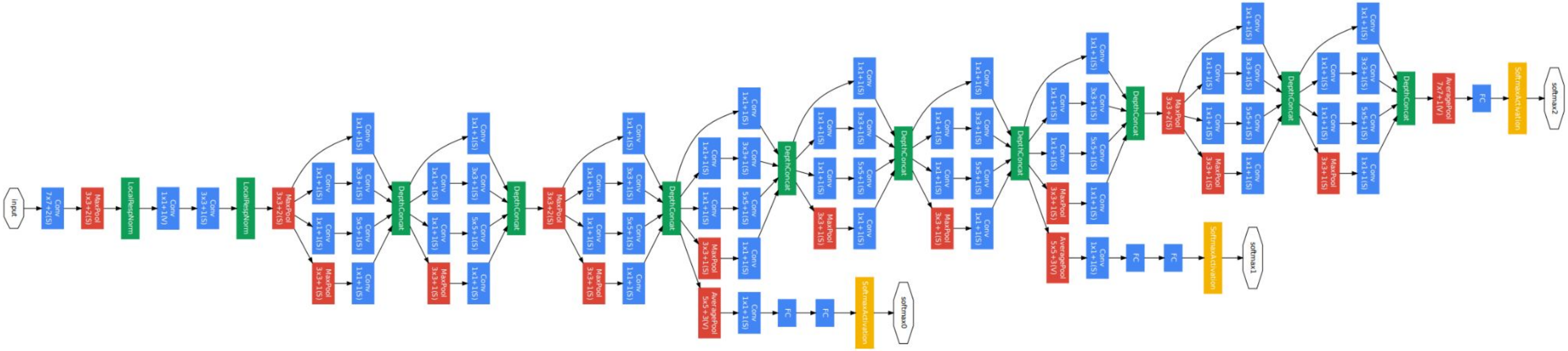
Learning Transferable Architectures for Scalable Image Recognition

Recent Trends in Automated Machine Learning
Technische Universität München

Ahmed Bahnasy
June 2nd, 2021

Motivation



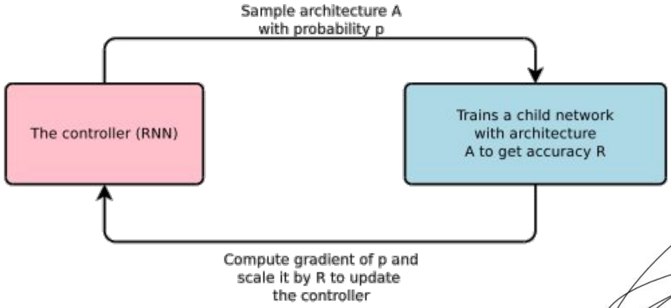
GoogleLeNet Architecture with all bells and whistles

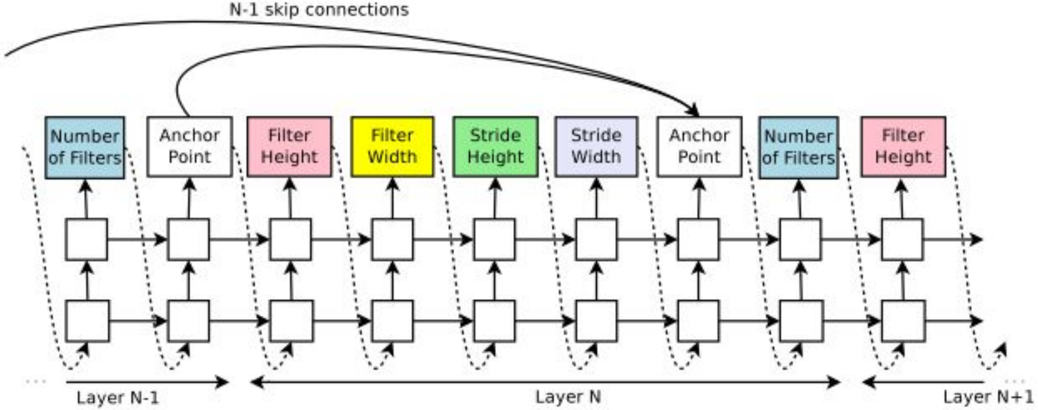
- CNN models requires significant architectural Engineering !
- Can we design an algorithm to design the architecture for us?!

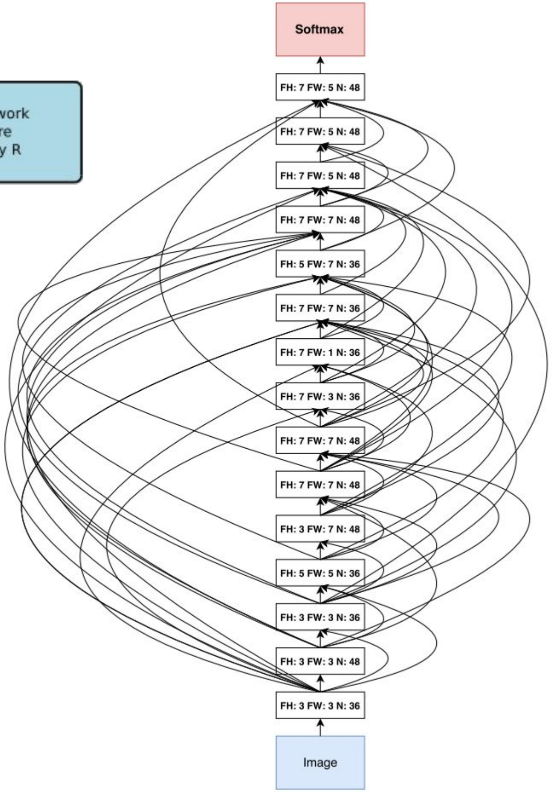
Source: https://arxiv.org/abs/1409.4842

NAS (2017) Revisited

- NAS Components
  - Search Space
  - Search Algorithms
  - Child Model Evolution







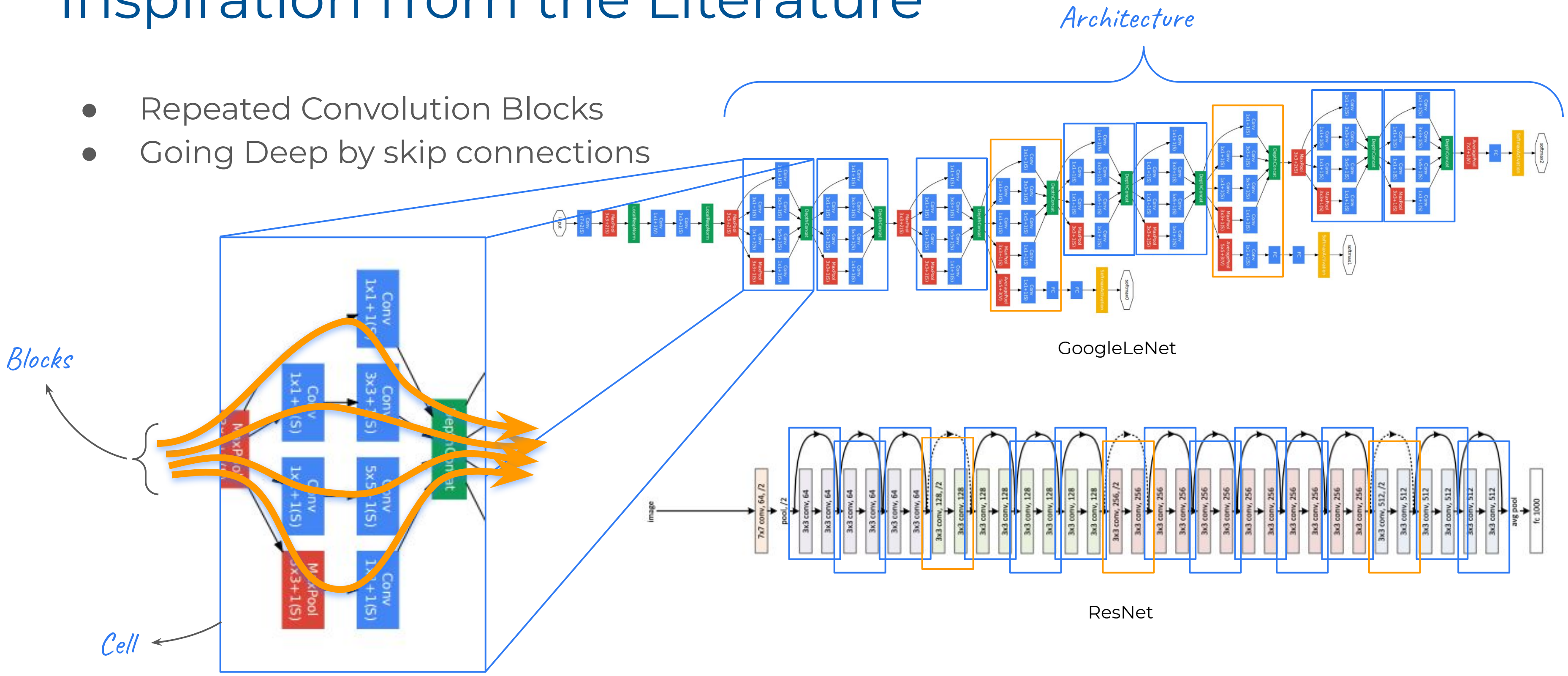
Neural Architecture Search with Reinforcement Learning [Barret & Quoc 2017]

NAS (2017) vs. NASNet (2018)

- NAS (2017) Limitations
  - Computationally expensive for small dataset
  - No transferability between datasets
- NASNet (2018)
  - Reduced Computation Cost
  - Transferable from small dataset to large dataset
  - Generic learned features for multiple Computer Vision tasks

Inspiration from the Literature

- Repeated Convolution Blocks
- Going Deep by skip connections



NASNet Search Space

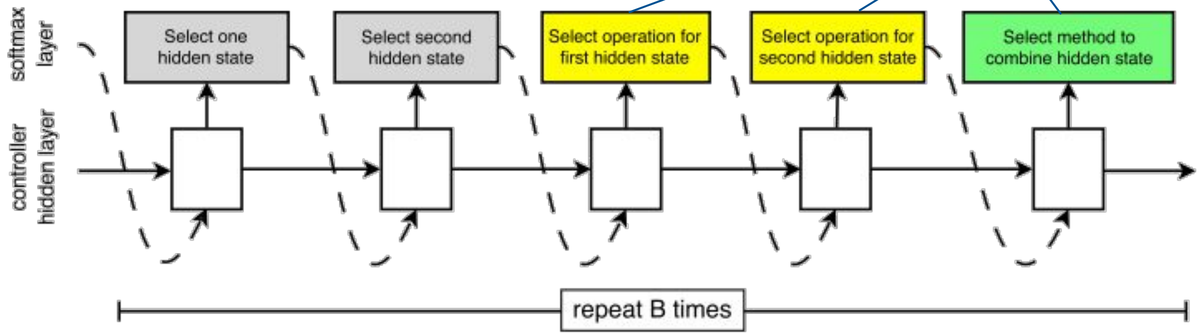
- Design a Cell, Repeat, Stack = Architecture
- Search Space outputs Cells **not** Architectures
- Architecture Complexity is **decoupled** from depth of the network and size of the input images

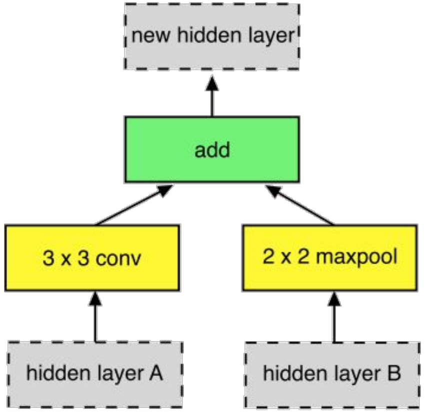
NASNet Search Space

Manually Chosen !

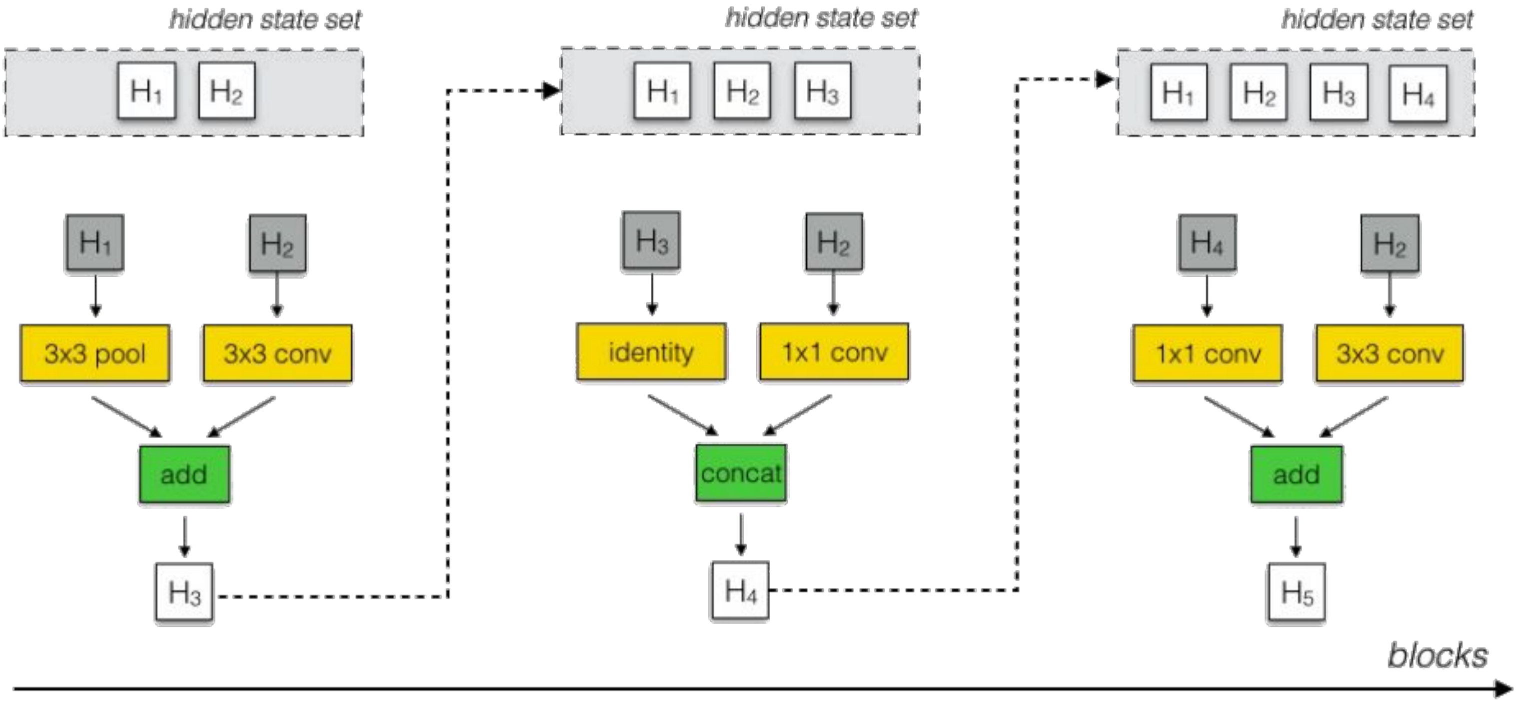
- Predictions of each cell are grouped into **B** blocks
- Each block has 5 prediction steps
- The combination can be **addition or concatenation**

- identity
- 1x7 then 7x1 convolution
- 3x3 average pooling
- 5x5 max pooling
- 1x1 convolution
- 3x3 depthwise-separable conv
- 7x7 depthwise-separable conv
- 1x3 then 3x1 convolution
- 3x3 dilated convolution
- 3x3 max pooling
- 7x7 max pooling
- 3x3 convolution
- 5x5 depthwise-seperable conv



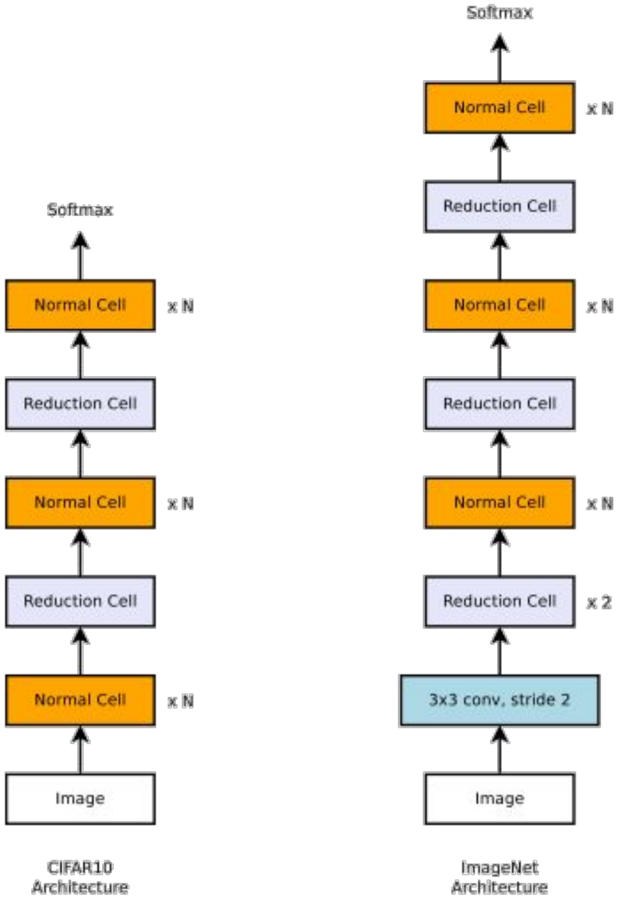


NASNet Search Space

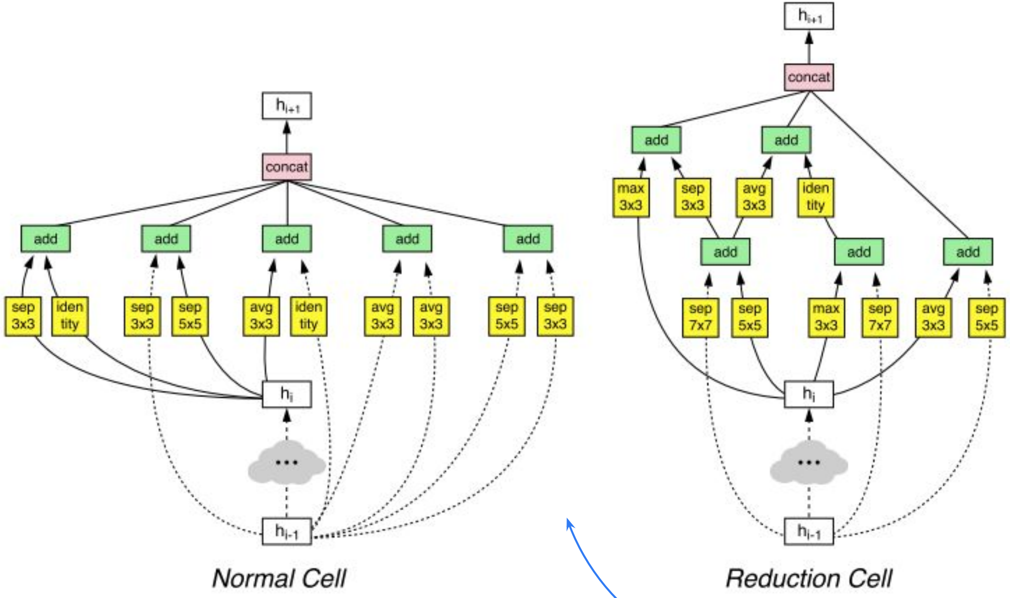


NASNet Architecture

- Composed of **repetitive** convolution cells
  - Normal Cell
  - Reduction Cell
- Free Parameters
  - # Block repetitions (**N**)
  - # initial convolutions



Experiments: CIFAR-10



model	depth	# params	error rate (%)
DenseNet ($L = 40, k = 12$) [26]	40	1.0M	5.24
DenseNet($L = 100, k = 12$) [26]	100	7.0M	4.10
DenseNet ($L = 100, k = 24$) [26]	100	27.2M	3.74
DenseNet-BC ($L = 100, k = 40$) [26]	190	25.6M	3.46
Shake-Shake 26 2x32d [18]	26	2.9M	3.55
Shake-Shake 26 2x96d [18]	26	26.2M	2.86
Shake-Shake 26 2x96d + cutout [12]	26	26.2M	2.56
NAS v3 [71]	39	7.1M	4.47
NAS v3 [71]	39	37.4M	3.65
NASNet-A (6 @ 768)	-	3.3M	3.41
NASNet-A (6 @ 768) + cutout	-	3.3M	2.65
NASNet-A (7 @ 2304)	-	27.6M	2.97
NASNet-A (7 @ 2304) + cutout	-	27.6M	2.40
NASNet-B (4 @ 1152)	-	2.6M	3.73
NASNet-C (4 @ 640)	-	3.1M	3.59

NASNet-A Architecture, best one !

Experiments: Computing Resources

- NAS (2017)
  - 800 GPUs
  - 28 Days
  - 22,400 GPU hours
  - Nvidia K40 GPUs
- NASNet (2018)
  - 500 GPUs
  - 4 Days
  - 2,000 GPU hours
  - NVidia P100s

Discounting the fact of different Hardware, it is estimated to be roughly 7x more efficient

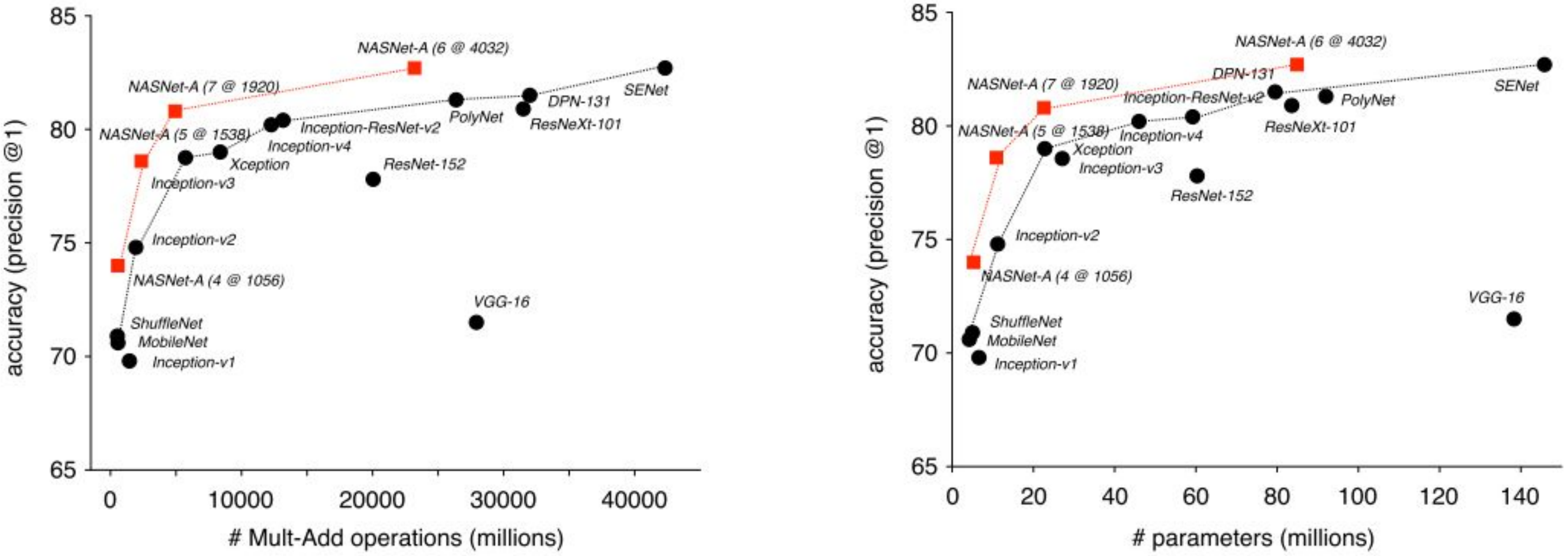
Experiments: ImageNet

Mobile Scale Networks

Model	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V1 [59]	6.6M	1,448 M	69.8 †	89.9
MobileNet-224 [24]	4.2 M	569 M	70.6	89.5
ShuffleNet (2x) [70]	∼ 5M	524 M	70.9	89.8
NASNet-A (4 @ 1056)	**5.3 M**	**564 M**	**74.0**	**91.6**
NASNet-B (4 @ 1536)	5.3M	488 M	72.8	91.3
NASNet-C (3 @ 960)	4.9M	558 M	72.5	91.0

Model	image size	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V2 [29]	224×224	11.2 M	1.94 B	74.8	92.2
NASNet-A (5 @ 1538)	**299×299**	**10.9 M**	**2.35 B**	**78.6**	**94.2**
Inception V3 [60]	299×299	23.8 M	5.72 B	78.8	94.4
Xception [9]	299×299	22.8 M	8.38 B	79.0	94.5
Inception ResNet V2 [58]	299×299	55.8 M	13.2 B	80.1	95.1
NASNet-A (7 @ 1920)	**299×299**	**22.6 M**	**4.93 B**	**80.8**	**95.3**
ResNeXt-101 (64 x 4d) [68]	320×320	83.6 M	31.5 B	80.9	95.6
PolyNet [69]	331×331	92 M	34.7 B	81.3	95.8
DPN-131 [8]	320×320	79.5 M	32.0 B	81.5	95.8
SENet [25]	**320×320**	**145.8 M**	**42.3 B**	**82.7**	**96.2**
NASNet-A (6 @ 4032)	**331×331**	**88.9 M**	**23.8 B**	**82.7**	**96.2**

Experiments: ImageNet Computational Demand



Experiments: COCO Dataset

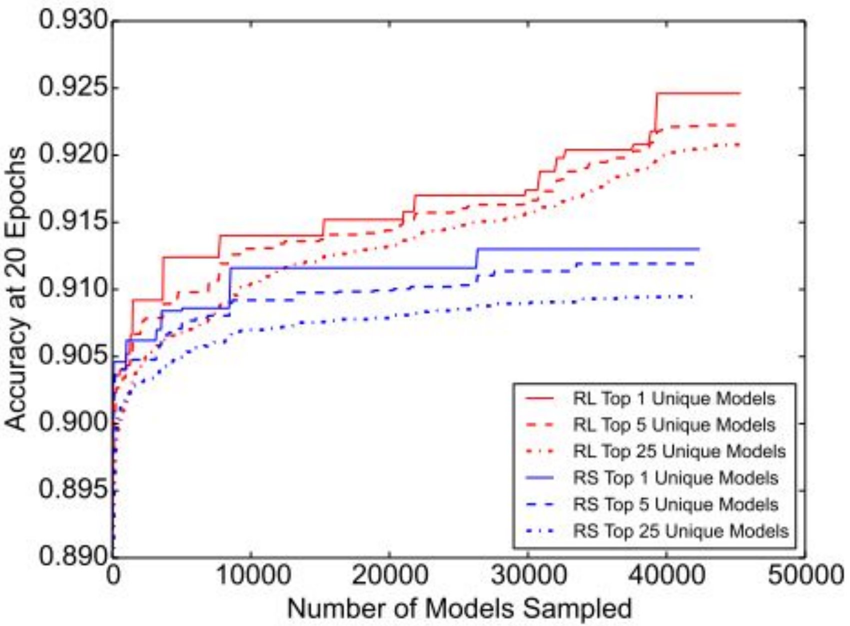
Object Detection Task

NASNet-A + RCNN Framework

Model	resolution	mAP (mini-val)	mAP (test-dev)
MobileNet-224 [24]	600 × 600	19.8%	-
ShuffleNet (2x) [70]	600 × 600	24.5%†	-
NASNet-A (4 @ 1056)	600 × 600	**29.6%**	-
ResNet-101-FPN [36]	800 (short side)	-	36.2%
Inception-ResNet-v2 (G-RMI) [28]	600 × 600	35.7%	35.6%
Inception-ResNet-v2 (TDM) [52]	600 × 1000	37.3%	36.8%
NASNet-A (6 @ 4032)	800 × 800	41.3%	40.7%
NASNet-A (6 @ 4032)	1200 × 1200	**43.2%**	**43.1%**
ResNet-101-FPN (RetinaNet) [37]	800 (short side)	-	39.1%

Experiments: Architecture search methods

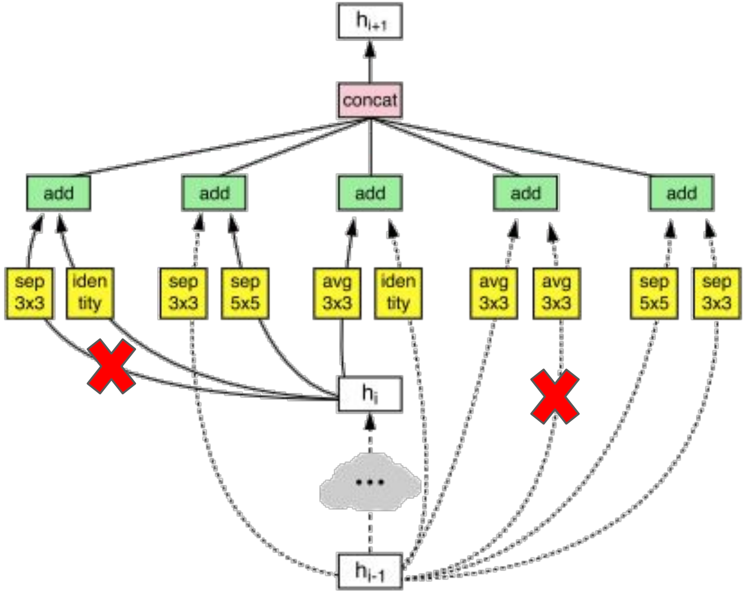
- Random Search
  - Strong Baseline !
- Reinforcement Learning
  - Entire range superior quality models
  - Best model differs ~ 1% accuracy
  - Better mean performance on the top-5 and top-25



Experiments: Scheduled DropPath

- DropPath
  - each path in the cell is stochastically dropped with some fixed probability during training
- Scheduled DropPath
  - each path in the cell is dropped out with a probability that is linearly increased over the course of training
- Observed through experiments

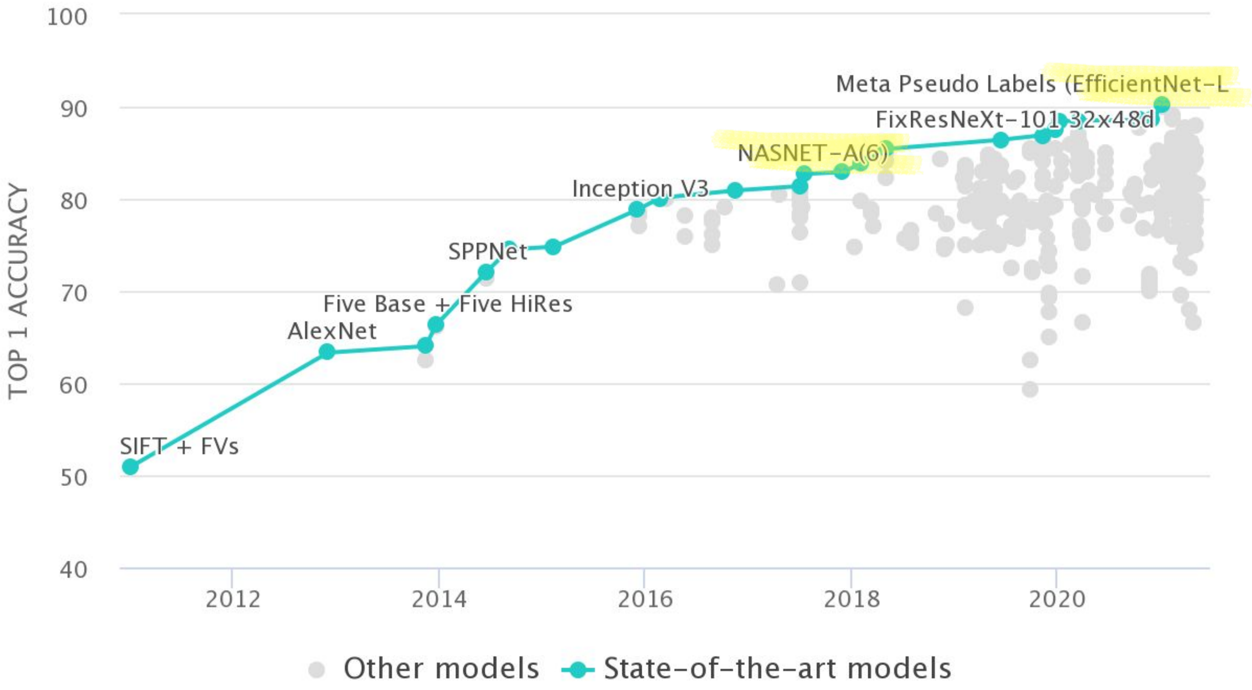
  No Further Explanation !



Normal Cell

The paper through the eyes of 2021

- EfficientNet-L2 90.2%
- ViT - 88.55%
- NASNet - 82.7%



https://paperswithcode.com/sota/image-classification-on-imagenet

Thank you :)

Questions ??

References

- Learning Transferable Architectures for Scalable Image Recognition
- Neural architecture search with reinforcement learning
- FractalNet: Ultra-Deep Neural Networks without Residuals
- EfficientNet: Rethinking model scaling for convolutional neural networks
- EfficientNetV2: Smaller Models and Faster Training