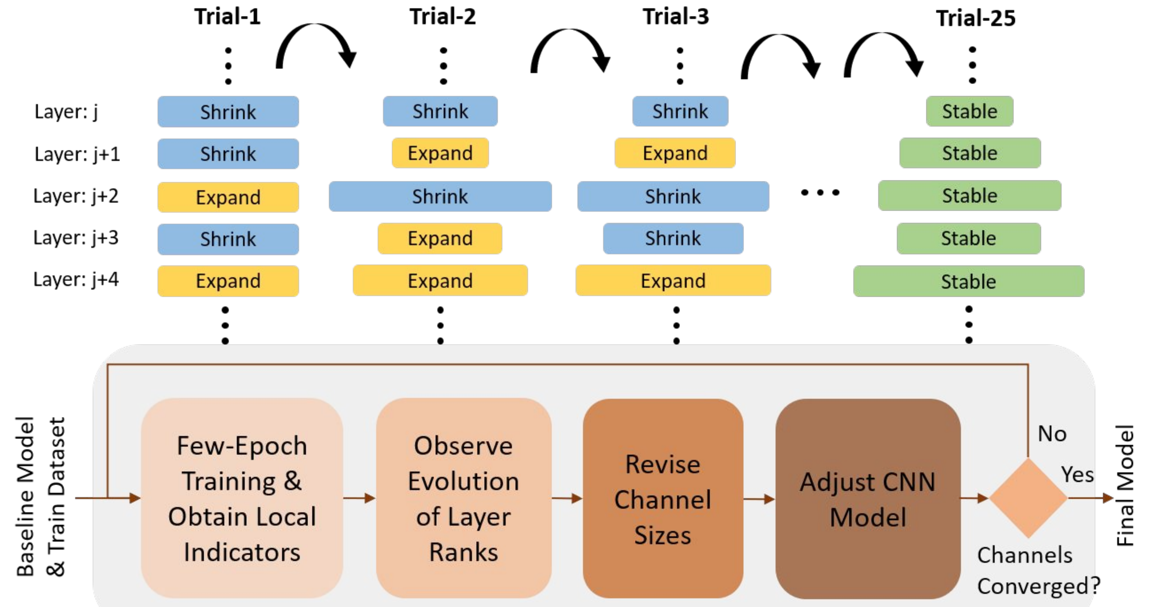
CONet: Channel Optimization for Convolutional Neural Networks

Mahdi S. Hosseini[1*], Jia Shu Zhang[2*], Zhe Li[2*], Andre Fu[2*], Jingxuan Su[2], Mathieu Tuli[2], Konstantinos N. Plataniotis[2]

2021 ICCV OCTOBER 11-17 VIRTUAL

Background and Motivation

This paper focuses on automated optimization of the channel size of CNNs for Image classification problem.



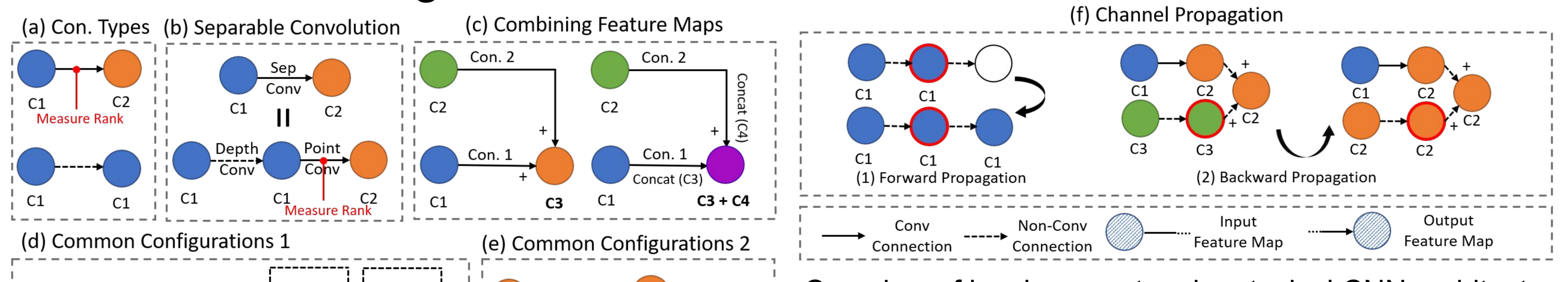
CONet Overview. The algorithm begins with a small baseline model and trains a few epochs using the train dataset. Every channel dynamically shrinks/expands based on the rank evolution. This is repeated until all channel sizes have stabilized.

Issues of Existing Channel Scaling Methods

Current SOTA methods (i.e. hand-crafted or NAS) for designing cell-topologies in CNNs use heuristic approaches for setting up the channel size in different convolutional layers. The rule of thumb is to increase the number of channels for deeper layers.

Local Indicators and Metric Development

Central to our approach is to develop a metric to probe independent channels in CNN and evaluate how well the convolutional weights are trained.



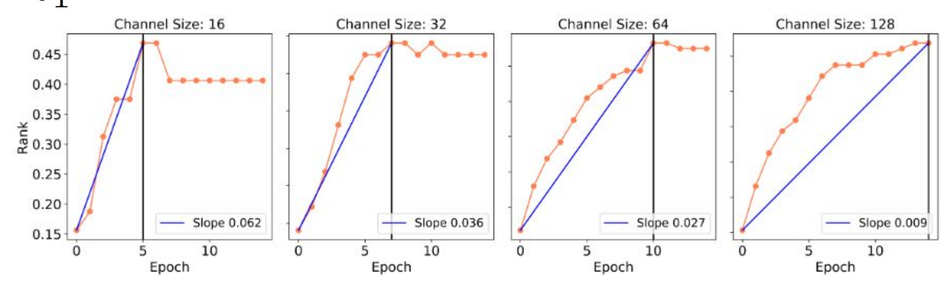
Overview of local connectors in a typical CNN architectures to identify independent channels for probing: (a) Elementary connections; (b) Rank measure on separable conv; (c) Sum/Concat of feature maps; (d) configurations within residual net; (e) configuration within inception / cell-based nets; (f) Propagation of Channel Constraints.

Metrics for Convolutional Layers

We define **rank-slope** by computing the relative ratio of the rank gain to the number of epochs taken to plateau.

$$S = \frac{\overline{R}(\widehat{W}_d^{t_2}) - \overline{R}(\widehat{W}_d^{t_1})}{t_2 - t_1}$$

R: #singular values / channel size



CONet Algorithm

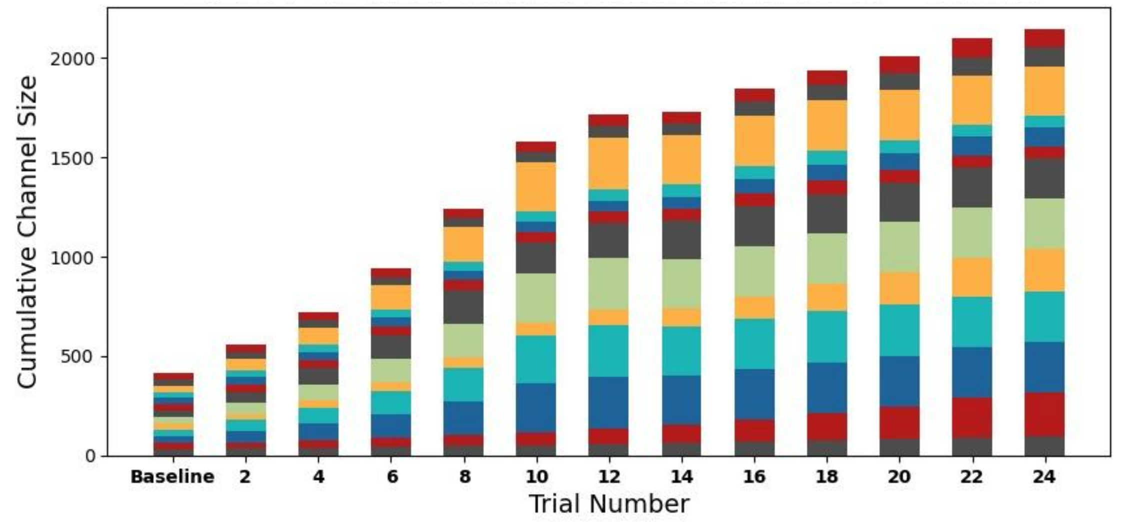
https://github.com/mahdihosseini/CONet

Algorithm 2 Channel Size Optimizaion

Input: Network, hyperparameters $\delta, \gamma, \mu, \phi$
Output: Optimized Network

for trial **in** trials **do**
  Train current model for a few epochs
  Compute average rank ($\overline{R}$) per layer
  Compute average MC ($\overline{\kappa}$) per layer
  From $\overline{R}$ get rank-slope (S) per layer
  for layer **in** network **do**
    Set action to Expand
    if S[layer] $< \delta$ or $\overline{\kappa}$[layer] $> \mu$ **then**
      Set action to Shrink
    if Last action $\neq$ current action **then**
      if $\phi < \gamma$ **then**
        Stop, channel sizes have converged
      $\phi \leftarrow \phi/2$
    Calculate New Channels
  Instantiate model with new channels

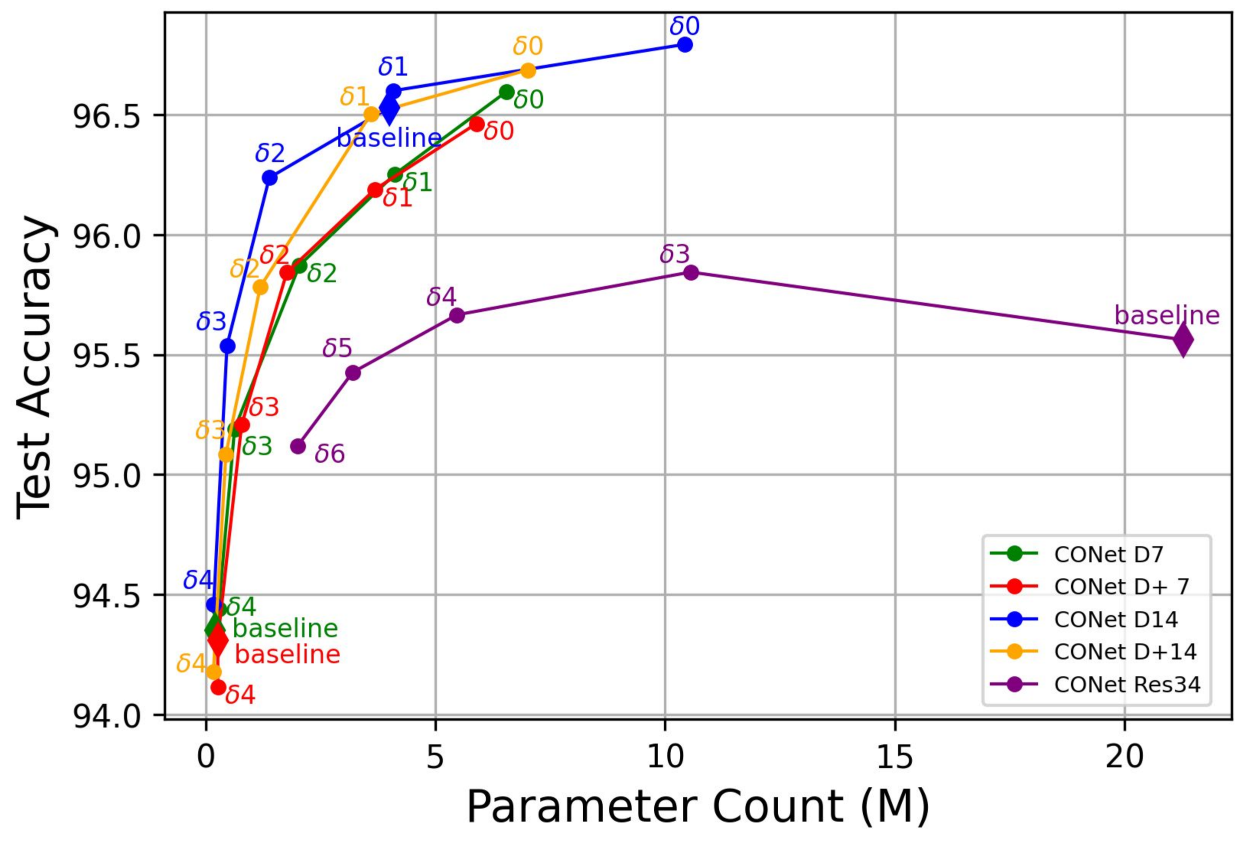
R	List of input and output ranks for each layer.
κ	List of input and output Mapping condition.
$\overline{R}$	Rank averages.
$\overline{\kappa}$	Mapping condition averages.
S	Rank average slopes.
δ	*Rank average slopes threshold.*
μ	*Mapping condition threshold.*
C_ℓ^o	Old channel sizes, per layer (ℓ).
C_ℓ^n	New channel sizes, per layer (ℓ).
ω_ℓ	Last operation list, per layer (ℓ).
ϕ	Scale factors of channel sizes.
γ	*Stopping condition for scaling factor.*



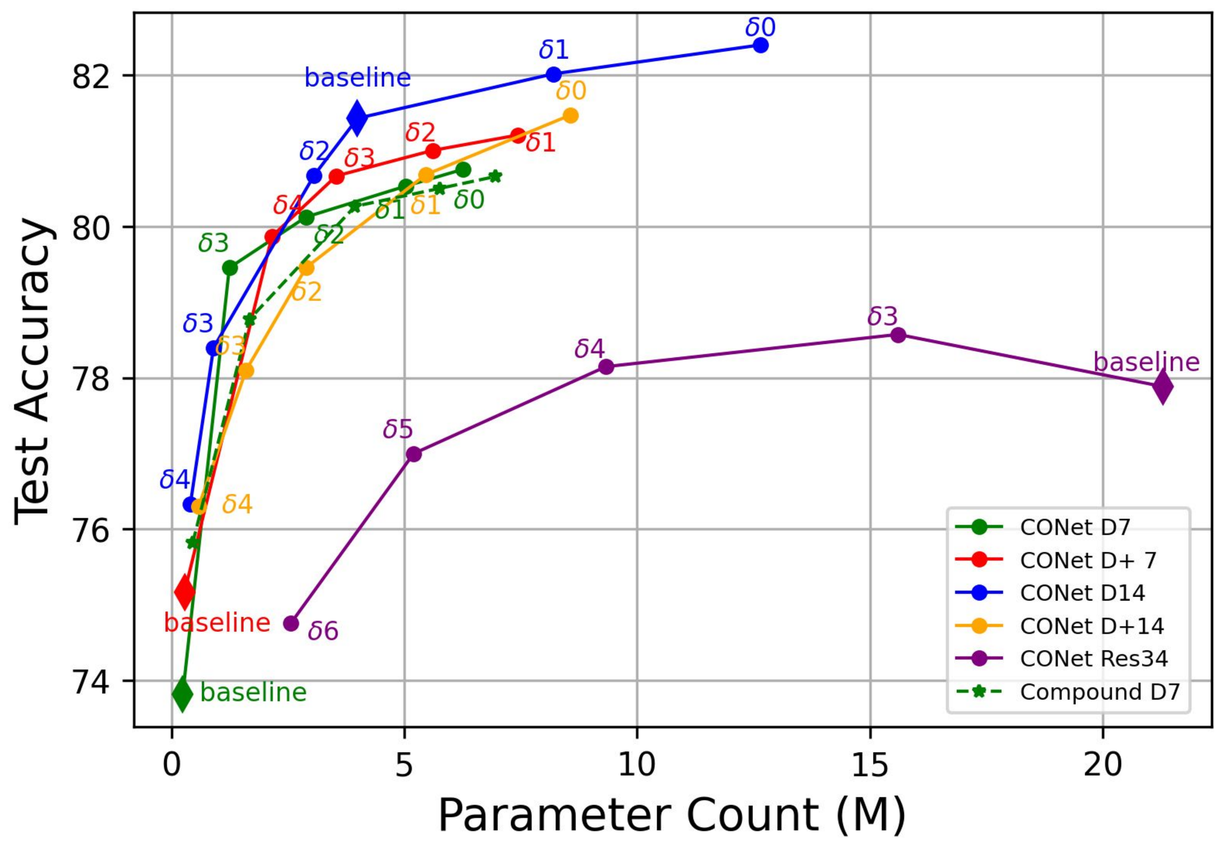
Channel size evolution example of DARTS7 using CONet algorithm on CIFAR100.

Results

CIFAR10



CIFAR100



ImageNet

Architecture	ImageNet Results			Architecture Search			ImageNet Training Setup			
	Top-1 (%)	Top-5 (%)	Params (M)	Search Cost (GPU-days)	Searched On Dataset	GPUs	Training Epochs	Batch Size	Optimizer	LR-Schedular
NASNet-A[51]	74.0	91.6	5.3	3-4	CIFAR10	450	250	128	SGD	StepLR: Gamma=0.97,Step-size=1
AmoebaNet-C[28]	75.7	92.4	6.4	7	CIFAR10	250	250	128	SGD	StepLR: Gamma=0.97,Step-size=2
DARTS-7[22]	52.9	76.9	0.5	4	CIFAR10	1	200	128	SGD	StepLR: Gamma=0.5,Step-size=25
DARTS-14[22]	73.3	91.3	4.7	4	CIFAR10	1	250	128	SGD	StepLR: Gamma=0.97,Step-size=1
GHN[47]	73.0	91.3	6.1	0.84	CIFAR10	1	250	128	SGD	StepLR: Gamma=0.97,Step-size=1
ProxylessNAS[3]	74.6	92.2	5.8	8.3	ImageNet	—	300	256	Adam	—
SNAS[42]	72.7	90.8	4.3	1.5	CIFAR10	1	250	128	SGD	StepLR: Gamma=0.97,Step-size=1
BayesNAS[49]	73.5	91.1	3.9	0.2	CIFAR10	1	250	128	SGD	StepLR: Gamma=0.97,Step-size=1
P-DARTS[5]	75.6	92.6	4.9	0.3	CIFAR10	8	250	1024	SGD	OneCycleLR: Linear Annealing
GDAS[12]	72.5	90.9	4.4	0.17	CIFAR10	1	250	128	SGD	StepLR: Gamma=0.97,Step-size=1
PC-DARTS[43]	75.8	92.7	5.3	3.83	ImageNet	8	250	1024	SGD	OneCycleLR: Linear Annealing
DARTS+[21]	76.3	92.8	5.1	0.2	CIFAR100	1	800	2048	SGD	OneCycleLR: Cosine Annealing
NASP[44]	73.7	91.4	9.5	0.1	CIFAR10	1	250	128	SGD	StepLR: Gamma=0.97,Step-size=1
SGAS[19]	75.9	92.7	5.4	0.25	CIFAR10	1	250	1024	SGD	OneCycleLR: Linear Annealing
MiLeNAS[13]	75.3	92.4	5.3	0.3	CIFAR10	1	250	128	SGD	StepLR: Gamma=0.97,Step-size=1
SDARTS[4]	75.3	92.2	3.3	1.3	CIFAR10	1	250	1024	SGD	OneCycleLR: Linear Annealing
FairDARTS[7]	75.6	92.6	4.3	3	ImageNet	1	250	1024	SGD	OneCycleLR: Linear Annealing
DARTS7(δ_3)	67.9	88.0	1.8	0.3	CIFAR100	1	200	128	SGD	StepLR: Gamma=0.5,Step-size=25
DARTS7(δ_2)	72.0	90.6	3.4	0.4	CIFAR100	1	200	128	SGD	StepLR: Gamma=0.5,Step-size=25
DARTS7(δ_1)	74.1	91.7	5.6	0.5	CIFAR100	1	200	128	SGD	StepLR: Gamma=0.5,Step-size=25
DARTS-14(δ_2)	67.2	87.6	1.8	0.8	CIFAR10	1	200	128	SGD	StepLR: Gamma=0.97,Step-size=1
DARTS-14(δ_1)	74.0	91.8	4.8	0.8	CIFAR10	1	200	128	SGD	StepLR: Gamma=0.97,Step-size=1
DARTS14(δ_1)	**76.6**	**93.2**	9.0	0.9	CIFAR100	1	200	128	SGD	StepLR: Gamma=0.5,Step-size=25

1: University of New Brunswick; 2: University of Toronto; *: equal contribution
Contact: mahdi.hosseini@unb.ca