Efficient and Adaptive ConvNets for Face Recognition: A Channel Prioritized Approach

Deep Learning for Computer Vision (DLCV) Competition, 2018 Spring

Chun-Min Chang, Chia-Ching Lin, Tsun-Hsien Tang

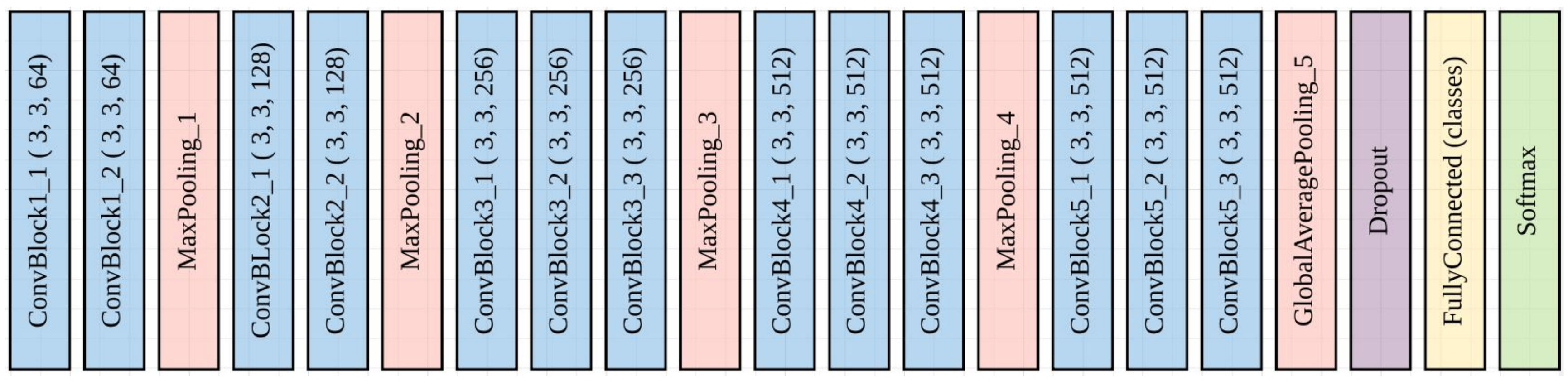
Motivation

- Deploying ConvNets on resource-limited platforms is often impeded by constraints in size and speed
- Need an one-for-all model to meet different power consumption, accuracy, and latency requirements

Therefore, we propose to train a ConvNet to achieve model reduction and dynamically trade-offs between performance and resource

Base Model Architecture

- VGG-like ConvNets using global average pooling
- Every conv layer is followed by a BN layer



Method: Channel Prioritization for Sparsity and Adaptivity

- We leverage scaling factors of batch normalization since these factors represent the importance of channels
- We enforce to sparsify and prioritize channels along their indices using three techniques

(i) Monotonically decreasing initialization

$$\gamma_l^{(k)} = 1 - \frac{k-1}{N_l}, \quad k = 1, \dots, N_l$$

(ii) Monotonicity-induced penalty

$$L_{m,l}^{(k)} = \begin{cases} \gamma_l^{(k+1)} - \gamma_l^{(k)} & , \text{ if } \gamma_l^{(k+1)} > \gamma_l^{(k)} \\ 0 & , \text{ otherwise} \end{cases}$$

(iii) Sparsity penalty

$$|\gamma_l^{(k)}|$$

$$L_{obj} = Loss + \lambda_s \sum_{k,l} |\gamma_l^{(k)}| + \lambda_m \sum_{k,l} L_{m,l}^{(k)}$$

- Based on the learned channel priority, a small sub-network can be built by removing (100-p)% insignificant channels. By jointly training the sub-network at various utilization levels, we can obtain the adaptive property

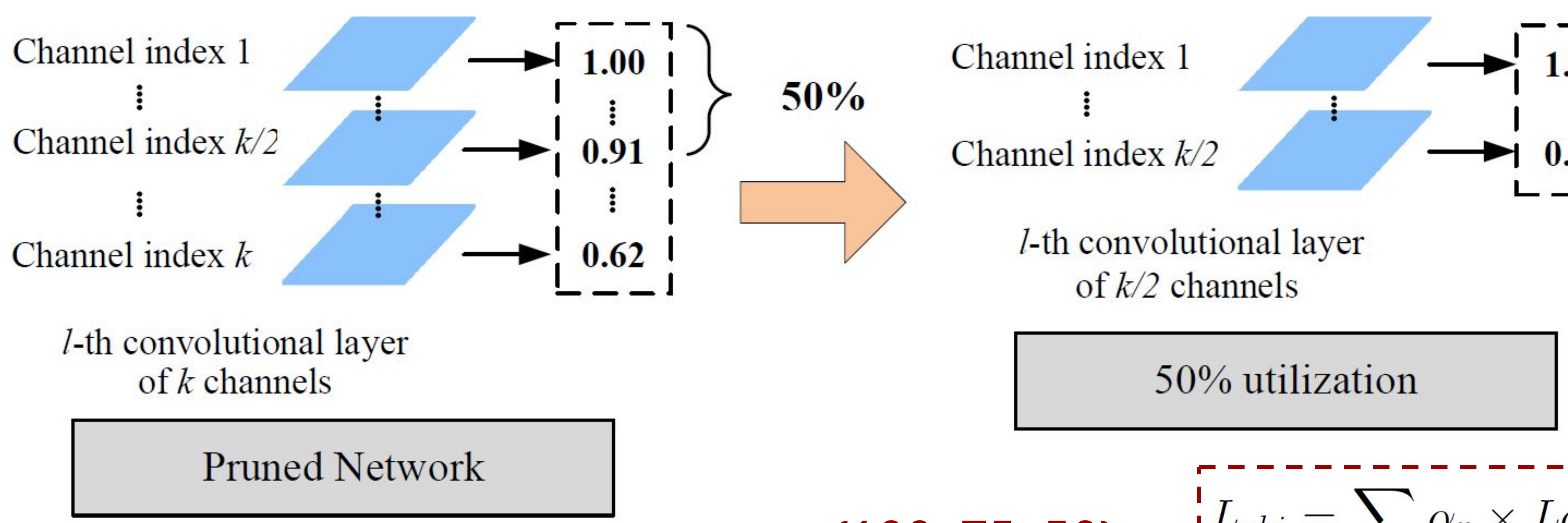
Pre-training stage

1. Monotonically decreasing initialization on scaling factors
2. Train with sparsity and monotonicity-induced penalties
3. Prune channels by a global threshold

Magnitude-based Pruning threshold: 0.1

Fine-tuning stage

4. Fix all parameters in BN layers
5. Define and sum up the losses at different utilization levels
6. Fine-tune the pruned network with the aggregated loss



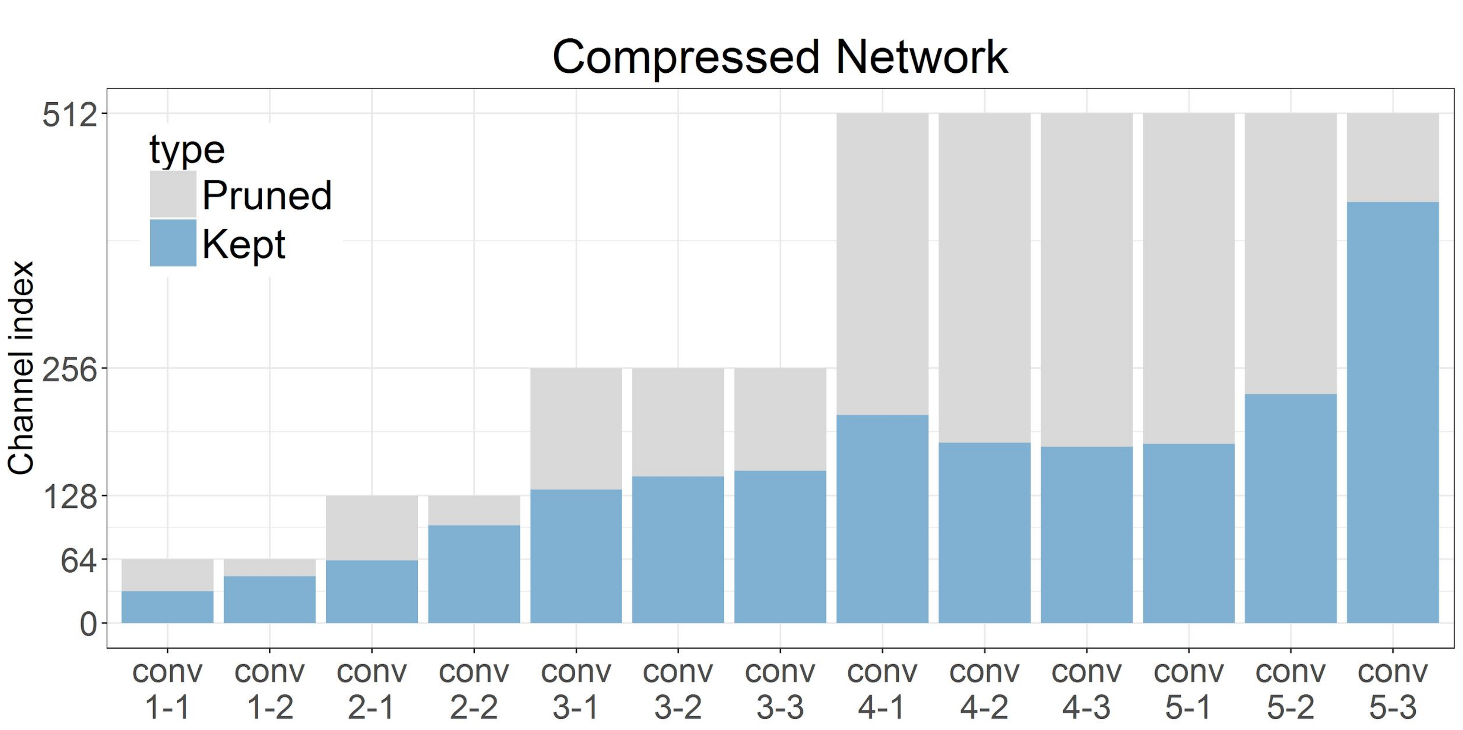
p={100, 75, 50}

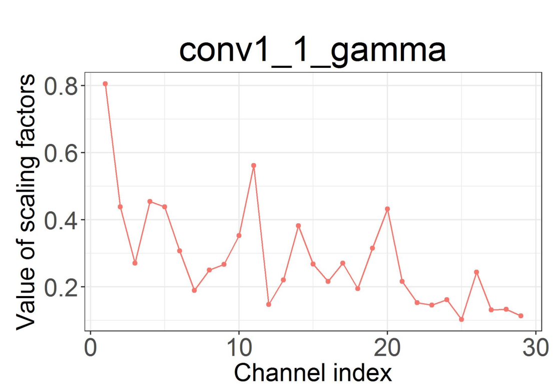
$$L_{obj} = \sum_{p \in P} \alpha_p \times Loss_p$$

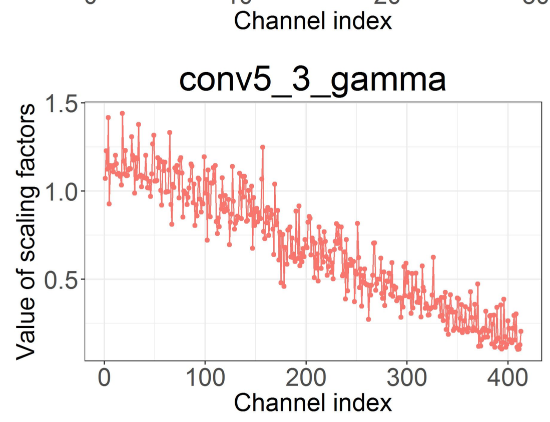
Experiment Results

	Accuracy (%)	Multi-Adds	Parameters
VGG-16 (baseline)	86.28	24.71×10^9	15.91×10^6
MobileNet-224 [1]	65.31	5.73×10^9	5.62×10^6
SqueezeNet-v1.0 [2]	60.02	**8.61×10^8**	1.75×10^6
Extracted feature+NearestOne*	84.05	7.30×10^9	4.05×10^6
Our (100%)	**86.38**	6.87×10^9	4.05×10^6
Our (75%)	81.40	3.84×10^9	2.28×10^6
Our (50%)	78.47	1.68×10^9	**1.01×10^6**

*Features extracted from the best CNN model, compute centers for all classes using training and validation data, and predict the test class with the smallest cosine similarity







Training Details

- Objective: softmax loss+center_loss

Hyper-parameters		Optimizer and Augmentation	
lambda_s	1e-3	Optimizer	Adam, beta1=0.5
lambda_m	4e-5	Batch size	64
lambda_c	1e-3	lr	1e-4
alpha_c	0.5	Rotation	[-45, 45]
weight_decay	1e-5	Scale	+- 15%
dropout	0.8	Shift	+- 15 %
early stop	10	Flip	with prob = 0.5

Conclusions

- We propose a generalized method to train an efficient and adaptive CNNs, also suitable for face recognition
- ConvNet with prioritized channels can adapt to various requirements
- Experiments showed 2.5, 14.5x reduction with 0, 8% accuracy drop

[1] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).
[2] Iandola, Forrest N., et al. "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size." *arXiv preprint arXiv:1602.07360* (2016).