Deep Learning for Image Classification

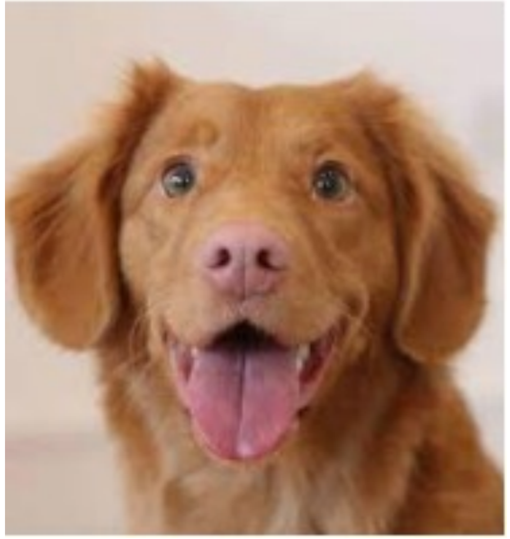
CS6420

Wei Ji

Problem Definition

Input: image

Output: image label among a fixed set of categories



Cat

Dog

Boat

Horse

Car

……

Core tasks in Computer Vision

What animal in the picture?

image classification

What animals in the picture and where?

object detection

Which pixels belong to which object?

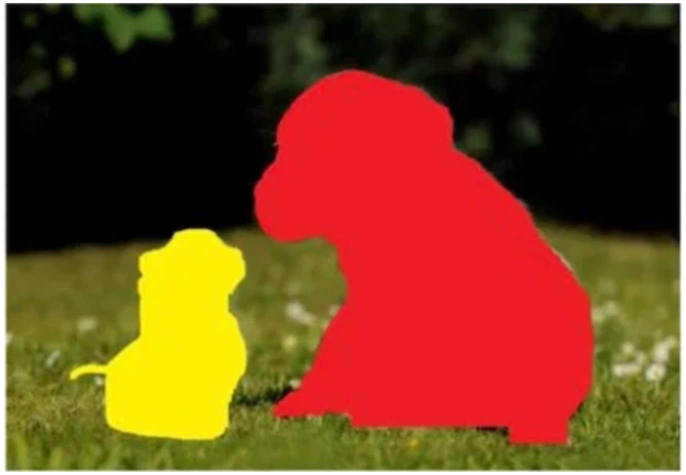
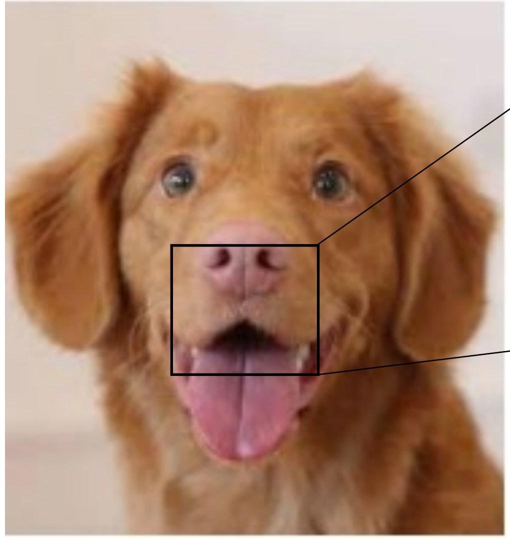


image segmentation

Problem: Semantic Gap

Film “The Matrix”

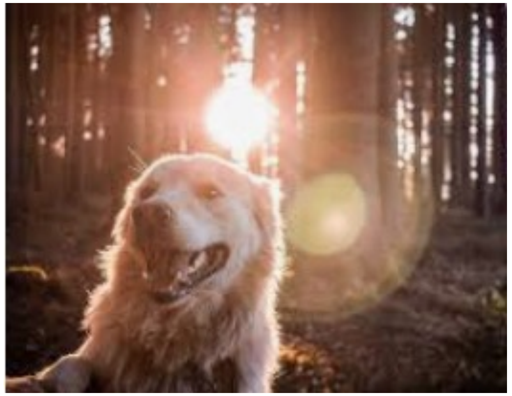


```
[[105 112 108 111 104  99 106  99  96 103 112 119 104  97  93  87]
 [ 91  98 102 106 104  79  98 103  99 105 123 136 110 105  94  85]
 [ 76  85  90 105 128 105  87  96  95  99 115 112 106 103  99  85]
 [ 99  81  81  93 120 131 127 100  95  98 102  99  96  93 101  94]
 [106  91  61  64  69  91  88  85 101 107 109  98  75  84  96  95]
 [114 108  85  55  55  69  64  54  64  87 112 129  98  74  84  91]
 [133 137 147 103  65  81  80  65  52  54  74  84 102  93  85  82]
 [128 137 144 140 109  95  86  70  62  65  63  63  60  73  86 101]
 [125 133 148 137 119 121 117  94  65  79  80  65  54  64  72  98]
 [127 125 131 147 133 127 126 131 111  96  89  75  61  64  72  84]
 [115 114 109 123 150 148 131 118 113 109 100  92  74  65  72  78]
 [ 89  93  90  97 108 147 131 118 113 114 113 109 106  95  77  80]
 [ 63  77  86  81  77  79 102 123 117 115 117 125 125 130 115  87]
 [ 62  65  82  89  78  71  80 101 124 126 119 101 107 114 131 119]
 [ 63  65  75  88  89  71  62  81 120 138 135 105  81  98 110 118]
 [ 87  65  71  87 106  95  69  45  76 130 126 107  92  94 105 112]
 [118  97  82  86 117 123 116  66  41  51  95  93  89  95 102 107]
 [164 146 112  80  82 120 124 104  76  48  45  66  88 101 102 109]
 [157 170 157 120  93  86 114 132 112  97  69  55  70  82  99  94]
 [130 128 134 161 139 100 109 118 121 134 114  87  65  53  69  86]
 [128 112  96 117 150 144 120 115 104 107 102  93  87  81  72  79]
 [123 107  96  86  83 112 153 149 122 109 104  75  80 107 112  99]
 [122 121 102  80  82  86  94 117 145 148 153 102  58  78  92 107]
 [122 164 148 103  71  56  78  83  93 103 119 139 102  61  69  84]]
```

An image is a big grid of numbers between [0,255]:

e.g. 400 *400*3 (3 channels RGB)

Challenge: Illumination Changes



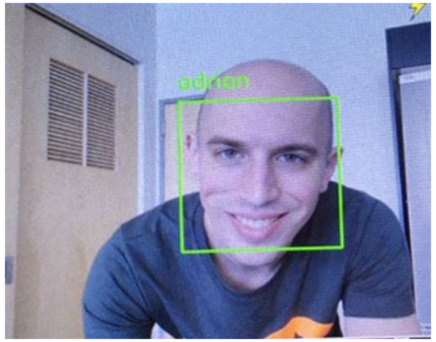
Challenge: Viewpoint Variation

Challenge: Fine-Grained Categories

Applications:

- Face recognition



- Galaxy Classification

- Traffic Sign

- ……

Image Classification: Data-Driven Approach

- 1. Collect a dataset of images and labels

- 2. Use NN to train a classifier

- 3. Evaluate the classifier on new images

Image Classification : Data-Driven Approach

- 1. Collect a dataset of images and labels
- 2. Use NN to train a classifier
- 3. Evaluate the classifier on new images

Image Classification Dataset: MNIST

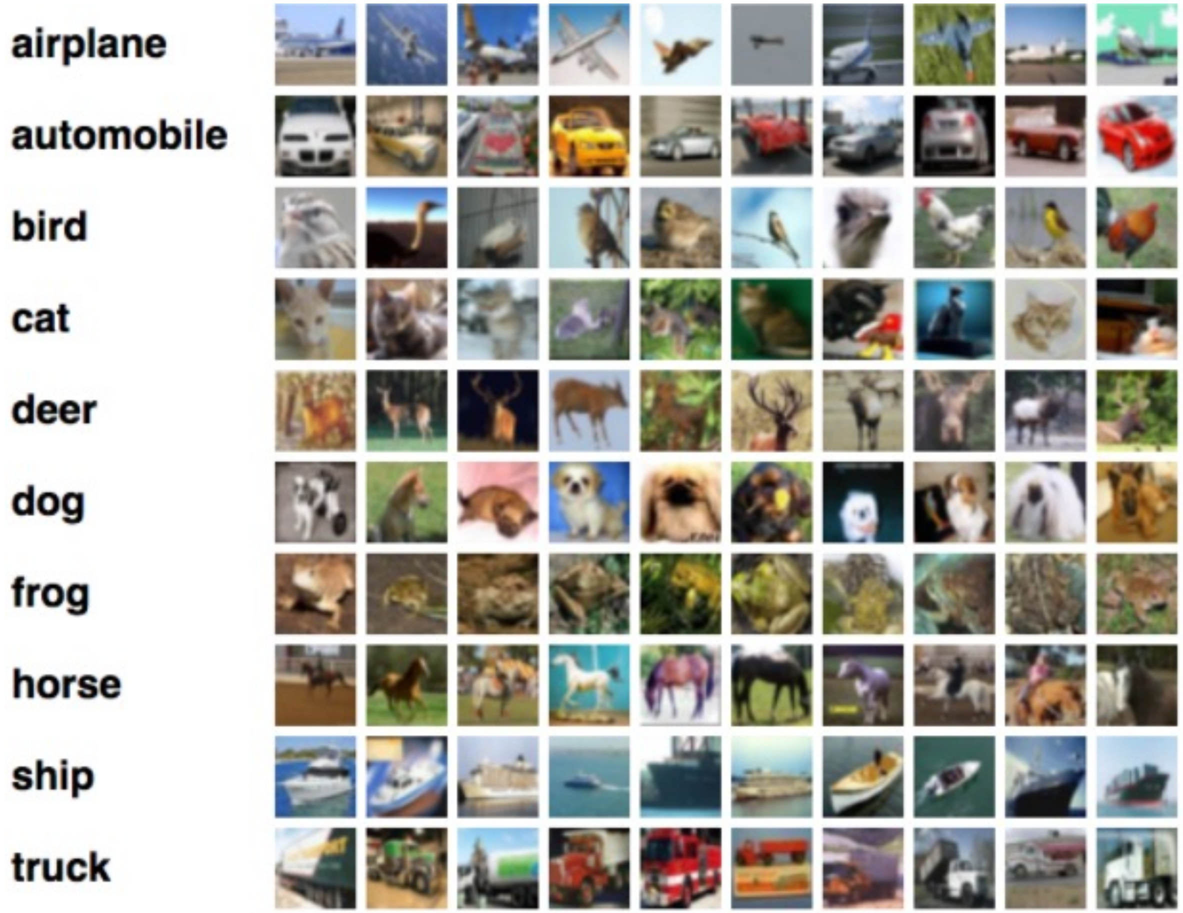
10 classes: Digits 0 to 9

28*28 grayscale images

60k training images

10k test images

Image Classification Dataset: CIFAR-10



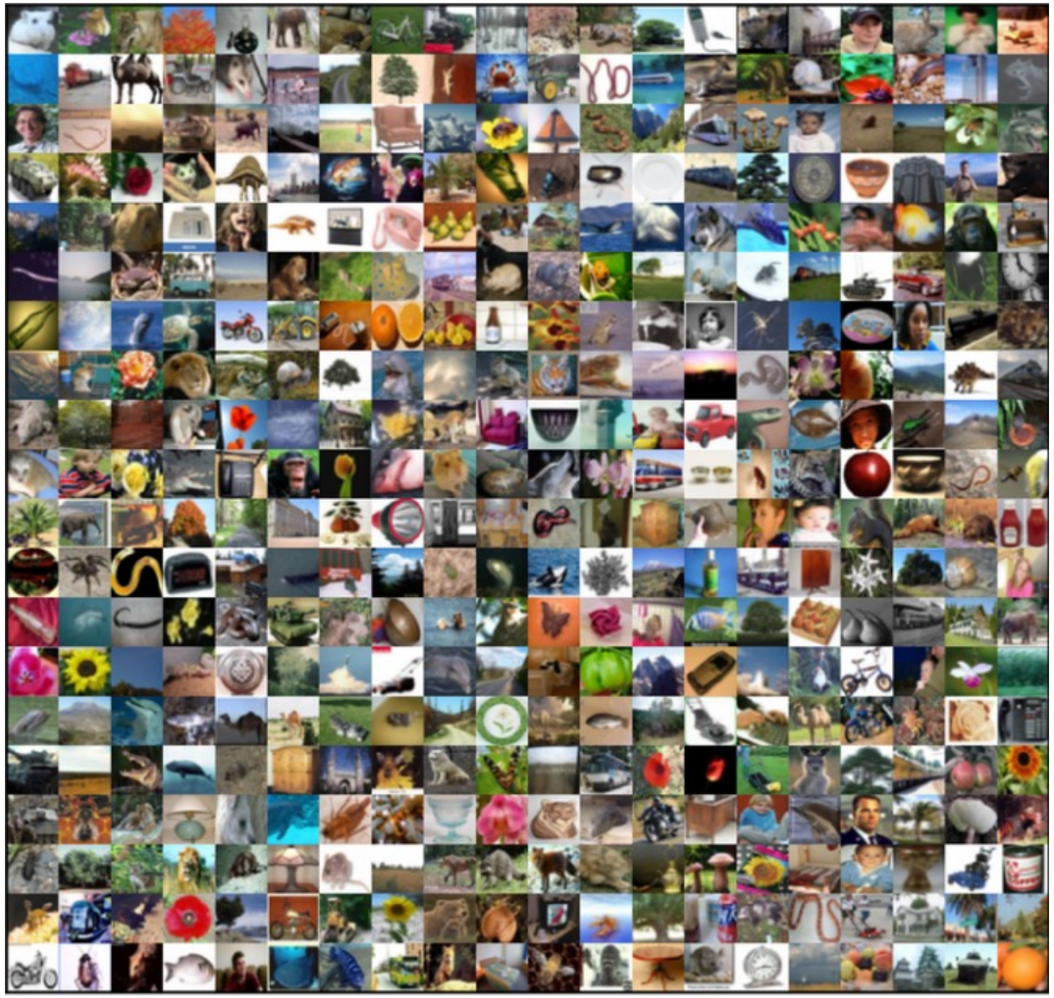
10 classes

50k training images (5k per class)

10k testing images (1k per class)

32*32 RGB images

Image Classification Dataset: CIFAR-100



100 classes

50k training images (500 per class)

10k testing images (100 per class)

32*32 RGB images

Image Classification Dataset: ImageNet

1000 classes

~1.3M training images (~1.3K per class)

50K validation images (50 per class)

100K testing images (100 per class)

Performance metric: Top 5 accuracy
Algorithm predicts 5 labels for each image;
One of them needs to be right

Images have variable size, but often resized to
256*256 for training

Image Classification : Data-Driven Approach

- 1. Collect a dataset of images and labels

  MNIST; CIFAR; ImageNet
- 2. Use NN to train a classifier
- 3. Evaluate the classifier on new images

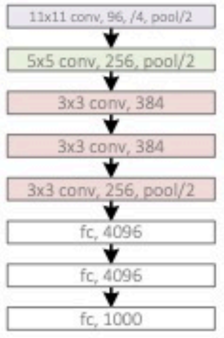
Image Classification : Data-Driven Approach

- 1. Collect a dataset of images and labels

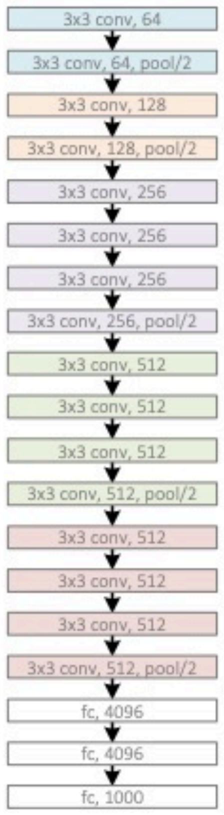
  MNIST; CIFAR; ImageNet
- 2. Use NN to train a classifier
- 3. Evaluate the classifier on new images

Background: AlexNet & VGG & GoogLeNet

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



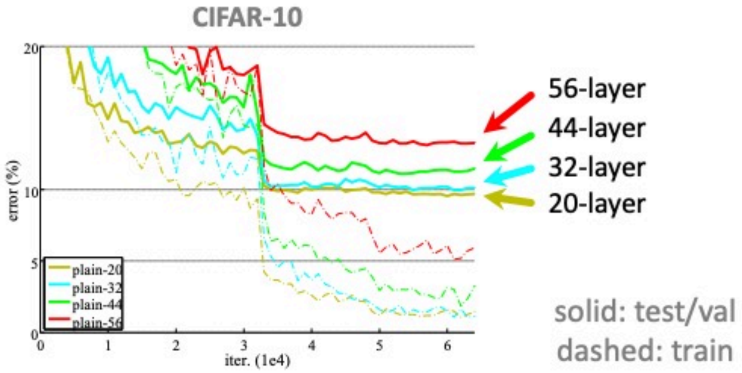
GoogleNet, 22 layers
(ILSVRC 2014)

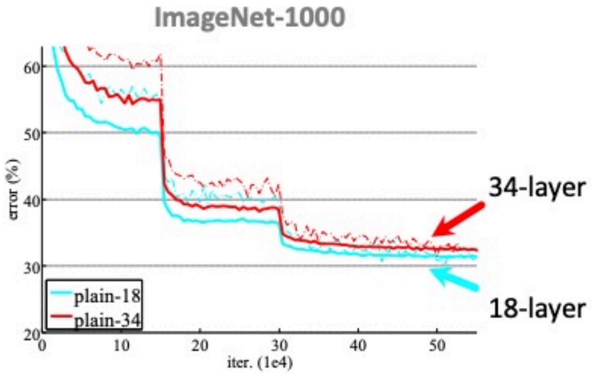
AlexNet: Group Convolution for 6G GPU memory constraint

VGG: Same blocks in all layers

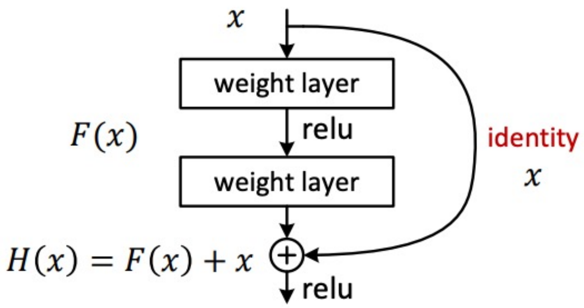
GoogleNet: Concatenate multi-branch for multi-scale information

ResNet





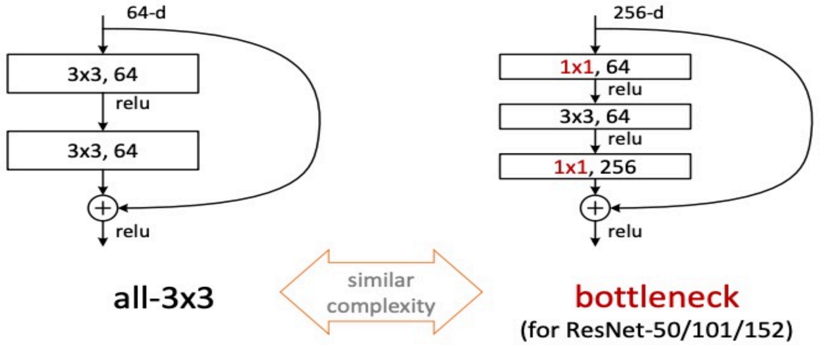
- Residual net



- “Overly deep” plain nets have higher training error (not caused by overfitting).
- <u>Problem is caused by optimization issues, instead of representational abilities</u>.
- Residual Learning: add identity mapping (a deeper model should produce no higher training error than its shallower counterpart).

ResNet

- A practical design of going deeper



why directly **identity shortcuts** instead of projection shortcuts?

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Table 3. Error rates (%, **10-crop** testing) on ImageNet validation.

- A: zero-padding for increase dims
- B: projection shortcuts for increase dims, identity for others
- C: all shortcuts are projection shortcuts

Shortcuts choices

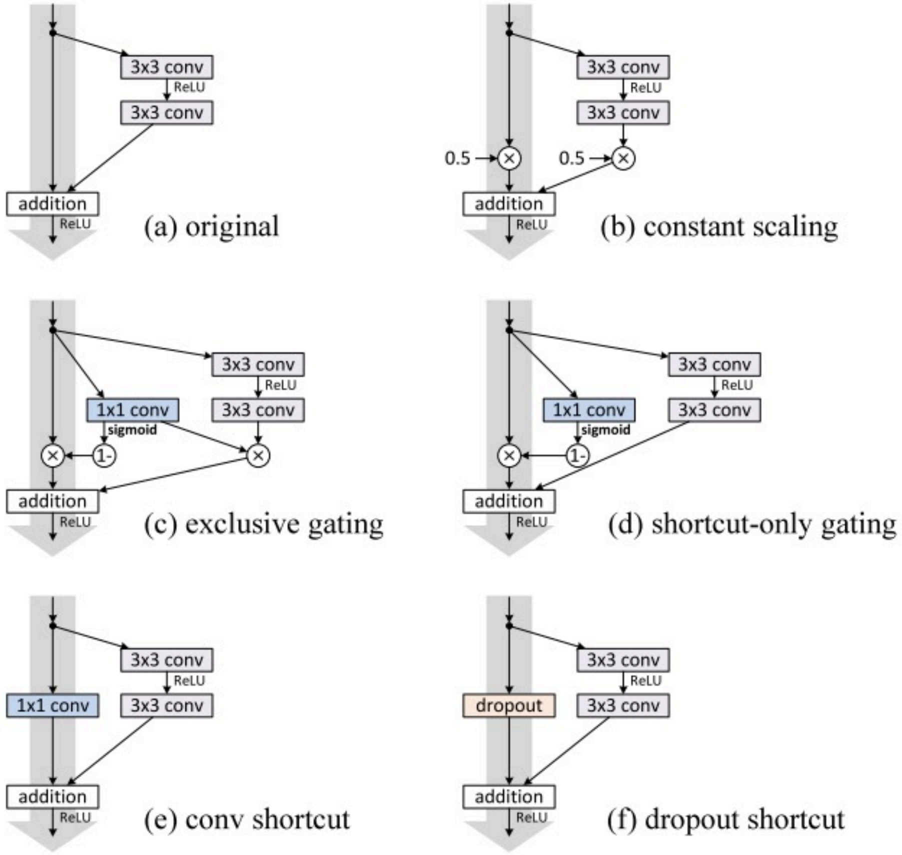


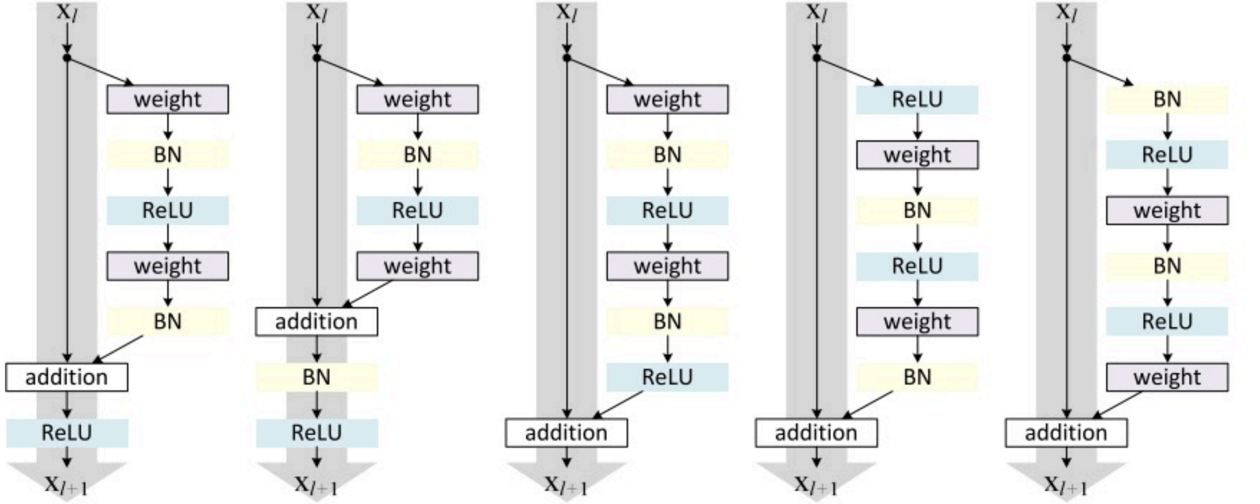
Table 1. Classification error on the CIFAR-10 test set using ResNet-110 [1], with different types of shortcut connections applied to all Residual Units. We report "fail" when the test error is higher than 20%.

case	Fig.	on shortcut	on $\mathcal{F}$	error (%)	remark
original [1]	Fig. 2(a)	1	1	**6.61**	
constant scaling	Fig. 2(b)	0	1	fail	This is a plain net
		0.5	1	fail	
		0.5	0.5	12.35	frozen gating
exclusive gating	Fig. 2(c)	$1-g(\mathbf{x})$	$g(\mathbf{x})$	fail	init b_g=0 to −5
		$1-g(\mathbf{x})$	$g(\mathbf{x})$	8.70	init b_g=-6
		$1-g(\mathbf{x})$	$g(\mathbf{x})$	9.81	init b_g=-7
shortcut-only gating	Fig. 2(d)	$1-g(\mathbf{x})$	1	12.86	init b_g=0
		$1-g(\mathbf{x})$	1	6.91	init b_g=-6
1×1 conv shortcut	Fig. 2(e)	1×1 conv	1	12.22	
dropout shortcut	Fig. 2(f)	dropout 0.5	1	fail	

Identity shortcuts is the best choices!

(a) is a special case of (d) and (e), results shows there is still optimization issues in other choices.

Activation function choices

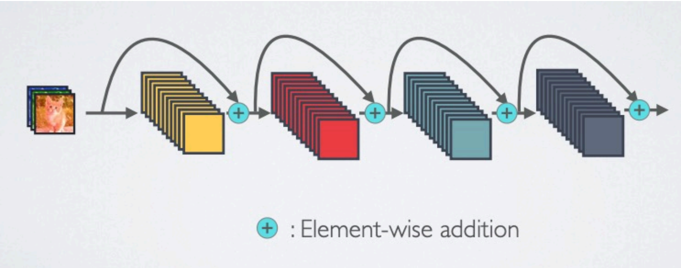


(a) original (b) BN after addition (c) ReLU before addition (d) ReLU-only pre-activation (e) **full pre-activation**

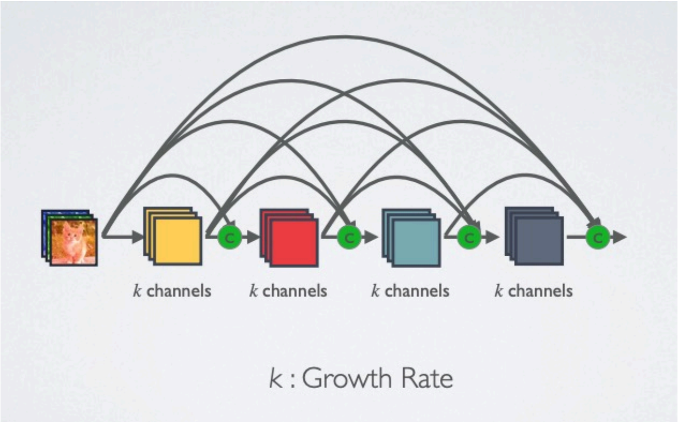
case	Fig.	ResNet-110	ResNet-164
original Residual Unit [1]	Fig. 4(a)	6.61	5.93
BN after addition	Fig. 4(b)	8.17	6.50
ReLU before addition	Fig. 4(c)	7.84	6.14
ReLU-only pre-activation	Fig. 4(d)	6.71	5.91
full pre-activation	Fig. 4(e)	**6.37**	**5.46**

DenseNet

ResNet



DenseNet



Size:
DenseNet-121 < ResNet-50

•**ResNet**: combines features through summation of identity shortcuts and residual block, which may impede the information flow.

•**DenseNet**: directly passes each feature maps to all preceding layers.

- Alleviate the vanishing-gradient problem
- Strengthen feature propagation
- Substantially reduce the number of parameters

SENet

•**Squeeze**: global information embedding (global average pooling)

•**Excitation**: Adaptive Recalibration

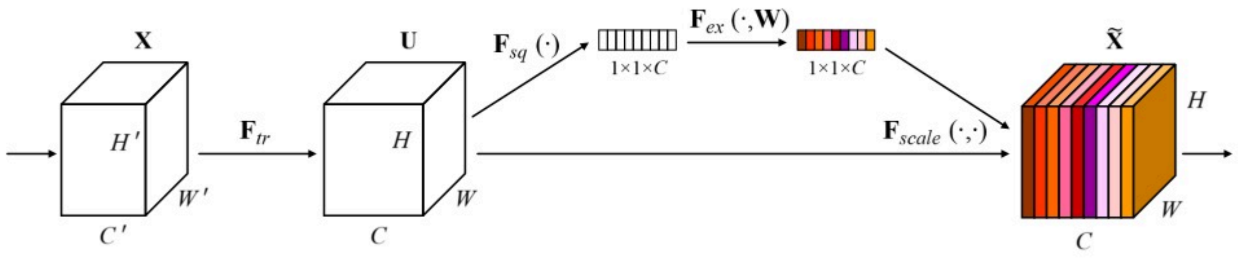


Figure 1: A Squeeze-and-Excitation block.

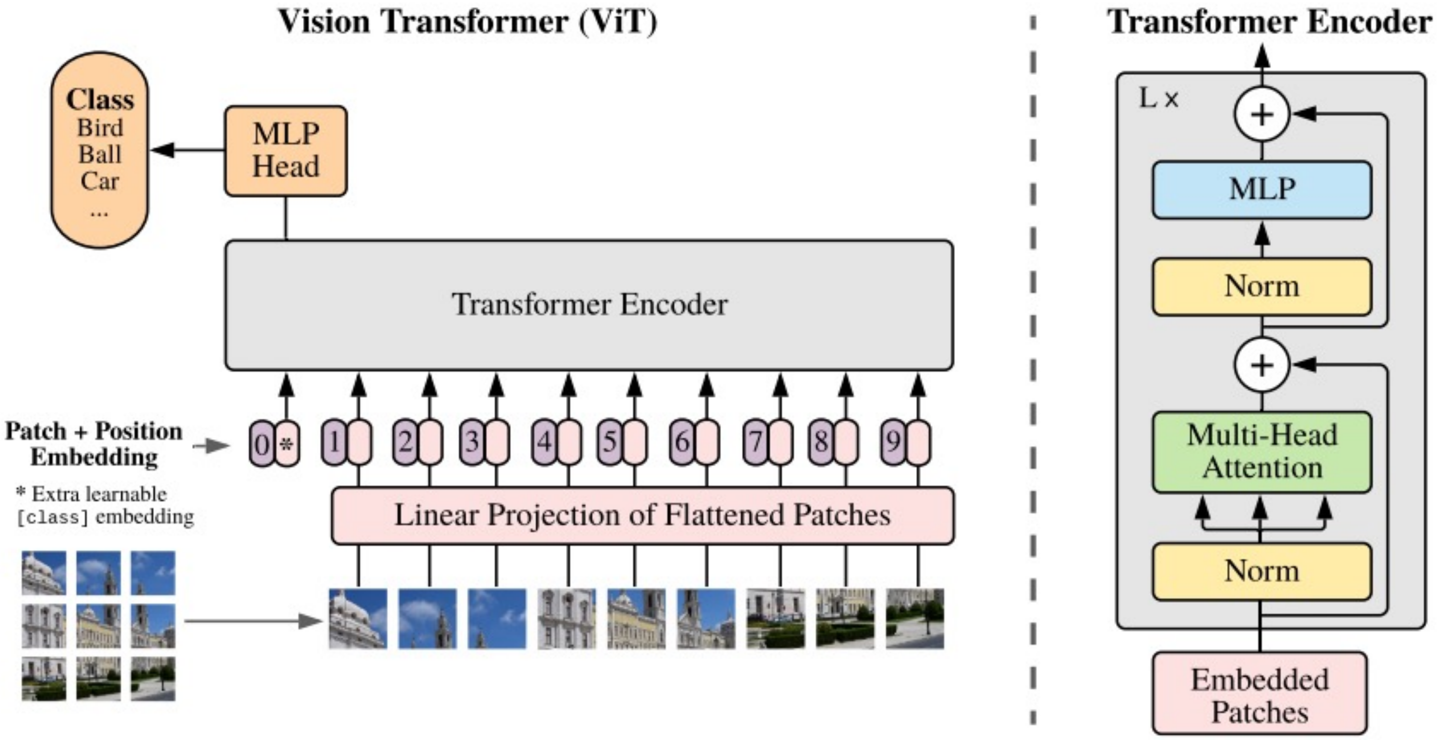
•**Squeeze**:

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j).$$

•**Excitation**:

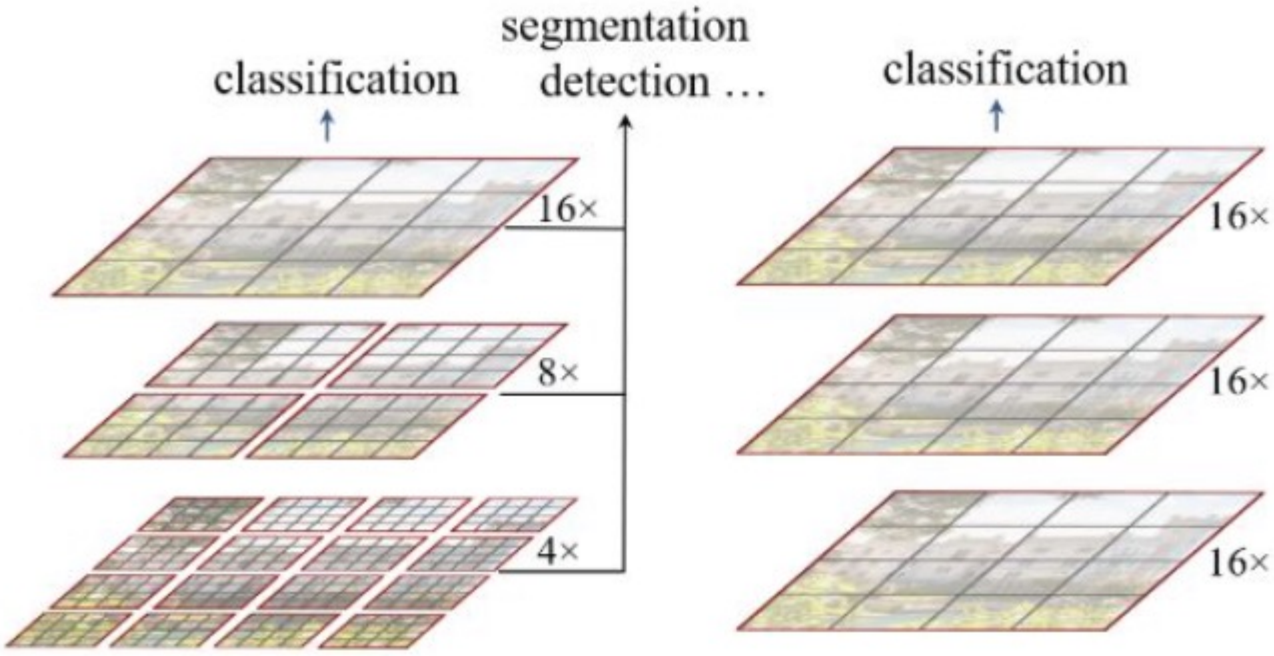
$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z},\mathbf{W}) = \sigma(g(\mathbf{z},\mathbf{W})) = \sigma(\mathbf{W}_2\delta(\mathbf{W}_1\mathbf{z})),$$

ViT (Vision Transformer)



Patch number is fixed due to Position Embedding.

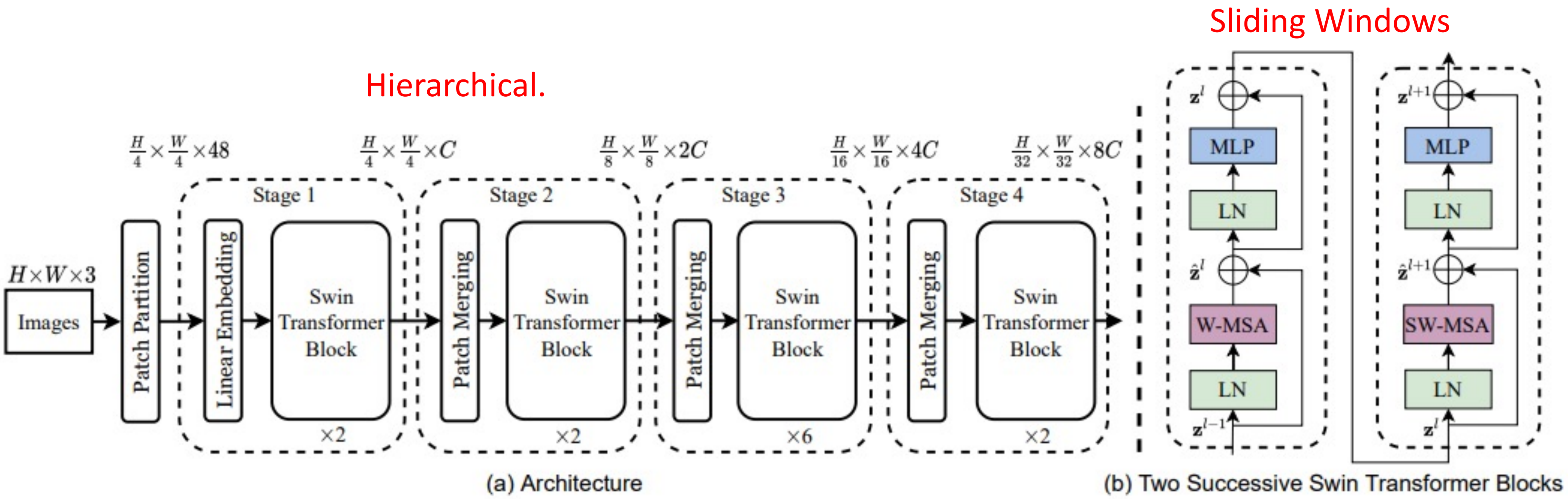
Swin transformer: Hierarchical vision transformer using shifted windows (*CNN-like ViT*)



(a) Swin Transformer (ours) (b) ViT

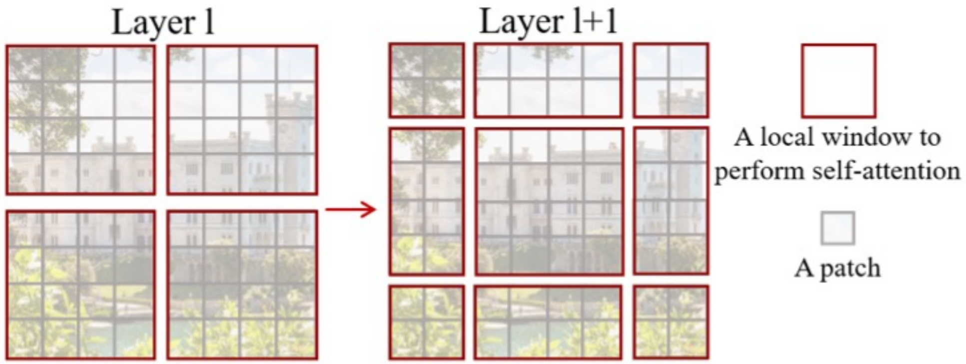
- Hierarchical representation by starting from small-sized patches (outlined in gray) and gradually merging neighboring patches in deeper Transformer layers.
- The linear computational complexity is achieved by computing self-attention locally within non-overlapping windows that partition an image (outlined in red).

Swin transformer: Hierarchical vision transformer using shifted windows



(a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks. W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Swin transformer: Hierarchical vision transformer using shifted windows



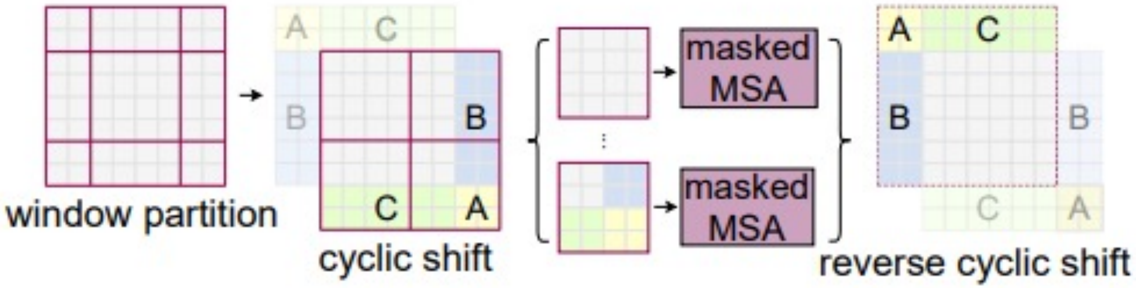


Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

(a) Regular ImageNet-1K trained models					
method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
RegNetY-4G [48]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [48]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [48]	224^2	84M	16.0G	334.7	82.9
EffNet-B3 [58]	300^2	12M	1.8G	732.1	81.6
EffNet-B4 [58]	380^2	19M	4.2G	349.4	82.9
EffNet-B5 [58]	456^2	30M	9.9G	169.1	83.6
EffNet-B6 [58]	528^2	43M	19.0G	96.9	84.0
EffNet-B7 [58]	600^2	66M	37.0G	55.1	84.3
ViT-B/16 [20]	384^2	86M	55.4G	85.9	77.9
ViT-L/16 [20]	384^2	307M	190.7G	27.3	76.5
DeiT-S [63]	224^2	22M	4.6G	940.4	79.8
DeiT-B [63]	224^2	86M	17.5G	292.3	81.8
DeiT-B [63]	384^2	86M	55.4G	85.9	83.1
Swin-T	224^2	29M	4.5G	755.2	81.3
Swin-S	224^2	50M	8.7G	436.9	83.0
Swin-B	224^2	88M	15.4G	278.1	83.5
Swin-B	384^2	88M	47.0G	84.7	84.5
(b) ImageNet-22K pre-trained models					
method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [38]	384^2	388M	204.6G	-	84.4
R-152x4 [38]	480^2	937M	840.5G	-	85.4
ViT-B/16 [20]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [20]	384^2	307M	190.7G	27.3	85.2
Swin-B	224^2	88M	15.4G	278.1	85.2
Swin-B	384^2	88M	47.0G	84.7	86.4
Swin-L	384^2	197M	103.9G	42.1	87.3

Comparison of different backbones on ImageNet-1K classification.

Image Classification : Data-Driven Approach

- 1. Collect a dataset of images and labels

    MNIST; CIFAR; ImageNet

- 2. Use NN to train a classifier

    ResNet; DenseNet; SENet; ……

- 3. Evaluate the classifier on new images

Image Classification : Data-Driven Approach

- 1. Collect a dataset of images and labels

  MNIST; CIFAR; ImageNet

- 2. Use NN to train a classifier

  ResNet; DenseNet; SENet; ……

- 3. Evaluate the classifier on new images

Evaluation

Trained Model

Cat	0.02
Dog	0.94
Boat	0.01
Horse	0.02
Car	0.01

Dog <-> Cat <-> Plane

Benchmarks

Add a Result

These leaderboards are used to track progress in Image Classification

Trend	Dataset	Best Model	Paper	Code	Compare
	ImageNet	CoAtNet-7			See all
	CIFAR-10	ViT-H/14			See all
	CIFAR-100	EffNet-L2 (SAM)			See all
	STL-10	Wide-ResNet-101 (Spinal FC)			See all
	MNIST	Homogeneous ensemble with Simple CNN			See all
	SVHN	WRN28-10 (SAM)			See all
	ImageNet ReaL	Meta Pseudo Labels (EfficientNet-B6-Wide)			See all
	Flowers-102	CCT-14/7x2			See all
	iNaturalist 2018	MAE (ViT-H, 448)			See all
	Clothing1M	PGDF			See all

https://paperswithcode.com/task/image-classification

Thank you!