Advanced Deep Spatial Models

AI602: Recent Advances in Deep Learning Lecture 2

> Slide made by Jongheon Jeong and Sukmin Yun KAIST EE

- Assignment: 1 presentation + 1 report
- We will no longer receive emails regarding presentation topic assignment.
- Instead, we will move on to Google Sheets.
 - If you have not chosen your topic yet, please do so (due 3/12)
 - Update: you may now freely choose any topic by 3/12; Otherwise, you will be assigned to a random topic.
 - Once you have chosen the topic, you must choose the paper you want to present on (due one week before your presentation)
 - You must choose a paper published in 2019 2021
 - You cannot choose a paper chosen by another student (first-come-first-serve)
- Presentation schedules will be announced on **3/14**
- FAQ
 - Q. Is it enough to present on the paper's content only, or do we need to include our own experiment results?
 - A. You only have to present on the paper's content only.

Convolutional neural networks have been tremendously successful in practical applications;

Classification and retrieval [Krizhevsky et al., 2012]



Segmentation [Farabet et al., 2013]



Detection [Ren et al., 2015]



- Neural networks that use convolution in place of general matrix multiplication
 - Sharing parameters across multiple image locations
 - Translation equivariant (invariant with **pooling**) operation
- Specialized for processing data that has a known, grid-like topology
 - e.g. time-series data (1D grid), image data (2D grid)



*sources :

- https://www.cc.gatech.edu/~san37/post/dlhc-cnn/
- http://colah.github.io/posts/2014-07-Conv-Nets-Modular/

4

- Typically, designing a CNN model requires some effort
 - There are a lot of design choices: # layers, # filters, sizes of kernel, pooling, ...
 - It is costly to measure the performance of each model and choose the best one
- Example: LeNet for handwritten digits recognition [LeCun et al., 1998]



- However, LeNet is not enough to solve real-world problems in AI domain
 - CNNs are typically applied to extremely complicated domains, e.g. raw RGB images
 - We need to design a larger model to solve them adequately

- **Problem**: The larger the network, the more difficult it is to design
 - **1.** Optimization difficulty
 - When the training loss is degraded
 - Deeper networks are typically much harder to optimize
 - Related to gradient vanishing and exploding
 - 2. Generalization difficulty
 - The training is done well, but the testing error is degraded
 - Larger networks are more likely to over-fit, i.e., regularization is necessary
- Good architectures should be **scalable** that solves both of these problems



*sources : - He et al. "Deep residual learning for image recognition". CVPR 2016.

Algorithmic Intelligence Laboratory

https://upload.wikimedia.org/wikipedia/commons/thumb/6/68/Overfitted_Data.png/300px-Overfitted_Data.png

1. Evolution of CNN Architectures

- AlexNet and ZFNet
- VGGNet and GoogLeNet

2. Architectural Innovations in CNNs

- Batch normalization and ResNet
- ResNet oriented architectures
- Toward automation of network design
- Principle of network scaling

3. Explorations in CNN Modules

- Dilated and Deformable convolution
- Attention module in CNNs

4. Observational Study on CNN Architectures

- ResNets behave like ensembles of relatively shallow nets
- Visualizing the loss landscape of neural nets
- Essentially no barriers in neural network energy landscape

1. Evolution of CNN Architectures

- AlexNet and ZFNet
- VGGNet and GoogLeNet
- 2. Architectural Innovations in CNNs
 - Batch normalization and ResNet
 - ResNet oriented architectures
 - Toward automation of network design
 - Principle of network scaling
- 3. Explorations in CNN Modules
 - Dilated and Deformable convolution
 - Attention module in CNNs

4. Observational Study on CNN Architectures

- ResNets behave like ensembles of relatively shallow nets
- Visualizing the loss landscape of neural nets
- Essentially no barriers in neural network energy landscape

• ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- ImageNet dataset: a large database of visual objects
 - ~14M labeled images, 20K classes
 - Human labels via Amazon MTurk
- Classification: 1,281,167 images for training / 1,000 categories
- Annually ran from 2010 to 2017, and now hosted by Kaggle
- For details, see [Russakovsky et al., 2015]



ILSVRC contributed greatly to development of CNN architectures



• ILSVRC contributed greatly to development of CNN architectures



- The first winner to use CNN in ILSVRC, with an astounding improvement
 - Top-5 error is largely improved: $25.8\% \rightarrow 15.3\%$
 - The 2nd best entry at that time was **26.2%**
- 8-layer CNN (5 Conv + 3 FC)
- Utilized 2 GPUs (GTX-580 \times 2) for training the network
 - Split a single network into 2 parts to distribute them into each GPU



Algorithmic Intelligence Laboratory

*source : Krizhevsky et al. "Imagenet classification with deep convolutional neural networks". NIPS 2012 12

- Local response normalization layers (LRN)
 - Detects high-frequency features with a big neuron response
 - Dampens responses that are uniformly large in a local neighborhood
- Useful when using neurons with unbounded activations (e.g. ReLU)

$$b_{x,y}^{i} = a_{x,y}^{i} / \left(k + \alpha \sum_{j=\max(0,i-\frac{n}{2})}^{\min(N-1,i+\frac{n}{2})} (a_{x,y}^{j})^{2} \right)^{\beta}$$



Evolution of CNN Architectures: ZFNet [Zeiler et al., 2014]

- A simple variant of AlexNet, placing the 3^{rd} in ILSVRC'13 (15.3% \rightarrow **13.5%**)
 - Smaller kernel at input: $11 \times 11 \rightarrow 7 \times 7$
 - Smaller stride at input: $4 \rightarrow 2$
 - The # of hidden filters are doubled
- Lessons:
 - 1. Design principle: Use smaller kernel, and smaller stride
 - 2. CNN architectures can be very sensitive on hyperparameters



• ILSVRC contributed greatly to development of CNN architectures



Networks were getting deeper

- AlexNet: 8 layers
- VGGNet: 19 layers
- GoogleNet: 24 layers
- Both focused on parameter efficiency of each block
 - Mainly to allow larger networks computable at that time



16

- The 2nd place in ILSVRC'14 (11.7% \rightarrow **7.33%**)
- Designed using only 3×3 kernels for convolutions
- Lesson: Stacking multiple 3 × 3 is advantageous than using other kernels
- **Example**: ((3×3)×3) v.s. (7×7)
 - Essentially, they get the same receptive field
 - ((3×3)×3) have less # parameters

•
$$3 \times (C \times ((3 \times 3) \times C)) = 27C^2$$

- $C \times ((7 \times 7) \times C) = 49C^2$
- ((3×3)×3) gives more non-linearities





Evolution of CNN Architectures: GoogleNet [Szegedy et al., 2015]

- The winner of ILSVRC'14 (11.7% → 6.66%)
- Achieved 12× fewer parameters than AlexNet

Inception module

- Multiple operation paths with different receptive fields
- Each of the outputs are **concatenated** in filter-wise
- Capturing sparse patterns in a stack of features



۲ 001+070 001+070 001+17

Evolution of CNN Architectures: GoogleNet [Szegedy et al., 2015]

- The winner of ILSVRC'14 (11.7% \rightarrow 6.66%)
- Achieved 12× fewer parameters than AlexNet
- Use of 1×1 convolutions
 - Naïve inceptions can be too expensive to scale up
 - Dimension reduction before expensive convolutions
 - They also gives more non-linearities



(b) Inception module with dimensionality reduction

-SEC +274 HEIG HEIG +272 HEIG +272 HEIG +272 (V)1+171 (V)1+171 4900 (S1+070 490) 001+070 001+070 001+17

Evolution of CNN Architectures: GoogleNet [Szegedy et al., 2015]

- The winner of ILSVRC'14 (11.7% \rightarrow 6.66%)
- Achieved 12× fewer parameters than AlexNet

• *cf.* $\mathbf{1} \times \mathbf{1}$ convolutions

- Linear transformation done in pixel-wise
- Can be represented by a matrix
- Useful for changing # channels efficiently





20

*sources :

- Szegedy et al., "Going deeper with convolutions". CVPR 2015

Algorithmic Intelligence Laboratory

- Lana Lazebnik, "Convolutional Neural Network Architectures: from LeNet to ResNet".

ILSVRC contributed greatly to development of CNN architectures



- **1. Evolution of CNN Architectures**
 - AlexNet and ZFNet
 - VGGNet and GoogLeNet

2. Architectural Innovations in CNNs

- Batch normalization and ResNet
- ResNet oriented architectures
- Toward automation of network design
- Principle of network scaling
- 3. Explorations in CNN Modules
 - Dilated and Deformable convolution
 - Attention module in CNNs
- 4. Observational Study on CNN Architectures
 - ResNets behave like ensembles of relatively shallow nets
 - Visualizing the loss landscape of neural nets
 - Essentially no barriers in neural network energy landscape

- Training a deep network well had been a delicate task
 - It requires a careful initialization, with adequately low learning rate
 - Gradient vanishing: networks containing saturating non-linearity
- Ioffe et al. (2015): Such difficulties are come from internal covariate shift
- Motivation: "The cup game analogy"



- Similar problem happens during training of deep neural networks
- Updates in early layers may shift the inputs of later layers too much

*sources :

- Ioffe et al., "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". ICML 2015
 - http://pages.cs.wisc.edu/~shavlik/cs638/lectureNotes/Batch_Normalization.pptx

Architectural Innovations in CNNs: Batch normalization [Ioffe et al., 2015]

- **Batch normalization** (BN) accelerates neural network training by eliminating internal covariate shift inside the network
- Idea: A normalization layer that behaves differently in training and testing



- 1. During training, input distribution of y only depends on γ and β
 - Training mini-batches are always normalized into mean 0, variance 1
- 2. There is some gap between $\mu_{\mathcal{B}}$ and $\mathbb{E}[\mu_{\mathcal{B}}]$ ($\sigma_{\mathcal{B}}^2$, resp.)
 - Noise injection effect for each mini-batch \Rightarrow Regularization effect

- Batch normalization (BN) accelerates neural network training by eliminating internal covariate shift inside the network
 - BN allows much higher learning rates, i.e. faster training
 - BN stabilizes gradient vanishing on saturating non-linearities
 - BN also has its own **regularization effect**, so that it allows to reduce weight decay, and to remove dropout layers
- BN makes GoogLeNet much easier to train with great improvements

Model	Resolution	Crops	Models	Top-1 error	Top-5 error
GoogLeNet ensemble	224	144	7	-	6.67%
Deep Image low-res	256	-	1	-	7.96%
Deep Image high-res	512	-	1	24.88	7.42%
Deep Image ensemble	variable	-	-	-	5.98%
BN-Inception single crop	224	1	1	25.2%	7.82%
BN-Inception multicrop	224	144	1	21.99%	5.82%
BN-Inception ensemble	224	144	6	20.1%	4.9%*

Next, ResNet

*source : loffe et al., "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". ICML 2015 25

- The winner of ILSVRC'15 (6.66% → 3.57%)
- ResNet is the first architecture succeeded to train >100-layer networks
 - Prior works could until ~30 layers, but failed for the larger nets

What was the problem?

- 56-layer net gets higher training error than 20-layers network
- Deeper networks are much harder to optimize even if we use BNs
- It's not due to overfitting, but optimization difficulty
 - Quiz: Why is that?





- The winner of ILSVRC'15 (6.66% → 3.57%)
- ResNet is the first architecture succeeded to train >100-layer networks
 - Prior works could until ~30 layers, but failed for the larger nets

What was the problem?

- It's not due to overfitting, but optimization difficulty
 - Quiz: Why is that?
- If the 56-layer model optimized well, then it **must be better** than the 20-layer
 - There is a trivial solution for the 36-layer: identity





Architectural Innovations in CNNs: ResNet [He et al., 2016a]

- Motivation: A non-linear layer may struggle to represent an identity function
 - Due to its internal non-linearities, e.g. ReLU
 - This may cause the optimization difficulty on large networks
- Idea: Reparametrize each layer to make them easy to represent an *identity*
 - When all the weights are set to zero, the layer represents an identity



• Plain nets v.s. ResNets



• Deeper ResNets can be trained without any difficulty





34-laver plain

34-layer residual

*sources :

- He et al., "Deep residual learning for image recognition". CVPR 2016
- He, Kaiming, "Deep Residual Networks: Deep Learning Gets Way Deeper." 2016. 29

Architectural Innovations in CNNs: ResNet [He et al., 2016a]

- Identity connection resolved a major difficulty on optimizing large networks
- **Revolution of depth**: Training >100-layer network without difficulty
 - Later, ResNet is revised to allow to train up to >1000 layers [He et al., 2016b]
- ResNet also shows good generalization ability as well



ImageNet Classification top-5 error (%)

*sources :

- He et al., "Deep residual learning for image recognition". CVPR 2016
 - Kaiming He, "Deep Residual Networks: Deep Learning Gets Way Deeper." 2016.

Algorithmic Intelligence Laboratory

He et al. "Identity mappings in deep residual networks.", ECCV 2016

• Comparisons on ImageNet for a single model of popular CNNs



Various architectures now are based on ResNet

- ResNet with stochastic depth [Huang et al., 2016]
- Wide ResNet [Zagoruyko et al., 2016]
- ResNet in ResNet [Targ et al., 2016]
- ResNeXt [Xie et al., 2016]
- PyramidNet [Han et al., 2016]
- Inception-v4 [Szegedy et al., 2017]
- DenseNet [Huang et al., 2017]
- Dual Path Network [Chen et al., 2017]



- **Transition of design paradigm:** Optimization ⇒ Generalization
 - People are now less concerned about optimization problems in a model
 - Instead, they now focus more on its generalization ability
 - "How well does an architecture generalize as its scale grows?"

- Wide Residual Networks [Zagoruyko et al., 2016]
 - Residuals can also work to enlarge the width, not only its depth
 - Residual blocks with $\times k$ wider filters
 - Increasing width instead of depth can be more computationally efficient
 - GPUs are much better on handling "wide-but-shallow" than "thin-but-deep"
 - WRN-50 outperforms ResNet-152
- Deep Networks with Stochastic Depth [Huang et al., 2016]
 - Randomly drop a subset of layers during training
 - Bypassing via identity connections
 - Reduces gradient vanishing, and training time as well





Architectural Innovations in CNNs: ResNet oriented architectures

- ResNeXt [Xie et al., 2016]
 - Aggregating multiple parallel paths inside a residual block ("cardinality")
 - Increasing cardinality is more effective than going deeper or wider
- DenseNet [Huang et al. 2017]
 - Passing all the previous representation directly via concatenation of features
 - Strengthens feature propagation and feature reuse







- **ResNeXt** [Xie et al., 2016]
 - Aggregating multiple parallel paths inside a residual block ("cardinality")
 - Increasing cardinality is more effective than going deeper or wider
- DenseNet [Huang et al. 2017]
 - Passing all the previous representation directly via concatenation of features
 - Strengthens feature propagation and feature reuse



- Although the CNN architecture has evolved greatly, our **design principles are still relying on heuristics**
 - Smaller kernel and smaller stride, increase cardinality instead of width ...
- Recently, there have been works on automatically finding a structure which can outperform existing human-crafted architectures
 - 1. Search space: Naïvely searching every model is nearly impossible
 - 2. Searching algorithm: Evaluating each model is very costly, and black-boxed



A sample architecture found in [Brock et al., 2018]
- **Designing a good search space** is important in architecture searching
- **NASNet** reduces the search space by incorporating our design principles
- Motivation: modern architectures are built simply: a repeated modules
 - Try not to search the whole model, but only cells modules
 - Normal cell and Reduction cell (cell w/ stride 2)



- Designing a good search space is important in architecture searching
- **NASNet** reduces the search space by incorporating our design principles
- Each cell consists of *B* blocks
- Each block is determined by **selecting methods**
 - 1. Select two hidden states from h_i , h_{i-1} or of existing block
 - 2. Select methods to process for each of the selected states
 - 3. Select a method to combine the two states
 - (1) element-wise addition or (2) concatenation





- Designing a good search space is important in architecture searching
- **NASNet** reduces the search space by incorporating our design principles
- Each cell consists of *B* blocks
 - **Example**: *B* = 4



Algorithmic Intelligence Laboratory

*source : Zoph et al., "Learning Transferable Architectures for Scalable Image Recognition", CVPR 2018 39

- **Designing a good search space** is important in architecture searching
- **NASNet** reduces the search space by incorporating our design principles
- Set of methods to be selected based on their prevalence in the CNN literature
 - 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

- Any searching methods can be used
 - Random search [Bergstra et al., 2012] could also work
 - RL-based search [Zoph et al., 2016] is mainly used in this paper

- The pool of workers consisted of **500 GPUs**, processing over 4 days
- All architecture searches are performed on CIFAR-10
 - NASNet-A: State-of-the-art error rates could be achieved
 - NASNet-B/C: Extremely parameter-efficient models were also found

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 [70]	39	7.1M	4.47
NAS v3 [70]	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

- The pool of workers consisted of 500 GPUs, processing over 4 days
- All architecture searches are performed on CIFAR-10
 - NASNet-A: State-of-the-art error rates could be achieved
 - NASNet-B/C: Extremely parameter-efficient models were also found



*source : Zoph et al., "Learning Transferable Architectures for Scalable Image Recognition", CVPR 2018 42

- The pool of workers consisted of 500 GPUs, processing over 4 days
- All architecture searches are performed on CIFAR-10
- Cells found in CIFAR-10 could also transferred well into ImageNet

Model	image size	# parameters	Mult-Adds	Top 1 Acc. (%)	Top 5 Acc. (%)
Inception V2 [29] NASNet A (5 @ 1538)	224×224	11.2 M	1.94 B	74.8 78 6	92.2 94.2
NASNEL-A (5 @ 1538)	299×299	10.9 141	2.33 D	/0.0	74.2
Inception V3 [59]	299×299	23.8 M	5.72 B	78.0	93.9
Xception [9]	299×299	22.8 M	8.38 B	79.0	94.5
Inception ResNet V2 [57]	299×299	55.8 M	13.2 B	80.4	95.3
NASNet-A (7 @ 1920)	299×299	22.6 M	4.93 B	80.8	95.3
ResNeXt-101 (64 x 4d) [67]	320×320	83.6 M	31.5 B	80.9	95.6
PolyNet [68]	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

- The pool of workers consisted of 500 GPUs, processing over 4 days
- All architecture searches are performed on CIFAR-10
- Cells found in CIFAR-10 could also transferred well into ImageNet



- Architecture searching is still an active research area
 - AmoebaNet [Real et al., 2018]
 - Efficient-NAS (ENAS) [Pham et al., 2018]
 - NAONet [Luo et al., 2018]

Model	Error(%)	#params	GPU Days
DenseNet-BC [19]	3.46	25.6M	/
ResNeXt-29 [43]	3.58	68.1M	/
NASNet-A [48]	3.41	3.3M	2000
NASNet-B [48]	3.73	2.6M	2000
NASNet-C [48]	3.59	3.1M	2000
Hier-EA [28]	3.75	15.7M	300
AmoebaNet-A [38]	3.34	3.2M	3150
AmoebaNet-B [38]	3.37	2.8M	3150
AmoebaNet-B [38]	3.04	13.7M	3150
AmoebaNet-B [38]	2.98	34.9M	3150
AmoebaNet-B + Cutout [38]	2.13	34.9M	3150
ENAS [37]	3.54	4.6M	0.45
PNAS [27]	3.41	3.2M	225
DARTS + Cutout [29]	2.83	4.6M	4
NAONet	3.18	10.6M	200
NAONet	2.98	28.6M	200
NAONet + Cutout	2.07	128M	200
NAONet-WS	3.53	3.7M	0.4

Next, EfficientNet

Architectural Innovations in CNNs: Principle of Network Scaling

- Although **Scaling up** CNNs is widely used to achieve better generalization, the process of scaling has never been understood
 - The common way is scaling model depth, width, and image resolution
- Question: Is there a principled scaling method for better accuracy and efficiency?



Algorithmic Intelligence Laboratory

*source : Tan et al., "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", ICML 2019 46

- The state-of-the-art ILSVRC classification in 2019 (top-5 error rate 2.9%)
- EfficientNet uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients (called "compound scaling")
- **Motivation**: There exists certain relationship between network width, depth and image resolution
 - Scaling single dimension has a limitation
 - Gain diminishes for bigger models.



• Scaling all together with a fixed ratio

- **Compound scaling:** Scaling all together with a fixed ratio ϕ in a principled way
 - Depth $d = \alpha^{\phi}$, $\alpha \ge 1$
 - Width $w = \beta^{\phi}, \beta \ge 1$

Algorithmic Intelligence Laboratory

- Resolution $r = \gamma^{\phi}, \gamma \ge 1$
- Finding α , β , γ under compound constraint $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$
 - Why? Such scaling approximately increases total FLOPS by $(\alpha \cdot \beta^2 \cdot \gamma^2)^{\phi} \approx 2^{\phi}$



*source : Tan et al., "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", ICML 2019 48

5

- From EfficientNet-B0 to B7
 - EfficientNet-BO: Baseline model with $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$
 - EfficientNet-B1 to B7: Scaling up EfficientNet-B0 with different ϕ

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
EfficientNet-B0	77.1%	93.3%	5.3M	1x	0.39B	1x
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	79.1%	94.4%	7.8M	1x	0.70B	1x
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11 B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	80.1%	94.9%	9.2M	1x	1.0B	1x
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	81.6%	95.7%	12M	1x	1.8B	1x
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	82.9%	96.4%	19M	1x	4.2B	1x
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	83.6%	96.7%	30M	1x	9.9B	1x
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	84.0%	96.8%	43M	1x	19B	1x
EfficientNet-B7	84.3%	97.0%	66M	1x	37B	1x
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

Algorithmic Intelligence Laboratory

*source : Tan et al., "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", ICML 2019 49

- From EfficientNet-B0 to B7
 - EfficientNet-BO: Baseline model with $\alpha = 1.2$, $\beta = 1.1$, $\gamma = 1.15$
 - EfficientNet-B1 to B7: Scaling up EfficientNet-B0 with different ϕ



EfficientNet-B7 achieves new state-of-the-art 84.3% top-1 accuracy but being 1.3x smaller than NASNet-A.

EfficientNet-B1 is 7.6x smaller and 5.7x faster than ResNet-152

Next, Dilated Conv

- **1. Evolution of CNN Architectures**
 - AlexNet and ZFNet
 - VGGNet and GoogLeNet
- 2. Architectural Innovations in CNNs
 - Batch normalization and ResNet
 - Beyond ResNet
 - Toward automation of network design
 - Principle of network scaling

3. Explorations in CNN Modules

- Dilated and Deformable convolution
- Attention module in CNNs
- 4. Observational Study on CNN Architectures
 - ResNets behave like ensembles of relatively shallow nets
 - Visualizing the loss landscape of neural nets
 - Essentially no barriers in neural network energy landscape

- Objects in real-world often contain sophisticated spatial information
 - Multiple scales
 - Irregular shapes
- Drawbacks: geometric transformations are assumed fixed and known
 - Different size and shape of kernels may be required
 - But, regular kernels have fixed-size and shape



- Objects in real-world often contain sophisticated spatial information
 - Multiple scales
 - Irregular shapes
- Drawbacks: geometric transformations are assumed fixed and known
 - Different size and shape of kernels may be required
 - But, regular kernels have fixed-size and shape



- Motivation: Images in real-world usually contain multi-scale objects
 - Regular convolution has a fixed-size of field of view
 - Different size of kernels are required for multi-scale objects
 - But, large-size of kernels may increase computational costs
- **Dilated convolution:** Filling with **zero values** inside of large-size of kernels for efficient computation
 - It can enlarge field-of-view to incorporate multi-scale context



Explorations in CNN Modules: Dilated Convolution [Chen et al., 2017]

- Motivation: Images in real-world usually contain multi-scale objects
 - Regular convolution has a fixed-size of field of view
 - Different size of kernels are required for multi-scale objects
 - But, large-size of kernels may increase computational costs
- Example: Dilated convolution in semantic segmentation



- Motivation: Shape of objects in the real world are usually irregular
 - Different shape of kernels are required for irregular objects
 - Regular convolution has a fixed-shape of kernel
- **Deformable convolution:** Learning sampling location of kernels to capture irregular shape of objects
 - Adding offset field to generate irregular sampling locations



Different types of sampling locations

- Motivation: Shape of objects in the real world are usually irregular
 - Different shape of kernels are required for irregular objects
 - Regular convolution has a fixed-shape of kernel
- **Deformable convolution:** Learning sampling location of kernels to capture irregular shape of objects
 - Adding offset field to generate irregular sampling locations



Regular convolution

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n)$$

Deformable convolution

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta \mathbf{p}_n)$$

where Δp_n is generated by a sibling branch of regular convolution (offset field)

Algorithmic Intelligence Laboratory

*source : https://jifengdai.org/slides/Deformable_Convolutional_Networks_Oral.pdf 57

Explorations in CNN Modules: Deformable Convolution [Dai et al., 2017]

- Motivation: Shape of objects in the real world are usually irregular
 - Different shape of kernels are required for irregular objects
 - Regular convolution has a fixed-shape of kernel
- **Deformable convolution:** Learning sampling location of kernels to capture irregular shape of objects
 - Adding offset field to generate irregular sampling locations





(b) deformable convolution

Algorithmic Intelligence Laboratory

*source : https://jifengdai.org/slides/Deformable_Convolutional_Networks_Oral.pdf 58

Explorations in CNN Modules: Deformable Convolution [Dai et al., 2017]

- Motivation: Shape of objects in the real world are usually irregular
 - Different shape of kernels are required for irregular objects
 - Regular convolution has a fixed-shape of kernel
- Learned offsets in the deformable convolution layers are highly adaptive to the image content
 - Different size and shape of kernels for multiple objects



Visualizations of sampling locations

Algorithmic Intelligence Laboratory

*source: Dai et al., "Deformable Convolutional Networks", ICCV, 2017 59

Next, SENet

- Motivation: The deeper the model, the more feature maps are generated
 - Many of them might be important for classification task
 - Others might redundant or less important
- Squeeze and Excitation Network [He et al., 2018]
 - It selectively emphasizes informative feature maps and suppress less useful ones via global information in two steps
 - Squeeze step: obtaining global information by shrinking feature maps
 - Global average pooling
 - Excitation step: recalibrating weights of features by learning channel-wise weights
 - MLP of two fully-connected layers



Algorithmic Intelligence Laboratory

*source: He et al., "Squeeze-and-Excitation Networks", CVPR, 2018 60

Attention Module in CNNs: Squeeze and Excitation Module [He et al., 2018]

- Motivation: The deeper the model, the more feature maps are generated
 - Many of them might be important for classification task
 - Others might redundant or less important
- SE block integrates to Inception and ResNet module
 - SENet ranked first in the ILSVRC'17 (2.99% \rightarrow 2.25%)



*source: He et al., "Squeeze-and-Excitation Networks", CVPR, 2018 61

Attention Module in CNNs: Squeeze and Excitation Module [He et al., 2018]

- Motivation: The deeper the model, the more feature maps are generated
 - Many of them might be important for classification task
 - Others might redundant or less important
- SE block integrates to Inception and ResNet module
 - SENet ranked first in the ILSVRC'17 (2.99% \rightarrow 2.25%)

	orig	inal	re-implementation			SENet		
	top-1 err.	top-5 err.	top-1 err.	top-5 err.	GFLOPs	top-1 err.	top-5 err.	GFLOPs
ResNet-50 [13]	24.7	7.8	24.80	7.48	3.86	$23.29_{(1.51)}$	$6.62_{(0.86)}$	3.87
ResNet-101 [13]	23.6	7.1	23.17	6.52	7.58	$22.38_{(0.79)}$	$6.07_{(0.45)}$	7.60
ResNet-152 [13]	23.0	6.7	22.42	6.34	11.30	$21.57_{(0.85)}$	$5.73_{(0.61)}$	11.32
ResNeXt-50 [19]	22.2	-	22.11	5.90	4.24	$21.10_{(1.01)}$	$5.49_{(0.41)}$	4.25
ResNeXt-101 [19]	21.2	5.6	21.18	5.57	7.99	$20.70_{(0.48)}$	$5.01_{(0.56)}$	8.00
VGG-16 [11]	-	-	27.02	8.81	15.47	$25.22_{(1.80)}$	$7.70_{(1.11)}$	15.48
BN-Inception [6]	25.2	7.82	25.38	7.89	2.03	$24.23_{(1.15)}$	$7.14_{(0.75)}$	2.04
Inception-ResNet-v2 [21]	19.9^{\dagger}	4.9^{\dagger}	20.37	5.21	11.75	$19.80_{(0.57)}$	$4.79_{(0.42)}$	11.76

Next, Convolutional Block Attention Module

*source: He et al., "Squeeze-and-Excitation Networks", CVPR, 2018 62

Attention Module in CNNs: Convolutional Block Attention Module [Woo et al., 2018]

- Motivation: SENet only considers the contribution of feature maps
 - It ignores the spatial locality of the object in image
 - The spatial location of the object has a vital role in understanding image
- Convolutional Block Attention Module (CBAM) [He et al., 2018]
 - Learning 'what' and 'where' to attend in the channel and spatial axes respectively
 - Channel and Spatial attention modules



- Motivation: SENet only considers the contribution of feature maps
 - It ignores the spatial locality of the object in image
 - The spatial location of the object has a vital role in understanding image
- Channel attention module: It helps "what" to focus
 - Both average-pooling and max-pooling are important
 - **Max-pooling** provides the information of distinctive object features
 - Both pooled features share a MLP with two fully-connected layers



 $\mathbf{M_c}(\mathbf{F}) = \sigma(MLP(AvgPool(\mathbf{F})) + MLP(MaxPool(\mathbf{F})))$

Algorithmic Intelligence Laboratory

*source: Woo et al., "CBAM: Convolutional block attention module", ECCV, 2018 64

Attention Module in CNNs: Convolutional Block Attention Module [Woo et al., 2018]

- Motivation: SENet only considers the contribution of feature maps
 - It ignores the spatial locality of the object in image
 - The spatial location of the object has a vital role in understanding image
- Spatial attention module: It helps "where" to focus
 - Again, Both average-pooling and max-pooling are important
 - It aggregates channel information of feature maps by using two pooling operations
 - Capturing **spatial locality** via convolution



- Motivation: SENet only considers the contribution of feature maps
 - It ignores the spatial locality of the object in image
 - The spatial location of the object has a vital role in understanding image
- **CBAM** module integrated with ResNet outperforms SE module



Architecture	Param.	GFLOPs	Top-1 Error (%)	Top-5 Error (%)
ResNet18 [5]	11.69M	1.814	29.60	10.55
ResNet18 5 + SE [28]	11.78M	1.814	29.41	10.22
ResNet18 [5] + CBAM	11.78M	1.815	29.27	10.09
ResNet34 [5]	21.80M	3.664	26.69	8.60
$\operatorname{ResNet34}[5] + \operatorname{SE}[28]$	21.96M	3.664	26.13	8.35
$\operatorname{ResNet34}[5] + \operatorname{CBAM}$	21.96M	3.665	25.99	8.24
ResNet50 [5]	25.56M	3.858	24.56	7.50
ResNet50[5] + SE[28]	28.09M	3.860	23.14	6.70
ResNet50 [5] + CBAM	28.09M	3.864	22.66	6.31
ResNet101 [5]	44.55M	7.570	23.38	6.88
ResNet101 $[5] + SE [28]$	49.33M	7.575	22.35	6.19
m ResNet101~[5] + CBAM	49.33M	7.581	21.51	5.69
WideResNet18 [6] (widen=1.5)	25.88M	3.866	26.85	8.88
WideResNet18 [6] (widen= 1.5) + SE [28]	26.07M	3.867	26.21	8.47
WideResNet18 [6] (widen= 1.5) + CBAM	26.08M	3.868	26.10	8.43
WideResNet18 [6] (widen=2.0)	45.62M	6.696	25.63	8.20
WideResNet18 [6] (widen= 2.0) + SE [28]	45.97M	6.696	24.93	7.65
WideResNet18 [6] (widen=2.0) + CBAM	45.97M	6.697	24.84	7.63
ResNeXt50 [7] (32x4d)	25.03M	3.768	22.85	6.48
ResNeXt50 [7] (32x4d) + SE [28]	27.56M	3.771	21.91	6.04
${ m ResNeXt50}$ [7] (32x4d) $+$ CBAM	27.56M	3.774	21.92	5.91
ResNeXt101 [7] (32x4d)	44.18M	7.508	21.54	5.75
ResNeXt101 [7] (32x4d) + SE [28]	48.96M	7.512	21.17	5.66
$\operatorname{ResNeXt101} [7] (32 \mathrm{x4d}) + \operatorname{CBAM}$	48.96M	7.519	21.07	5.59

Grad-CAM visualization

*source: Woo et al., "CBAM: Convolutional block attention module", ECCV, 2018 66

- **1. Evolution of CNN Architectures**
 - AlexNet and ZFNet
 - VGGNet and GoogLeNet
- 2. Architectural Innovations in CNNs
 - Batch normalization and ResNet
 - Beyond ResNet
 - Toward automation of network design
 - Principle of network scaling
- 3. Explorations in CNN Modules
 - Dilated and Deformable convolution
 - Attention module in CNNs

4. Observational Study on CNN Architectures

- ResNets behave like ensembles of relatively shallow nets
- Visualizing the loss landscape of neural nets
- Essentially no barriers in neural network energy landscape

ResNets behave like ensembles of relatively shallow nets [Veit et al., 2016]

- ResNet improved generalization by revolution of depth
 Quiz: But, does it fully explain why deep ResNets generalize well?
- Increasing depth **does not always mean** better generalization
 - Naïve CNNs are very easy to overfit on deeper networks [Eigen et al., 2014]



ResNets behave like ensembles of relatively shallow nets [Veit et al., 2016]

- Veit et al. (2016): ResNet can be viewed as a collection of many paths, instead of a single ultra-deep network
 - Each module in a ResNet receives a **mixture of** 2^{n-1} **different distributions**

$$y_{3} = y_{2} + f_{3}(y_{2})$$

= $y_{1} + f_{2}(y_{1})$] + $f_{3}(y_{1} + f_{2}(y_{1}))$
= $y_{0} + f_{1}(y_{0}) + f_{2}(y_{0} + f_{1}(y_{0}))$] + $f_{3}(y_{0} + f_{1}(y_{0}) + f_{2}(y_{0} + f_{1}(y_{0})))$



(a) Conventional 5-block residual networ

- Veit et al. (2016): ResNet can be viewed as a collection of many paths, instead of a single ultra-deep network
 - Deleting a module in ResNet has a minimal effect on performance
 - Similar effect as removing 2^{n-1} paths out of 2^n : still 2^{n-1} paths alive!



Next, visualizing loss functions in CNN

- Trainability of neural nets is highly dependent on network architecture
- However, the effect of each choice on the underlying loss surface is unclear
 - Why are we able to minimize highly non-convex neural loss?
 - Why do the resulting minima generalize?
- Li et al. (2018) analyzes random-direction 2D plot of loss around local minima

$$f(\alpha,\beta) = L(\theta^* + \alpha\delta + \beta\eta)$$

Local minima Random directions

- δ and η are sampled from a random Gaussian distribution
- To remove some scaling effect, δ and η are normalized filter-wise

$$\delta_{i,j} \leftarrow \frac{\delta_{i,j}}{||\delta_{i,j}||} ||\theta_{i,j}|| \qquad i^{\text{th}} \text{ laver, } i^{\text{th}} \text{ filter}$$

- Li et al. (2018) analyzes random-direction 2D plot of loss around local minima
- Modern architectures prevent the loss to be chaotic as depth increases


- Li et al. (2018) analyzes random-direction 2D plot of loss around local minima
- Modern architectures prevent the loss to be chaotic as depth increases



- Li et al. (2018) analyzes random-direction 2D plot of loss around local minima
- Modern architectures prevent the loss to be chaotic as depth increases

1.00 0.75 0.75 0.75 0.50 0.50 0.50 0.25 0.25 0.25 0.00 0.00 0.00 -0.25 -0.25-0.25-0.50 -0.50 -0.50-0.75 -0.7-0.75-1.00 -1.00 -0.75 -0.50 -0.25 0.00 -0.75 -0.50 -0.25 0.00 -0.75 -0.50 -0.25 0.00 0.25 0.25 0.75 0.25 (d) ResNet-20-noshort, 8.18% (e) ResNet-56-noshort, 13.31% (f) ResNet-110-noshort, 16.44%

ResNet, **no shortcuts** ⇒ sharp minima

$ResNet \Rightarrow flat minima$



Algorithmic Intelligence Laboratory

*source : Li et al., "Visualizing the loss landscape of neural nets", ICLR Workshop 2018 74

- Li et al. (2018) analyzes random-direction 2D plot of loss around local minima
- Wide-ResNet lead the network toward more flat minimizer
 - WideResNet-56 with width-multiplier k = 1, 2, 4, 8
 - Increased width flatten the minimizer in ResNet



WRN-56, no shortcuts



Next, minimum energy paths in CNNs

Algorithmic Intelligence Laboratory

*source : Li et al., "Visualizing the loss landscape of neural nets", ICLR Workshop 2018 75

• Draxler et al. (2018) analyzes minimum energy paths [Jónsson et al., 1998] between two local minima θ_1 and θ_2 of a given model:

$$p(\theta_i, \theta_2)^* = \operatorname*{argmin}_{\text{path } p: \ \theta_1 \to \theta_2} \left(\max_{\theta \in p} L(\theta) \right)$$

- They found a path $\theta_1 \rightarrow \theta_2$ with almost zero barrier
 - A path that **keeps low loss constantly** both in training and test
- The gap vanishes as the model grows, especially on modern architectures
 - e.g. ResNet, DenseNet
- Minima of a loss of deep neural networks are perhaps on a single connected manifold



DenseNet-40-12

Essentially no barriers in neural network energy landscape [Draxler et al., 2018]

- For a given model with two local minima θ_1 and θ_2 , they applied AutoNEB [Kolsbjerg et al., 2016] to find a minimum energy path
 - A state-of the-art for connecting minima from molecular statistical mechanics
- The deeper and wider an architecture, the lower are the saddles between minima
- They essentially vanish for current-day deep architectures
- The test accuracy is also preserved
 - **CIFAR-10**: < +0.5%
 - CIFAR-100: < +2.2%



Essentially no barriers in neural network energy landscape [Draxler et al., 2018]

- The deeper and wider an architecture, the lower are the barriers
- They essentially vanish for current-day deep architectures
- Why do this phenomenon happen?
 - **Parameter redundancy** may help to **flatten** the neural loss



Algorithmic Intelligence Laboratory

Summary

- The larger the network, the more difficult it is to design
 - 1. Optimization difficulty
 - 2. Generalization difficulty
- ImageNet challenge contributed greatly to development of CNN architectures
- **ResNet**: Optimization ⇒ Generalization
 - Many variants of ResNet have been emerged
 - Very recent trends towards **network design and scaling**
- Many types of **CNN modules** are explored to capture detailed spatial information
 - Dilated and deformable convolution
 - Attention based modules
- Many observational study supports the advantages of modern CNN architectures

[Jónsson et al., 1998] Jónsson, H., Mills, G., & Jacobsen, K. W. (1998). Nudged elastic band method for finding minimum energy paths of transitions. In Classical and quantum dynamics in condensed phase simulations (pp. 385-404).

link : https://www.worldscientific.com/doi/abs/10.1142/9789812839664_0016

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324. link : <u>https://ieeexplore.ieee.org/abstract/document/726791/</u>

[Bergstra et al., 2012] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305. link : http://www.jmlr.org/papers/v13/bergstra12a.html

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105). link : <u>http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks</u>

[Farabet et al., 2013] Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2013). Learning hierarchical features for scene labeling. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1915-1929. link : <u>https://ieeexplore.ieee.org/abstract/document/6338939/</u>

[Eigen et al., 2014] Eigen, D., Rolfe, J., Fergus, R., & LeCun, Y. (2013). Understanding Deep Architectures using a Recursive Convolutional Network. ArXiv Preprint ArXiv:1312.1847, 1–9. link : <u>http://arxiv.org/abs/1312.1847</u>

[Simonyan et al., 2014] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. link: <u>https://arxiv.org/abs/1409.1556</u> [Zeiler et al., 2014] Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer, Cham. link : <u>https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53</u>

[Ioffe et al., 2015] Ioffe, S. & Szegedy, C.. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Proceedings of the 32nd International Conference on Machine Learning, in PMLR 37:448-456

link : <u>http://proceedings.mlr.press/v37/ioffe15.html</u>

[Ren et al., 2015] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*(pp. 91-99). link : <u>http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks</u>

[Russakovsky et al., 2015] Russakovsky, O. et al. (2015). Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), 211-252. link : <u>https://link.springer.com/article/10.1007/s11263-015-0816-y</u>

[Szegedy et al., 2015] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

link : <u>https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_</u> <u>CVPR_paper.html</u>

[Han et al., 2016] Han, D., Kim, J., & Kim, J. (2017, July). Deep pyramidal residual networks. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on (pp. 6307-6315). IEEE. link : <u>https://ieeexplore.ieee.org/document/8100151/</u>

[Yu et al., 2016] Yu, F., & Koltun, V. (2016) Multi-Scale Context Aggregation by Dilated Convolutions. In International Conference on Learning Representations. link : <u>https://arxiv.org/abs/1511.07122</u>

References

[He et al., 2016a] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778). link : <u>https://ieeexplore.ieee.org/document/7780459/</u>

[He et al., 2016b] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In European conference on computer vision (pp. 630-645). Springer, Cham. link : <u>https://link.springer.com/chapter/10.1007/978-3-319-46493-0_38</u>

[Huang et al., 2016] Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K. Q. (2016). Deep networks with stochastic depth. In European Conference on Computer Vision (pp. 646-661). link : <u>https://link.springer.com/chapter/10.1007/978-3-319-46493-0_39</u>

[Kolsbjerg et al., 2016] Kolsbjerg, E. L., Groves, M. N., & Hammer, B. (2016). An automated nudged elastic band method. The Journal of chemical physics, 145(9), 094107. link : <u>https://aip.scitation.org/doi/abs/10.1063/1.4961868</u>

[Targ et al., 2016] Targ, S., Almeida, D., & Lyman, K. (2016). Resnet in Resnet: generalizing residual architectures. arXiv preprint arXiv:1603.08029.

link : <u>https://arxiv.org/abs/1603.08029</u>

[Veit et al., 2016] Veit, A., Wilber, M. J., & Belongie, S. (2016). Residual networks behave like ensembles of relatively shallow networks. In Advances in Neural Information Processing Systems (pp. 550-558). link : <u>http://papers.nips.cc/paper/6556-residual-networks-behave-like-ensembles-of-relatively-shallow-networks</u>

[Xie et al., 2016] Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017, July). Aggregated residual transformations for deep neural networks. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on (pp. 5987-5995). IEEE.

link : <u>http://openaccess.thecvf.com/content_cvpr_2017/papers/Xie_Aggregated_Residual_Transformations_</u> <u>CVPR_2017_paper.pdf</u>

[Zagoruyko et al., 2016] Zagoruyko, S. and Komodakis, N. (2016). Wide Residual Networks. In Proceedings of the British Machine Vision Conference (pp. 87.1-87.12).

link : http://www.bmva.org/bmvc/2016/papers/paper087/index.html

[Zoph et al., 2016] Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.

link : https://arxiv.org/abs/1611.01578

[Chen et al., 2017] Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., & Feng, J. (2017). Dual path networks. In Advances in Neural Information Processing Systems (pp. 4467-4475).

link: https://papers.nips.cc/paper/7033-dual-path-networks

[Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017, July). Densely Connected Convolutional Networks. In CVPR (Vol. 1, No. 2, p. 3). link: http://openaccess.thecvf.com/content_cvpr_2017/papers/Huang_Densely_Connected_Convolutional_ CVPR 2017 paper.pdf

[Szegedy et al., 2017] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017, February). Inception-v4, inceptionresnet and the impact of residual connections on learning. In AAAI (Vol. 4, p. 12). link : https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/download/14806/14311

[Dai et al., 2017] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 764-773).

link : https://arxiv.org/abs/1703.06211v3

[Chen et al., 2017] Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.

link : https://arxiv.org/abs/1706.05587

[Draxler et al., 2018] Draxler, F., Veschgini, K., Salmhofer, M. & Hamprecht, F. (2018). Essentially No Barriers in Neural Network Energy Landscape. Proceedings of the 35th International Conference on Machine Learning, in PMLR 80:1309-1318.

link : <u>http://proceedings.mlr.press/v80/draxler18a.html</u>

[Luo et al., 2018] Luo, R., Tian, F., Qin, T., Chen, E. & Liu, T. (2018) Neural Architecture Optimization. arXiv preprint arXiv:1808.07233.

link : <u>https://arxiv.org/abs/1808.07233</u>

[Li et al., 2018] Li, H., Xu, Z., Taylor, G., & Goldstein, T. (2017). Visualizing the loss landscape of neural nets. arXiv preprint arXiv:1712.09913. link : https://arxiv.org/abs/1712.09913

[Pham et al., 2018] Pham, H., Guan, M., Zoph, B., Le, Q. & Dean, J.. (2018). Efficient Neural Architecture Search via Parameters Sharing. Proceedings of the 35th International Conference on Machine Learning, in PMLR 80:4095-4104 link : <u>http://proceedings.mlr.press/v80/pham18a.html</u>

[Real et al., 2018] Real, E., Aggarwal, A., Huang, Y., & Le, Q. V. (2018). Regularized evolution for image classifier architecture search. arXiv preprint arXiv:1802.01548. link : https://arxiv.org/abs/1802.01548

[Zoph et al., 2018] Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2017). Learning transferable architectures for scalable image recognition. arXiv preprint arXiv:1707.07012, 2(6).

link : <u>http://openaccess.thecvf.com/content_cvpr_2018/papers/Zoph_Learning_Transferable_Architectures_</u> <u>CVPR_2018_paper.pdf</u>

[Brock et al., 2018] Brock, A., Lim, T., Ritchie, J. M., & Weston, N. (2018). SMASH: one-shot model architecture search through hypernetworks. In International Conference on Learning Representations. link : <u>https://openreview.net/forum?id=rydeCEhs-</u>

References

[Hu et al., 2018] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132-7141). link : <u>https://arxiv.org/abs/1709.01507</u>

[Woo et al., 2018] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 3-19).

link : https://arxiv.org/abs/1807.06521v2

[Tan et al., 2019] Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning* (pp. 6105-6114). PMLR. link : <u>https://arxiv.org/abs/1905.11946</u>