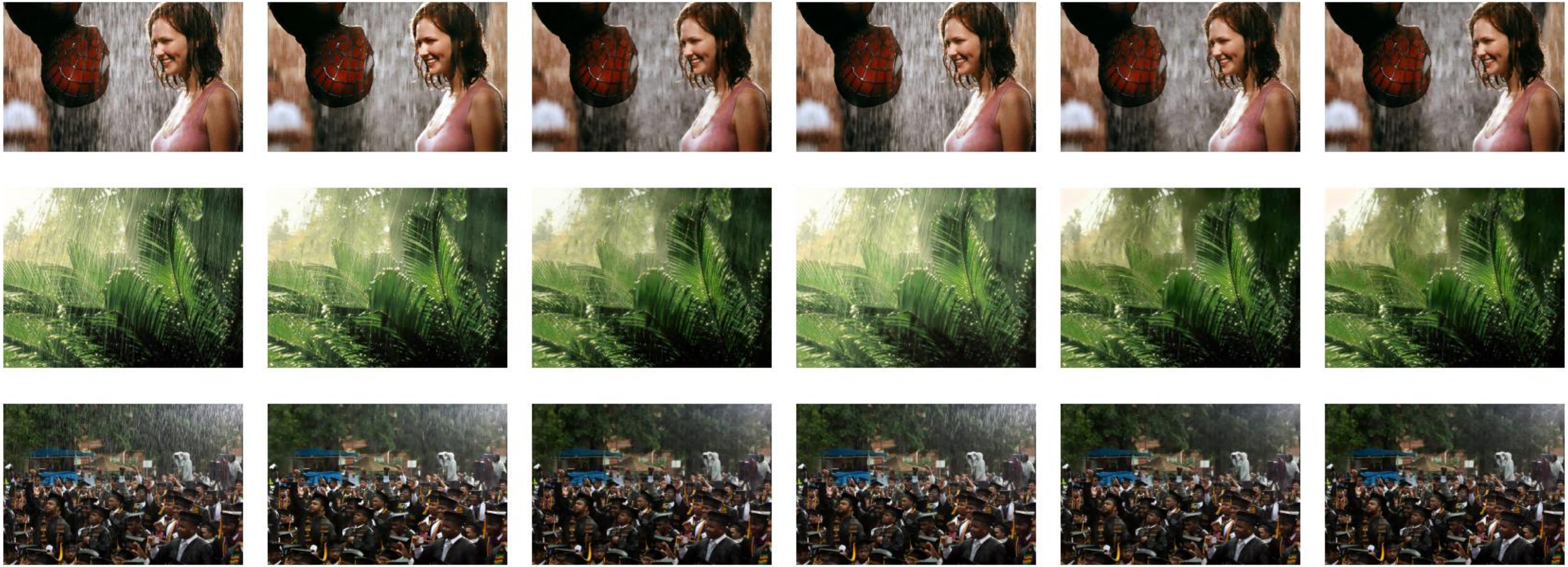
Table 4. Model Efficiency Comparison

	Rain12[24]	Rain100L[47]	Rain100H[47]	Rain1200[48]
Rainy Image	28.82/0.836	25.52/0.825	12.13/0.349	22.15/0.690
DetailNet[8]	28.89/0.897	26.25/0.856	12.65/0.420	**22.34/0.781**
SEMI[45]	24.14/0.775	25.03/0.842	16.56/0.486	22.39/0.740
MSPFN[16]	34.17/0.945	30.55/0.915	26.29/0.798	23.71/0.755
RESCAN[23]	33.60/0.953	31.76/0.946	27.43/0.841	22.58/0.754
PreNet[39]	34.79/0.964	36.09/0.976	28.06/0.884	22.28/0.741
SAPNet (ours)	**35.26/0.966**	**34.59/0.977**	**29.03/0.894**	23.06/0.756

Quantitative Comparison (PSNR/SSIM) on synthetic Images. Higher value indicate better visual quality. We use **Bold** for the highest SSIM score and blue for second highest SSIM score

Experiments

Real Rain Images (Qualitative)



(m) Rainy (n) DetailNet[8] (o) RESCAN[23] (p) MSPFN[16] (q) PreNet[39] (r) SAPNet (ours)

Visual Comparison on Real World Images

Experiments

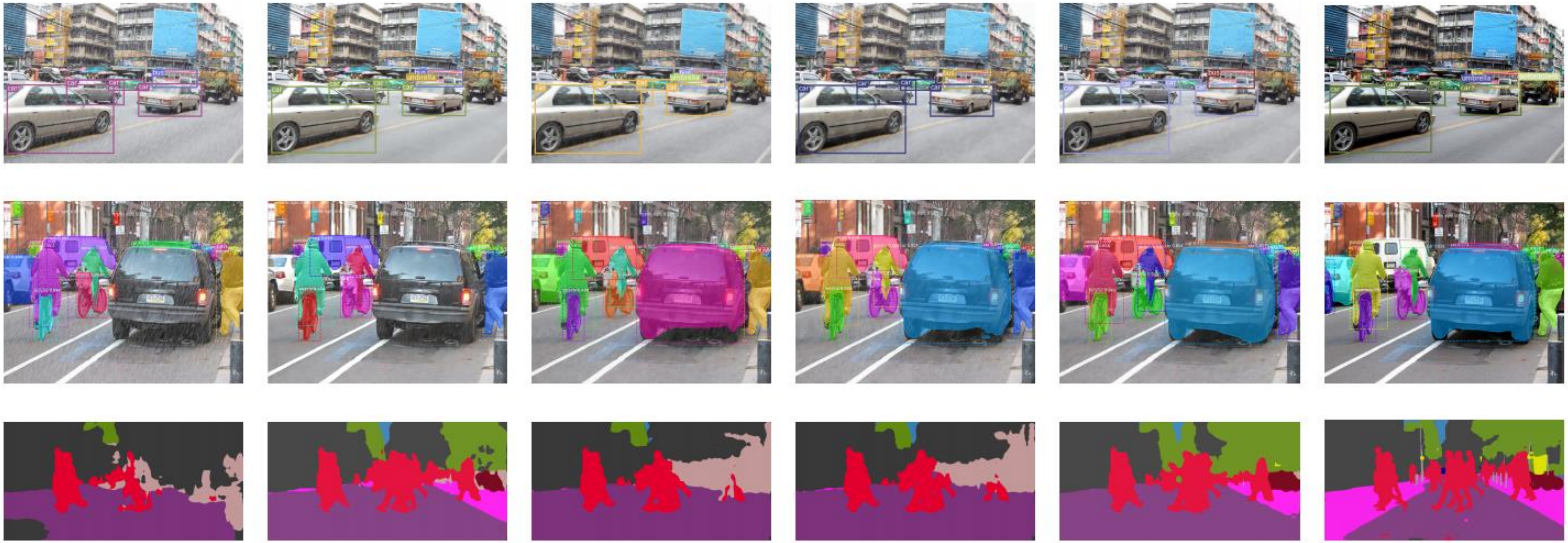
Real Rain Images (Quantitative)

	Image1	Image2	Image3
DetailNet[8]	19.95/11.29	16.60/30.52	15.86/27.59
SEMI[45]	24.41/6.820	14.26/21.53	**16.59/23.49**
RESCAN[23]	11.12 /6.905	12.40 /30.80	15.21 /26.78
MSPFN[16]	12.99 /22.77	13.58 /19.53	16.85 /29.98
PreNet[39]	13.78 /7.287	13.25 /12.13	15.56 /23.89
SAPNet (ours)	**12.98 /6.593**	**13.01 /11.48**	15.51 /26.27

Quantitative Comparison (NIQE/BRISQUE) on Real World Images. Smaller value indicate better visual quality. We use **Bold** for the smallest BRISQUE score and blue for second smallest BRISQUE score.

Experiments

Detection & Segmentation (Qualitative)



(m) Rainy (n) DetailNet[8] (o) RESCAN[23] (p) PreNet[39] (q) SAPNet (ours) (r) GroundTruth

. Visual Comparison on Object Detection (top row), Instance Segmentation (middle row) and semantic segmentation (bottom row)

Findings

温州肯恩大学
WENZHOU-KEAN UNIVERSITY

SAPNet: Segmentation-Aware Progressive Network for Single Image Deraining

✓ Segmentation-Aware Progressive Image Deraining Network

✓ Channel Residual Attention Block

✓ Unsupervised Rain Streak Segmentation

✓ Canny Edge Loss

✓ State-of-the-art on Rain100L , Rain100H, Rain12

✓ Beneficial for Detection and Segmentation

Thanks