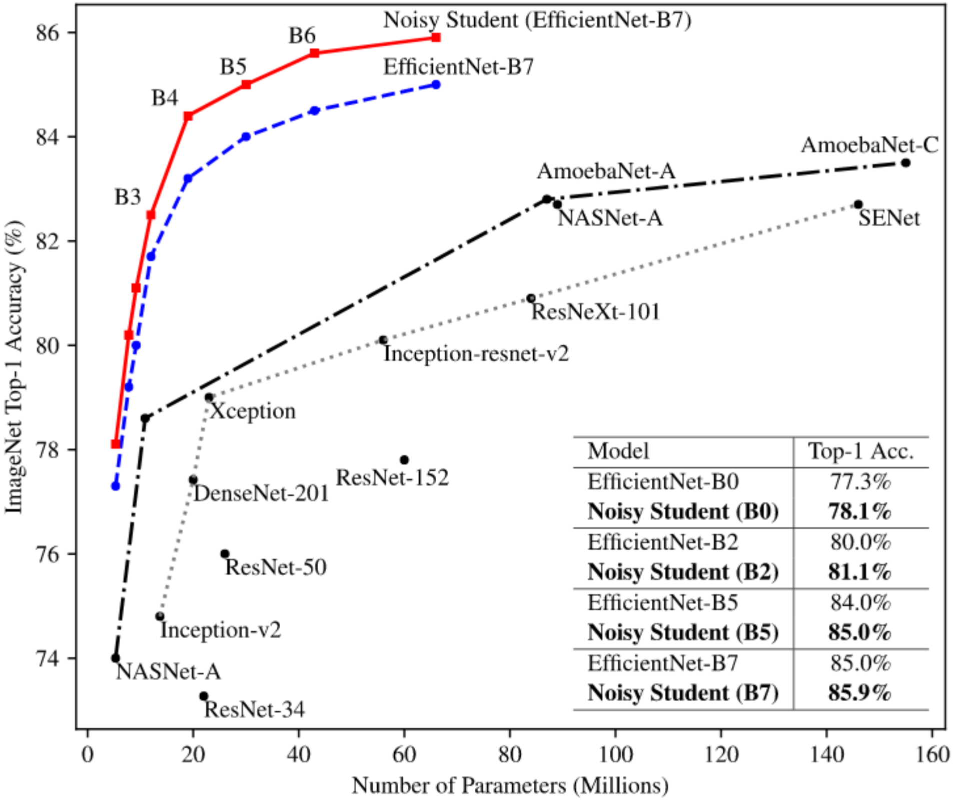
Self-training with Noisy Student improves ImageNet classification

https://arxiv.org/abs/1911.04252

Results

RANK	METHOD	TOP 1 ACCURACY	TOP 5 ACCURACY	NUMBER OF PARAMS	EXTRA TRAINING DATA	PAPER TITLE	YEAR
1	**FixEfficientNet-L2**	88.5%	98.7%	480M	✓	Fixing the train-test resolution discrepancy: FixEfficientNet	2020
2	**NoisyStudent** (EfficientNet-L2)	88.4%	98.7%	480M	✓	Self-training with Noisy Student improves ImageNet classification	2020
3	**BiT-L** (ResNet)	87.8%			✓	Large Scale Learning of General Visual Representations for Transfer	2019
4	**NoisyStudent** (EfficientNet-L2)	87.4%	98.2%	480M	✓	Self-training with Noisy Student improves ImageNet classification	2019
5	**FixEfficientNet-B7**	87.1%	98.2%	66M	✓	Fixing the train-test resolution discrepancy: FixEfficientNet	2020
6	**NoisyStudent** (EfficientNet-B7)	86.9%	98.1%	66M	✓	Self-training with Noisy Student improves ImageNet classification	2020
7	**FixEfficientNet-B6**	86.7%	98.0%	43M	✓	Fixing the train-test resolution discrepancy: FixEfficientNet	2020
8	**FixResNeXt-101 32x48d**	86.4%	98.0%	829M	✓	Fixing the train-test resolution discrepancy	2019
9	**NoisyStudent** (EfficientNet-B6)	86.4%	97.9%	43M	✓	Self-training with Noisy Student improves ImageNet classification	2020
10	**FixEfficientNet-B5**	86.4%	97.9%	30M	✓	Fixing the train-test resolution discrepancy: FixEfficientNet	2020
11	**NoisyStudent** (EfficientNet-B5)	86.1%	97.8%	30M	✓	Self-training with Noisy Student improves ImageNet classification	2020
12	**FixEfficientNet-B4**	85.9%	97.7%	19M	✓	Fixing the train-test resolution discrepancy: FixEfficientNet	2020
13	**MaxUp** (Fix-EfficientNet-B8)	85.8%			×	MaxUp: A Simple Way to Improve Generalization of Neural Network Training	2020
14	**FixEfficientNet-B8**	85.7%	97.6%	88M	×	Fixing the train-test resolution discrepancy: FixEfficientNet	2020
15	**AdvProp** (EfficientNet-B8)	85.5%	97.3%	88M	×	Adversarial Examples Improve Image Recognition	2019



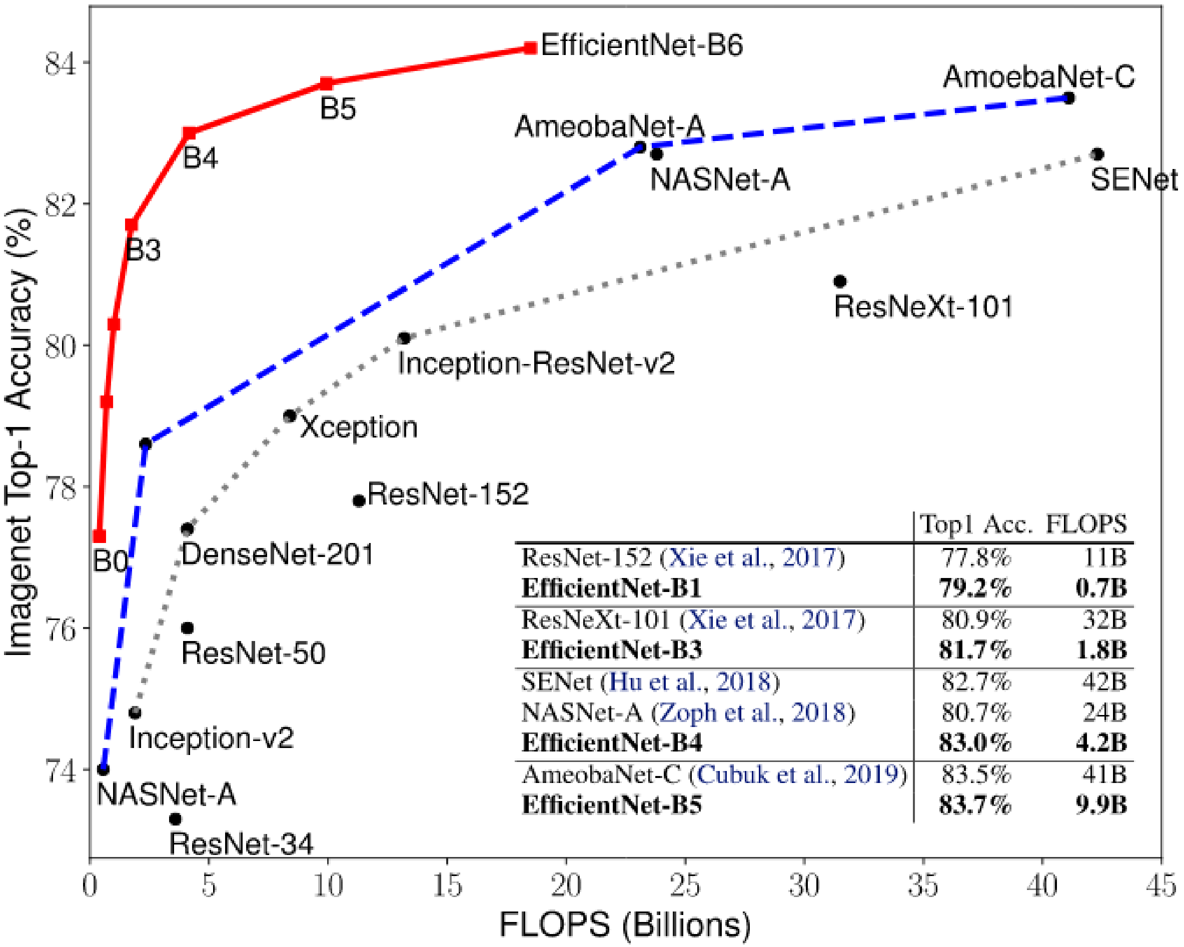
https://paperswithcode.com/sota/image-classification-on-imagenet (25 mar 2020)

What is this paper about?

- SOTA in image classification (ImageNet)
- Clever way of using external data (semi-supervised learning)
- Iterative enlarging model
- SOTA in robustness

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

https://arxiv.org/abs/1905.11946



	Top1 Acc.	FLOPS
ResNet-152 (Xie et al., 2017)	77.8%	11B
EfficientNet-B1	**79.2%**	**0.7B**
ResNeXt-101 (Xie et al., 2017)	80.9%	32B
EfficientNet-B3	**81.7%**	**1.8B**
SENet (Hu et al., 2018)	82.7%	42B
NASNet-A (Zoph et al., 2018)	80.7%	24B
EfficientNet-B4	**83.0%**	**4.2B**
AmeobaNet-C (Cubuk et al., 2019)	83.5%	41B
EfficientNet-B5	**83.7%**	**9.9B**

Figure 5. **FLOPS vs. ImageNet Accuracy –** Similar to Figure 1 except it compares FLOPS rather than model size.

Key contributions

- New architecture using NAS (neural architecture search)

- Rules for effective scaling the network

Scaling networks

- (Depth) Number of layers
  - resnet18, resnet34;
  - resnet50, resnet101, resnet150
- (Width) Number of channels
  - MNASNet x1, MNASNet x0.5, MNASNet x1.3, etc.
- (Resolution) Size of input image
  - MobileNets, ShuffleNets, etc.
- Which one is better? How to use all at the same time?

Architecture Search

Learning Transferable Architectures for Scalable Image Recognition
https://arxiv.org/abs/1707.07012

MnasNet: Platform-Aware Neural Architecture Search for Mobile
https://arxiv.org/abs/1807.11626

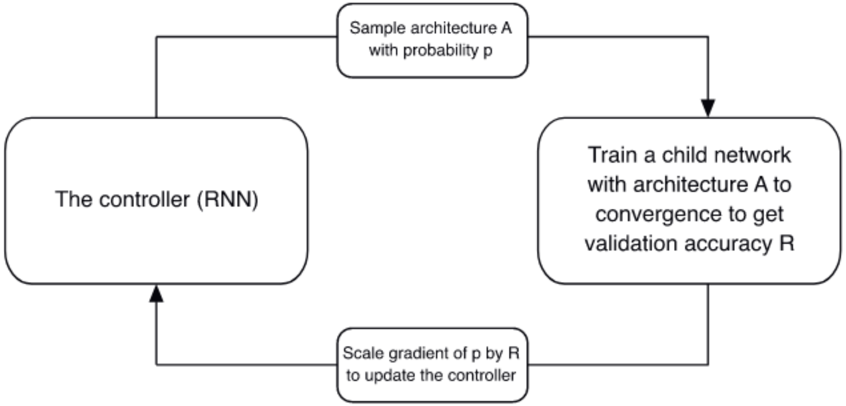


Figure 1. Overview of Neural Architecture Search [71]. A controller RNN predicts architecture A from a search space with probability p. A child network with architecture A is trained to convergence achieving accuracy R. Scale the gradients of p by R to update the RNN controller.

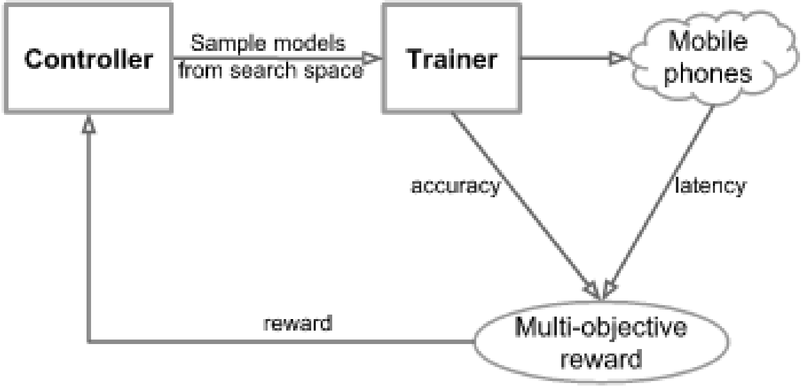
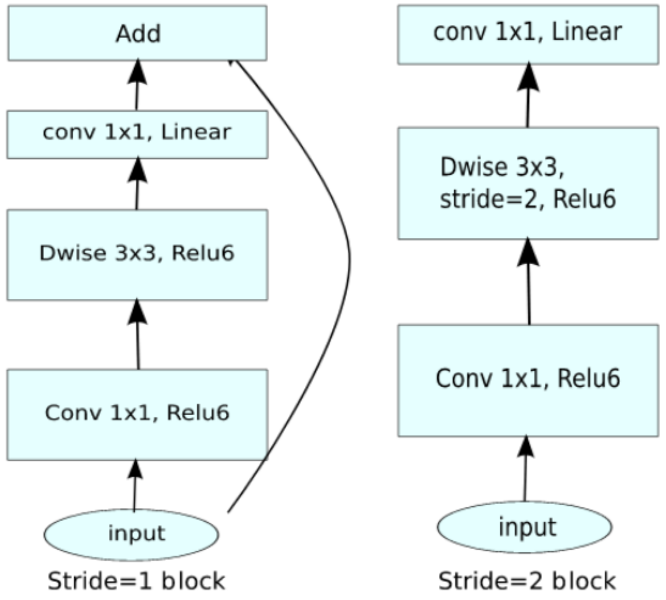


Figure 1: **An Overview of Platform-Aware Neural Architecture Search for Mobile.**

Table 1. **EfficientNet-B0 baseline network** – Each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. Notations are adopted from equation 2.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1



Input	Operator	Output
$h \times w \times k$	1x1 conv2d , ReLU6	$h \times w \times (tk)$
$h \times w \times tk$	3x3 dwise s=s, ReLU6	$\frac{h}{s} \times \frac{w}{s} \times (tk)$
$\frac{h}{s} \times \frac{w}{s} \times tk$	linear 1x1 conv2d	$\frac{h}{s} \times \frac{w}{s} \times k'$

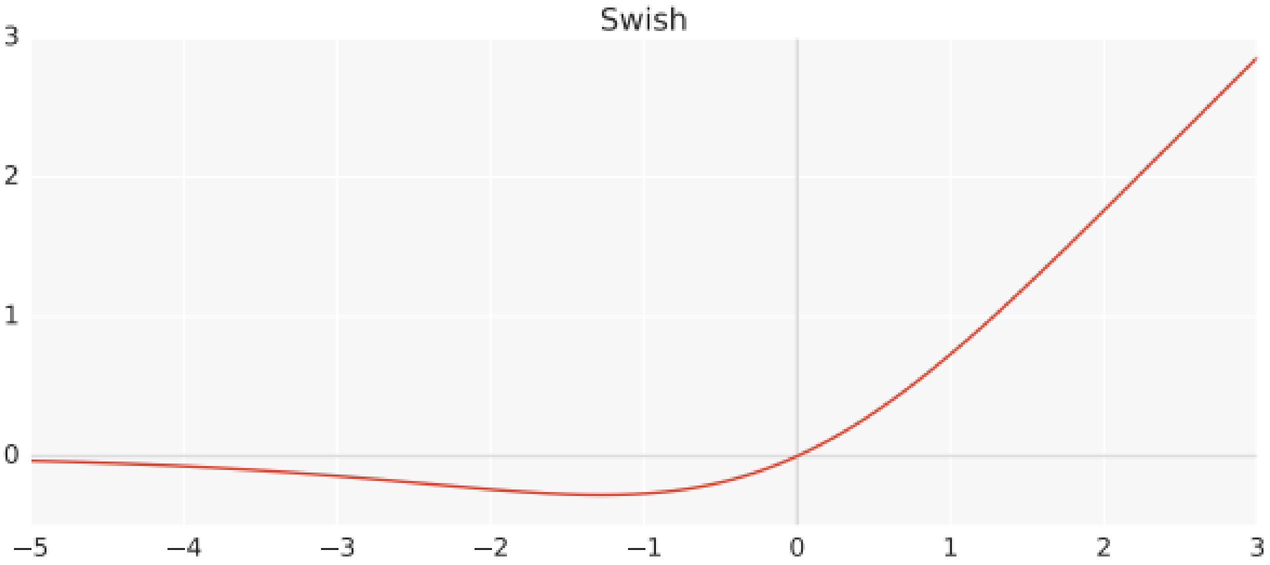
Table 1: *Bottleneck residual block* transforming from k to k' channels, with stride s, and expansion factor t.

MobileNetV2: Inverted Residuals and Linear Bottlenecks
https://arxiv.org/abs/1801.04381

Activation – Swish

swish(x) = x * sigmoid(x)

Searching for Activation Functions
https://arxiv.org/abs/1710.05941



Squeeze and excitation

Squeeze-and-Excitation Networks
https://arxiv.org/abs/1709.01507

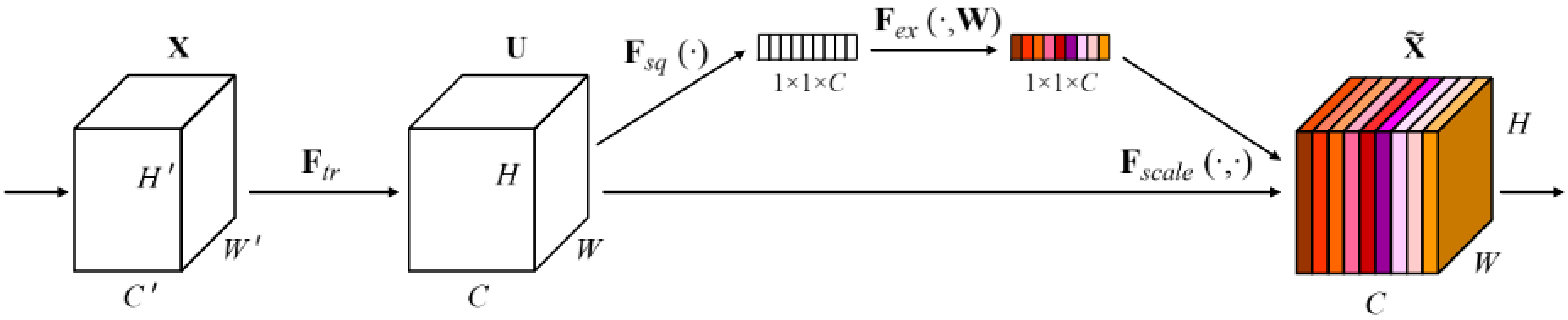


Fig. 1. A Squeeze-and-Excitation block.

Scaling

$$
\begin{aligned}
\text{depth: } & d = \alpha^{\phi} \\
\text{width: } & w = \beta^{\phi} \\
\text{resolution: } & r = \gamma^{\phi} \\
\text{s.t. } & \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\
& \alpha \geq 1, \beta \geq 1, \gamma \geq 1
\end{aligned}
$$

α= 1.2, β=1.1, γ= 1.15

Table 3. **Scaling Up MobileNets and ResNet.**

Model	FLOPS	Top-1 Acc.
Baseline MobileNetV1 (Howard et al., 2017)	0.6B	70.6%
Scale MobileNetV1 by width (w=2)	2.2B	74.2%
Scale MobileNetV1 by resolution (r=2)	2.2B	72.7%
compound scale (d=1.4, w=1.2, r=1.3)	**2.3B**	**75.6%**
Baseline MobileNetV2 (Sandler et al., 2018)	0.3B	72.0%
Scale MobileNetV2 by depth (d=4)	1.2B	76.8%
Scale MobileNetV2 by width (w=2)	1.1B	76.4%
Scale MobileNetV2 by resolution (r=2)	1.2B	74.8%
MobileNetV2 compound scale	**1.3B**	**77.4%**
Baseline ResNet-50 (He et al., 2016)	4.1B	76.0%
Scale ResNet-50 by depth (d=4)	16.2B	78.1%
Scale ResNet-50 by width (w=2)	14.7B	77.7%
Scale ResNet-50 by resolution (r=2)	16.4B	77.5%
ResNet-50 compound scale	**16.7B**	**78.8%**

Back to Noisy Student

Dataset

- Labeled dataset – ImageNet (14M images)
- Unlabeled dataset – JFT (300M images)

Revisiting Unreasonable Effectiveness of Data in Deep Learning Era
https://arxiv.org/abs/1707.02968

Require: Labeled images $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ and unlabeled images $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_m\}$.

1: Learn teacher model θ^t_* which minimizes the cross entropy loss on labeled images

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^t))$$

2: Use an unnoised teacher model to generate soft or hard pseudo labels for unlabeled images

$$\tilde{y}_i = f(\tilde{x}_i, \theta^t_*), \forall i = 1, \cdots, m$$

3: Learn an **equal-or-larger** student model θ^s_* which minimizes the cross entropy loss on labeled images and unlabeled images with **noise** added to the student model

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^s)) + \frac{1}{m}\sum_{i=1}^{m} \ell(\tilde{y}_i, f^{noised}(\tilde{x}_i, \theta^s))$$

4: Iterative training: Use the student as a teacher and go back to step 2.

Algorithm 1: Noisy Student method.

Note on student-teacher

- Student-teacher is usually used to create smaller/faster model

Distilling the Knowledge in a Neural Network
https://arxiv.org/abs/1503.02531

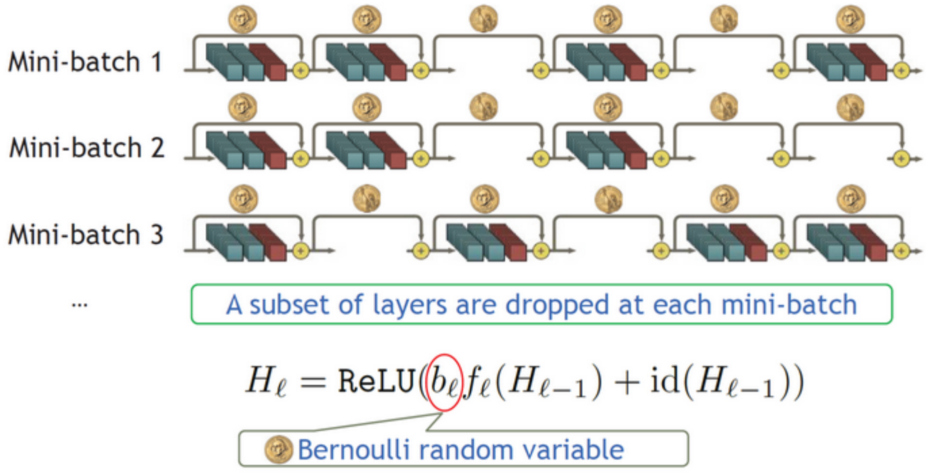
The Lottery Ticket Hypothesis:
Finding Sparse, Trainable Neural Networks
https://arxiv.org/abs/1803.03635

- In noisy student, a student is not smaller than a teacher and surpasses it

Noise

- Stochastic depth

Deep Networks with Stochastic Depth
https://arxiv.org/abs/1603.09382



Some ResBlocks are Dropped Randomly Based on Bernoulli Random Variable

- Dropout

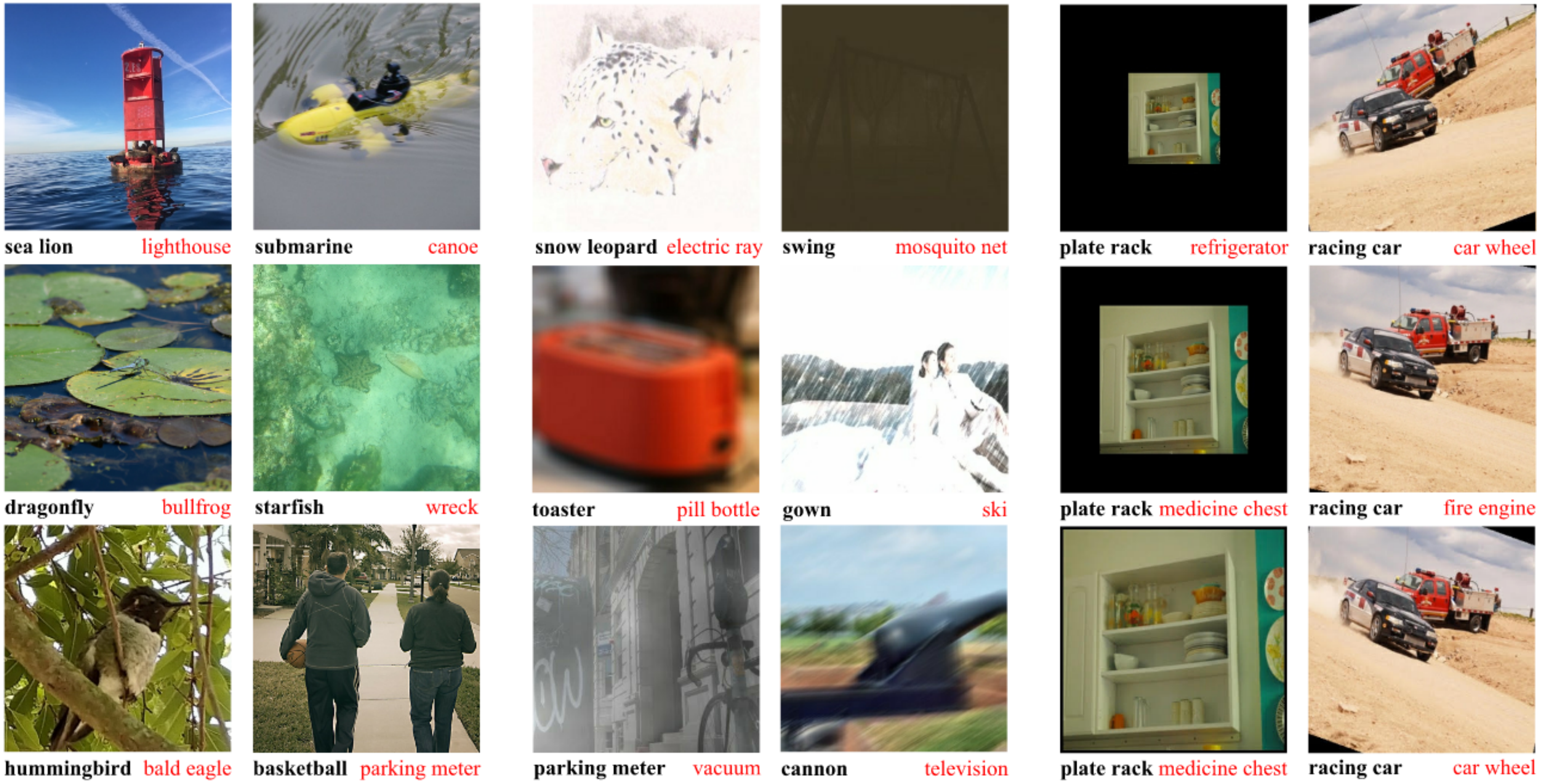
- RandAugment

RandAugment: Practical automated data augmentation with a reduced search space
https://arxiv.org/abs/1909.13719

Iterative training

- First teacher + 3 generations (iterations)
- Each 350–700 epochs
- 6 days on TPU v3 Pod with 2048 cores
- Est. ~$300,000

Robustness



(a) ImageNet-A (b) ImageNet-C (c) ImageNet-P

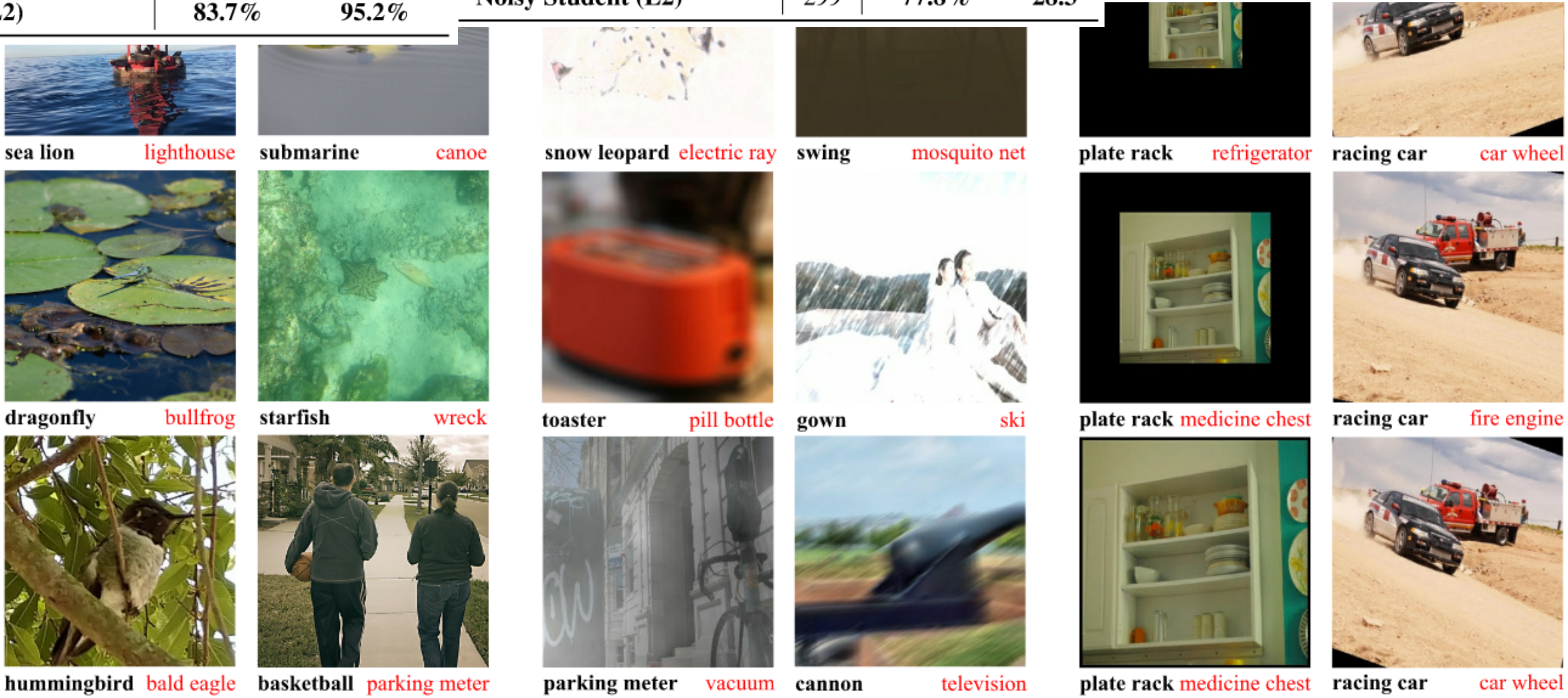
Benchmarking Neural Network Robustness to Common Corruptions and Perturbations
https://arxiv.org/abs/1903.12261

Why do deep convolutional networks generalize so poorly to small image transformations?
https://arxiv.org/abs/1805.12177

Method	Top-1 Acc.	Top-5 Acc.
ResNet-101 [30]	4.7%	-
ResNeXt-101 [30] (32x4d)	5.9%	-
ResNet-152 [30]	6.1%	-
ResNeXt-101 [30] (64x4d)	7.3%	-
DPN-98 [30]	9.4%	-
ResNeXt-101+SE [30] (32x4d)	14.2%	-
ResNeXt-101 WSL [51, 55]	61.0%	-
EfficientNet-L2	49.6%	78.6%
Noisy Student (L2)	**83.7%**	**95.2%**

Method	Res.	Top-1 Acc.	mCE
ResNet-50 [29]	224	39.0%	76.7
SIN [22]	224	45.2%	69.3
Patch Gaussian [47]	299	52.3%	60.4
ResNeXt-101 WSL [51, 55]	224	-	45.7
EfficientNet-L2	224	62.6%	47.5
Noisy Student (L2)	224	76.5%	30.0
EfficientNet-L2	299	66.6%	42.5
Noisy Student (L2)	299	**77.8%**	**28.3**

Method	Res.	Top-1 Acc.	mFR
ResNet-50 [29]	224	-	58.0
Low Pass Filter Pooling [92]	224	-	51.2
ResNeXt-101 WSL [51, 55]	224	-	27.8
EfficientNet-L2	224	80.4%	27.2
Noisy Student (L2)	224	85.2%	14.2
EfficientNet-L2	299	81.6%	23.7
Noisy Student (L2)	299	**86.4%**	**12.2**

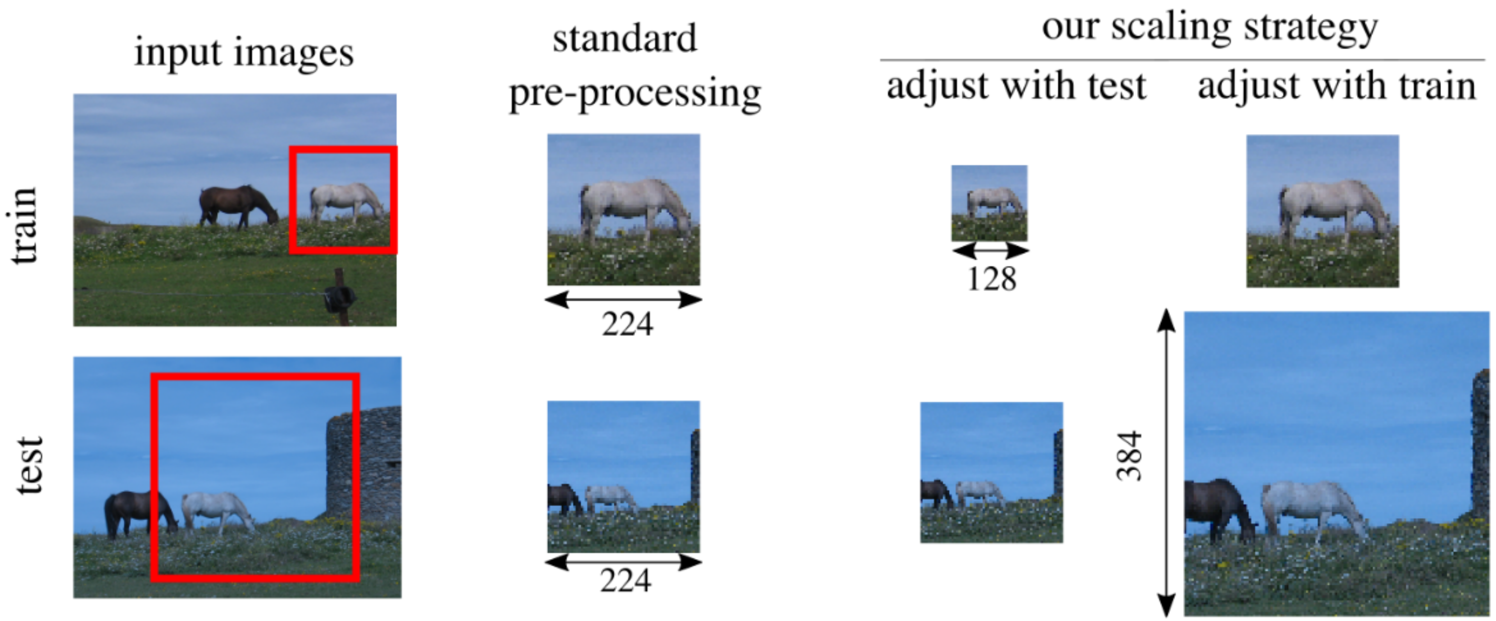


(a) ImageNet-A (b) ImageNet-C (c) ImageNet-P

Fixing the train-test resolution discrepancy: FixEfficientNet
https://arxiv.org/abs/2003.08237

= NoisyStudent +

Fixing the train-test resolution discrepancy
https://arxiv.org/abs/1906.06423



RANK	METHOD	TOP 1 ACCURACY	TOP 5 ACCURACY	NUMBER OF PARAMS	EXTRA TRAINING DATA	PAPER TITLE	YEAR
1	**FixEfficientNet-L2**	88.5%	98.7%	480M	✓	Fixing the train-test resolution discrepancy: FixEfficientNet	2020
2	**NoisyStudent** (EfficientNet-L2)	88.4%	98.7%	480M	✓	Self-training with Noisy Student improves ImageNet classification	2020
3	**BiT-L** (ResNet)	87.8%			✓	Large Scale Learning of General Visual Representations for Transfer	2019
4	**NoisyStudent** (EfficientNet-L2)	87.4%	98.2%	480M	✓	Self-training with Noisy Student improves ImageNet classification	2019
5	**FixEfficientNet-B7**	87.1%	98.2%	66M	✓	Fixing the train-test resolution discrepancy: FixEfficientNet	2020
6	**NoisyStudent** (EfficientNet-B7)	86.9%	98.1%	66M	✓	Self-training with Noisy Student improves ImageNet classification	2020
7	**FixEfficientNet-B6**	86.7%	98.0%	43M	✓	Fixing the train-test resolution discrepancy: FixEfficientNet	2020
8	**FixResNeXt-101 32x48d**	86.4%	98.0%	829M	✓	Fixing the train-test resolution discrepancy	2019
9	**NoisyStudent** (EfficientNet-B6)	86.4%	97.9%	43M	✓	Self-training with Noisy Student improves ImageNet classification	2020
10	**FixEfficientNet-B5**	86.4%	97.9%	30M	✓	Fixing the train-test resolution discrepancy: FixEfficientNet	2020

Is 88.5% top1 or 98.7% top5 impressive?

- It is.
- ImageNet has 1000 classes
- Best reported human top5 accuracy: 94.9%
  - https://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/
- For comparison: Inception ResNet v2
  - 80.1% top1, 95.1% top5

ImageNet Large Scale Visual Recognition Challenge
Section 6.4 Human accuracy on large-scale image classification
https://arxiv.org/abs/1409.0575

Q&A