INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

Efficient Semantic Segmentation using Gradual Grouping

Nikitha Vallurupalli[1], SriHarsha Annamaneni[1], Girish Varma[1], C V Jawahar[1], Manu Mathew[2], Soyeb Nagori[2]
IIIT Hyderabad[1], Texas Instruments Bangalore[2]

2018 CVPR
SALT LAKE CITY • JUNE 18-22

REAL TIME SEMANTIC SEGMENTATION

Consume energy efficiently (Portability)

Operation	Energy [pJ]	Relative Cost
32 bit int ADD	0.1	1
32 bit float ADD	0.9	9
32 bit Register File	1	10
32 bit int MULT	3.1	31
32 bit float MULT	3.7	37
32 bit SRAM Cache	5	50
32 bit DRAM Memory	**640**	**6400**

Give real-time output(30fps) in constrained memory and High accuracy for safety

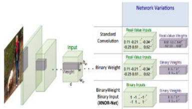
image/annotation from cityscapes dataset

Cloud not an option

Large latencies for real-time output
Violates user privacy.
Consumes network bandwidth.

MODEL COMPRESSION
Previous Approaches

Quantization

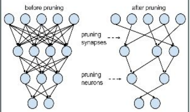


$$\hat{W}_{ij} = \begin{cases} 1 & \text{if } W_{ij} \geq 0, \\ -1 & \text{if } W_{ij} < 0. \end{cases}$$

High precision arithmetic not essential for obtaining high performance.

This results in memory savings and faster computation.

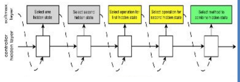
Pruning



DNNs have redundant parameters which can be removed without loss in performance.

Techniques deal with what to prune, how to prune, when to prune, etc.

Architecture Design



Learning Transferable Architectures for Scalable Image Recognition
Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V. Le

Hand engineered Architecture Design:
Uses heuristics and intuition.

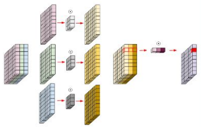
Automated Architecture Learning: Uses neural networks to design neural networks.

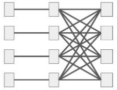
Sparse, Quantized, Full Frame CNN for Low Power Embedded Devices, Oral Presentation at **CVPRW, 2017** Manu Mathew, Kumar Desappan, Pramod Kumar, Soyeb Nagori

XNOR-Net: Imagenet Classification Using Binary Convolutional Neural Networks
Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi ECCV 2016

RECENT TECHNIQUES TO MAKE CNN'S EFFICIENT

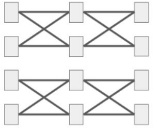
Depthwise separable convolutions





3x3 depthwise followed by 1x1 pointwise convolutions

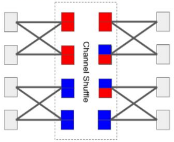
Grouped Convolutions



Grouped convolutions can be thought of as a dense convolution with certain weights zeroed out.

Simple way of having structured sparsity in convolutions.

Shuffled Convolutions

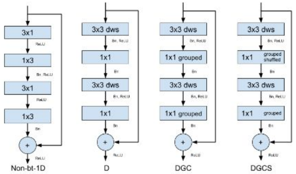


Channel shuffling operation enables cross-group information flow for multiple group convolutions.

layer with g groups whose output has $g \times n$ channels.

- reshape the output channel dimension into (g, n)
- transpose output
- flatten output back

MICRO LEVEL ARCHITECTURE MODULES



Non-bt-1D is the non-bottleneck layer used in ERFNet, D, DGC and and DGCS are our proposed micro level layer architectures.

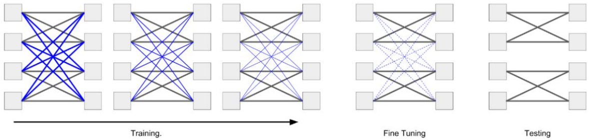
We have experimented with these proposed micro level layers on the ERF Net baseline macro architecture and studied the effect of changing each module in the encoder. Decoder is not yet optimized.

When micro architectural change is applied to every layer in the encoder, we observed that accuracy degradation is over **10%** while the model is compressed by **4x.**

Selective application of micro level CNN modules is done by leaving few initial layers and applying micro architectural changes only to the later layers in the encoder. In this case, we observe that the accuracy change is negligible but the models are not highly compressed.

	Layer	Type	out-chann	out-Res
ENCODER	1	**Downsampler block**	16	512x256
	2	**Downsampler block**	64	256x128
	3-5	**3 x Non-bt-1D**	128	128x64
	5-7	**2 x Conv-module**	64	256x128
	8	**Downsampler block**	128	128x64
	9	**Non-bt-1D**(dilated 2)	128	128x64
	10	**Non-bt-1D**(dilated 4)	128	128x64
	11	**Non-bt-1D**(dilated 8)	128	128x64
	12	**Non-bt-1D**(dilated 16)	128	128x64
	13	**Conv-module**(dilated 2)	128	128x64
	14	**Conv-module**(dilated 4)	128	128x64
	15	**Conv-module**(dilated 8)	128	128x64
	16	**Conv-module**(dilated 16)	128	128x64
DECODER	17	**Deconvolution**(upsampling)	64	256x128
	18-19	**2 x Non-bt-1D**	64	256x128
	20	**Deconvolution**(upsampling)	16	512x256
	21-22	**2 x Non-bt-1D**	16	512x256
	23	**Deconvolution**(upsampling)	C	1024x512

PROPOSED TRAINING METHOD GRADUAL GROUPING



Training procedure where the train time optimization happens in the higher dimensional space of dense convolutions and gradually evolves towards grouped convolutions.

- Start with a dense convolution and multiply the blue edges by a parameter α.
- Decrease α gradually during training time from 1 and by the end of the training α becomes 0.
- In fine tuning phase, α remains 0. Finally at test time, the convolutions can be implemented as grouped convolutions which gives better efficiency.

RESULTS

Models	IOU	Params	GFLOPs
ERFNet-pretrained	72.10	2038448	27.705
D*-proposed	68.39	431312	5.773
DG2*-proposed	66.10	279760	4.029
DG4*-proposed	63.80	203984	3.156

Our method gives a 5X reduction in FLOPs with only 4% degradation in accuracy.

Blue points representing models trained by gradual grouping gives the best performance tradeoffs.

Our prime focus was on obtaining compressed models with < 20 GFLOPs and with minimal loss in accuracy.

We pretrain our proposed encoder on Imagenet dataset using gradual grouping, and then attach the light weight decoder to it.

Selective application of groups (green points) hardly degrades the accuracy while still giving a reasonable **reduction** in GFLOP of **1.5X over the baseline ERFNet** which runs at 27.7 GFLOPs