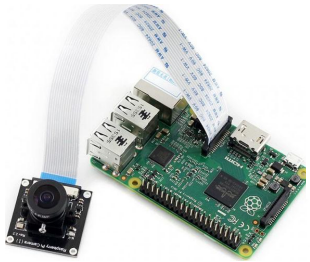
Efficient Semantic Segmentation using Gradual Grouping

Nikitha Vallurupalli[1], Sriharsha Annamaneni[1], Girish Varma[1], C V Jawahar[1], Manu Mathew[2], Soyeb Nagori[2]

IIIT Hyderabad[1], TI Bangalore[2]

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

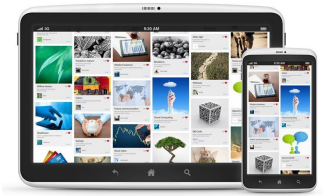
Image / annotation from cityscapes dataset

Cloud not an option

Large latencies for real-time output.

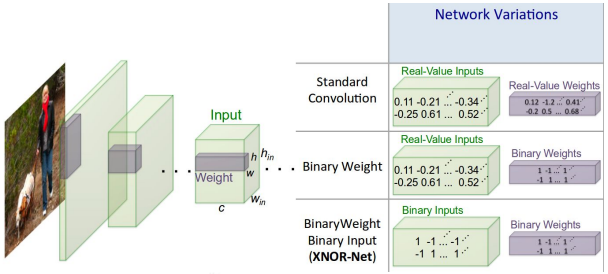
Violates user privacy.

Consumes network bandwidth.



Model Compression: Previous Approaches

Quantization



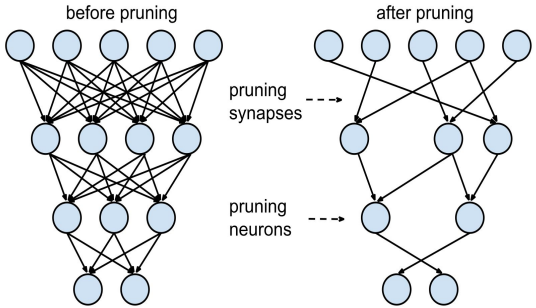
$$\hat{W}_{ij} = \begin{cases} 1 & \text{if } W_{ij} \geq 0, \\ -1 & \text{if } W_{ij} < 0. \end{cases}$$

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

High precision arithmetic not essential for obtaining high performance.

This results in memory savings and faster computation.

Pruning

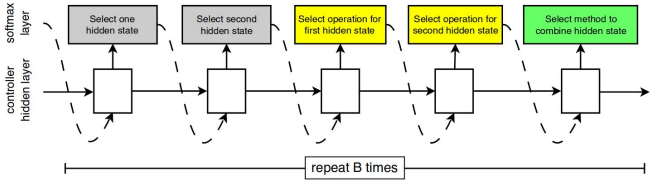


Learning both weights and connections for efficient Neural Network
Song Han, Jeff Pool, John Tran and William J.Dally NIPS 2015

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.

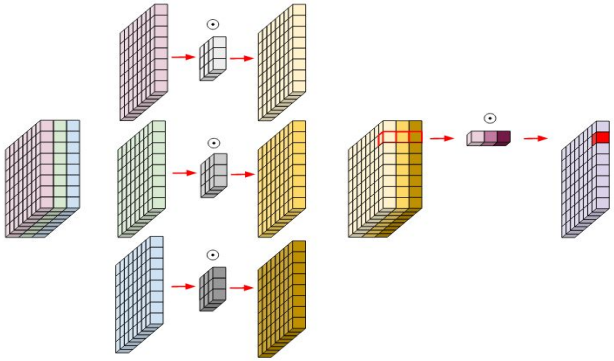
Problems with Previous Approaches

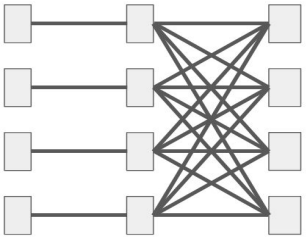
What we discovered in our past investigations into existing approaches...

- Bulky initial model needs to be trained:
  - Additional pruning-quantization-training cycles required to recover accuracy drops.
  - Limits the type of models that can be trained.
- Performance drops:
  - Binarization leads to significant performance drops.
  - Greedy layer-by-layer pruning leads to significant approximation errors.
- Architecture design is done by making **intelligent guesses** and **trial & error**.
- Neural architecture search requires vast amounts of resources.
  - It is mostly a **heuristic brute force search** with no guarantees.
  - Infeasible for large scale problems.

Few Recent Techniques to make CNN's efficient

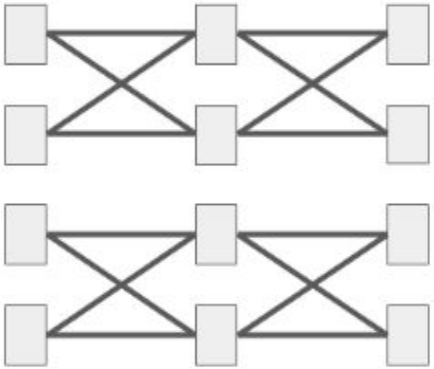
Depthwise separable convolutions





3x3 depthwise followed by 1x1 pointwise convolutions

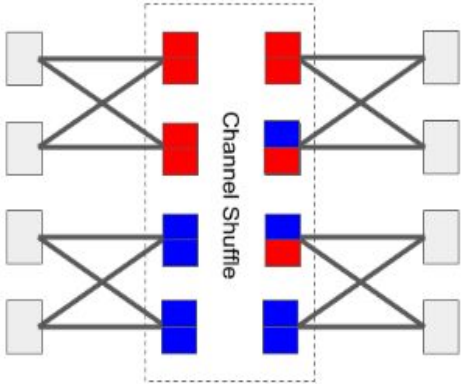
Grouped Convolutions



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*×*n* channels.

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

ERFNet as Baseline

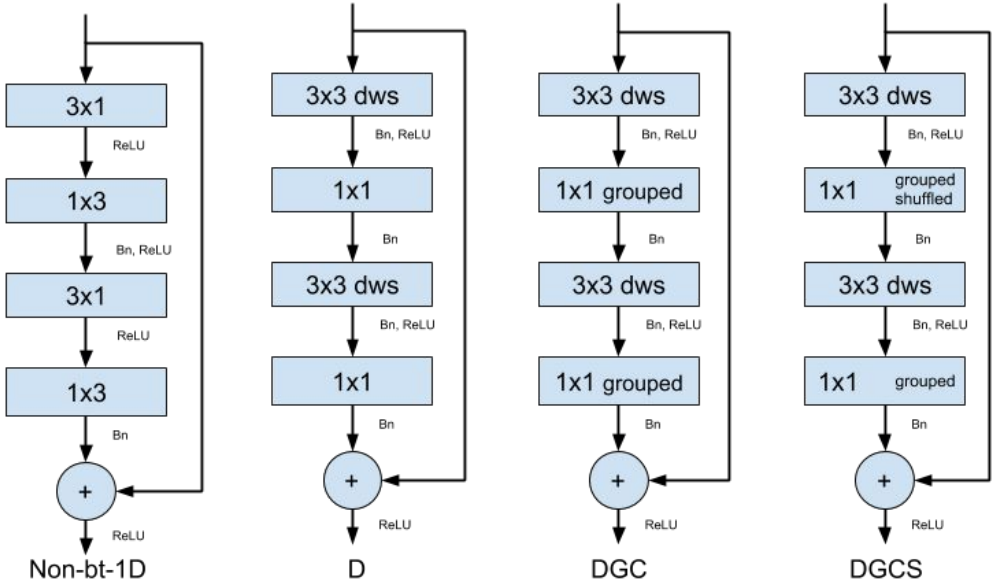
- **72.1** mIoU on cityscapes with **27 GFLOP's.**
- Architectural block uses spatially separable **3x1** and **1x3** convolutions with **residual** connections.
- Separable encoder and decoder modules.
- At 1024 x 512 resolution, network achieves **24ms (41 FPS)** on a single Titan X GPU.
- Best available trade-off between segmentation accuracy and computational resources.

Model	**Class-IoU**	**FPS** **ms** (Resolution)
SegNet	57.0	289 3.5 (1280x720)
ENet	58.	76.9 13 (1024x512)
SQ	59.8	n/a
ERFNet	**72.1**	41.7 24 (1024x512)

"ERFNet: Efficient Residual Factorized ConvNet for Real-time Semantic Segmentation", E. Romera, J. M. Alvarez, L. M. Bergasa and R. Arroyo, Transactions on Intelligent Transportation Systems (T-IT) '17

Micro Level Architecture

We experiment with different convolutional modules by making each module more efficient at every stage.



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

Approach: Effect of different micro level architectural units

Models	IOU	Params	GFLOPs
ERFNet	70.45	2038448	27.705
D*	68.55	547120	10.597
DG2*	65.35	395568	8.852
DG4*	61.42	319792	7.980
DG8*	59.15	281904	7.543
DG2S*	65.36	395568	8.852
DG4S*	61.27	319792	7.980
DG8S*	59.89	281904	7.543

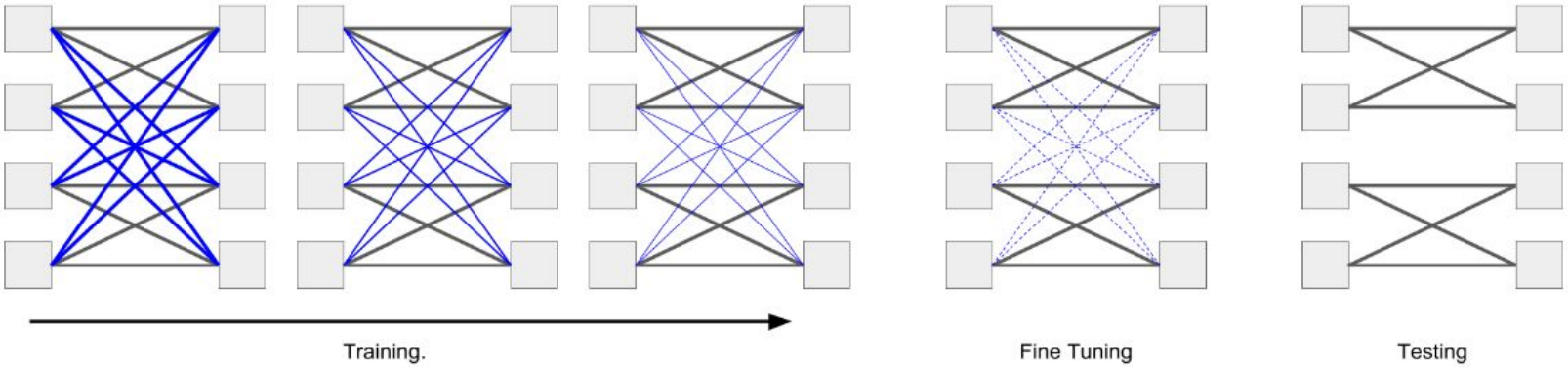
- Accuracy degradation is over **10%** when model is compressed by **4x**
  - Micro architectural change is applied to every layer in the encoder.
  - Decoder is not yet efficiently compressed.
  - Accuracy degrades as number of groups increase.

Approach: Effect of Selective Application

Models	IOU	Params	GFlops
D	69.26	1291648	19.025
DG2	69.71	1238960	18.998
DG4	68.98	1202096	18.595
DG8	69.57	1183664	18.394
DG2S	70.62	1238960	18.998
DG4S	69.59	1202096	18.595
DG8S	69.57	1183664	18.394

- Accuracy almost remains the same, but the models are **not highly compressed**.
  - Leaving few initial layers, micro architectural changes are applied only to the later layers in the encoder.
  - Decoder is not yet efficiently compressed.

Proposed Method: Training using 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.

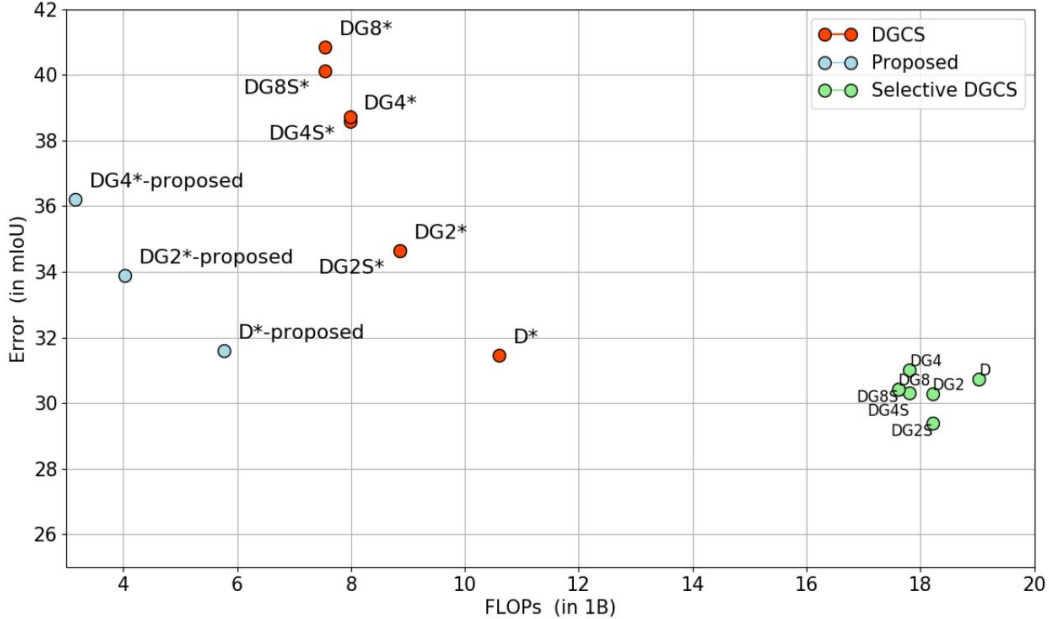
Results

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.

- Our proposed models are having FLOPs ranging from 5.77 GFLOPs to 3.15 GFLOPs while the best accuracy is around 68%. Since the optimization that is happening at training time is in the higher dimensional space of dense convolutions, we can obtain better accuracies than traditional training for grouped convolution.
- In these proposed models, we further reduce the number of FLOP's by attaching an efficient light weight decoder by changing the 3x3 deconvolution(upsampling operations to 1x1 operations).
- We pretrain our proposed encoder model on Imagenet dataset using gradual grouping, and then attach the light weight decoder to it.

Performance Trade-off Graph



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

Thank you!