Computer Vision: Fall 2022 — Lecture 12

Dr. Karthik Mohan

Univ. of Washington, Seattle

November 9, 2022

Check-In

1. How did the first checkpoint on the MP1 go?

Check-In

1. How did the first checkpoint on the MP1 go?
2. Fill out the mid-course survey if you haven't yet!

References

1. Good Book for Machine Learning Concepts
2. Deep Learning Reference
3. Convolutional Neural Networks for Visual Recognition
4. Convolutional Neural Net Tutorial

CNN Publication References

1. Convolutional Neural Networks: A comprehensive survey, 2019
2. A survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, 2021
3. GoogLeNet
4. Top models on ImageNet
5. ResNet ILSVRC paper

Today

1. CNN Architectures Recap
2. ResNet

Popular CNN Architectures Recap

Arch	Year	Mention	Speciality
LeNet	1998	Yann LeCun et al	
AlexNet	2012	*Runner-up	Deeper, Bigger 8 % delta
ZFNet	2013	*Winner	Improvement on AlexNet
GoogLeNet	2014	*Winner	Inception Module 60 MM $\rightarrow$ 4 MM params
VGGNet	2014	*Runner-up	Deep network (16 layers) with 140 MM params
ResNet	2015	*Winner	Skip-connections and Batch-normalization

Table: Why competitions matter? *ILSVRC challenge (Evolution of CNN archs over the years)

Popular CNN Architectures Recap

Year	CNN	Developed By	Error Rates	No. of Parameters	Dataset
1998	LeNet	Yann LeCun		60 Thousand	ImageNet
2012	AlexNet	Alex Krizhevsky, Geoffrey Hinton and Ilya Sutskever	15.3 %	60 Million	
2013	ZFNet	Matthew Zeiler, Rob Fergus	14.8 %		
2014	GoogleNet	Google	6.67 %	4 Million	
2014	VGGNet	Simonyan, Zisserman	7.3 %	138 Million	
2015	ResNet	Kaiming He	3.6 %		

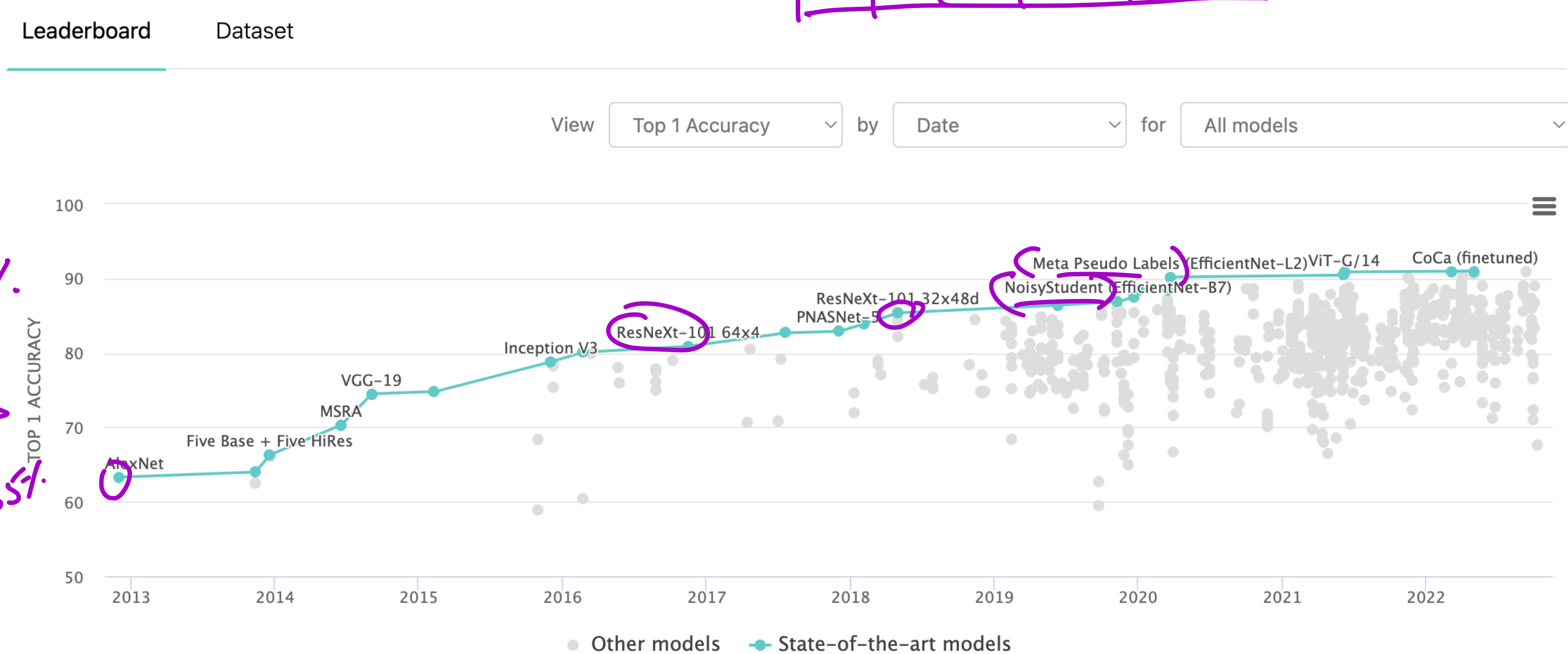
more prone to overfit

Top-1 Accuracy Evolution

Top-1 acc < Top-5 Acc

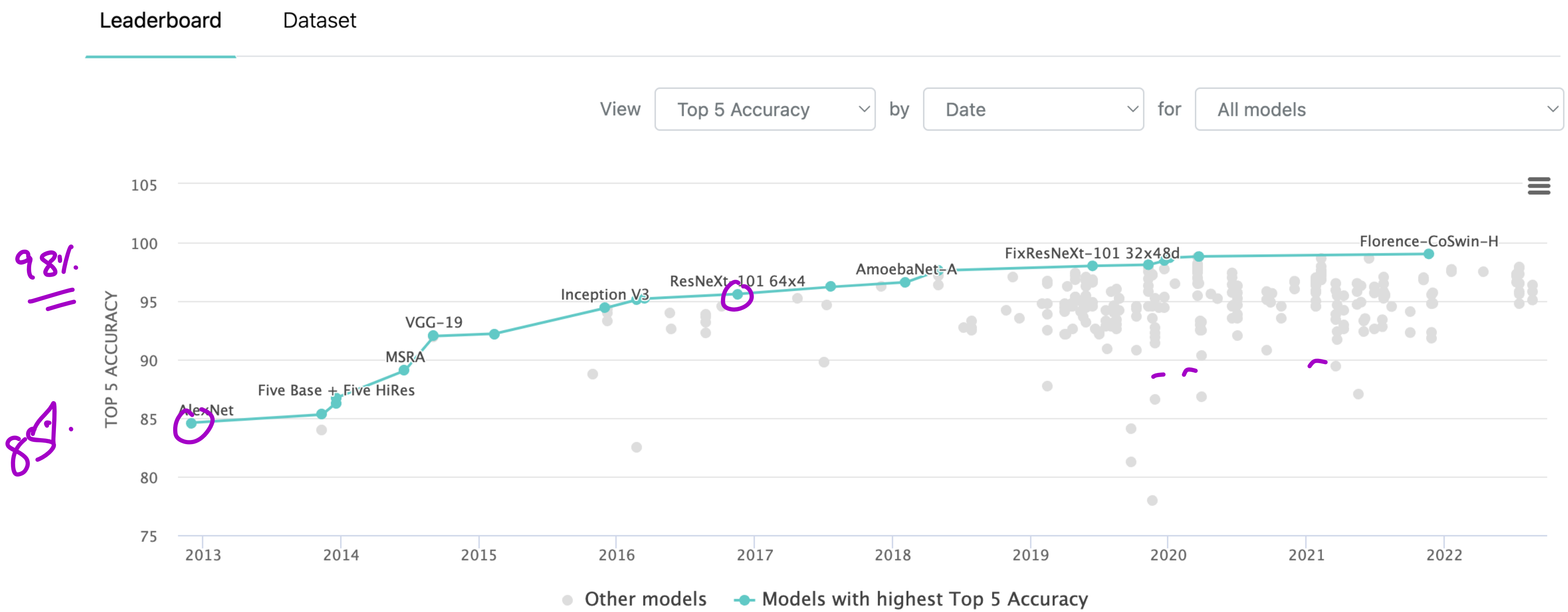
90%

65%



Top models on ImageNet

Top-5 Accuracy Evolution



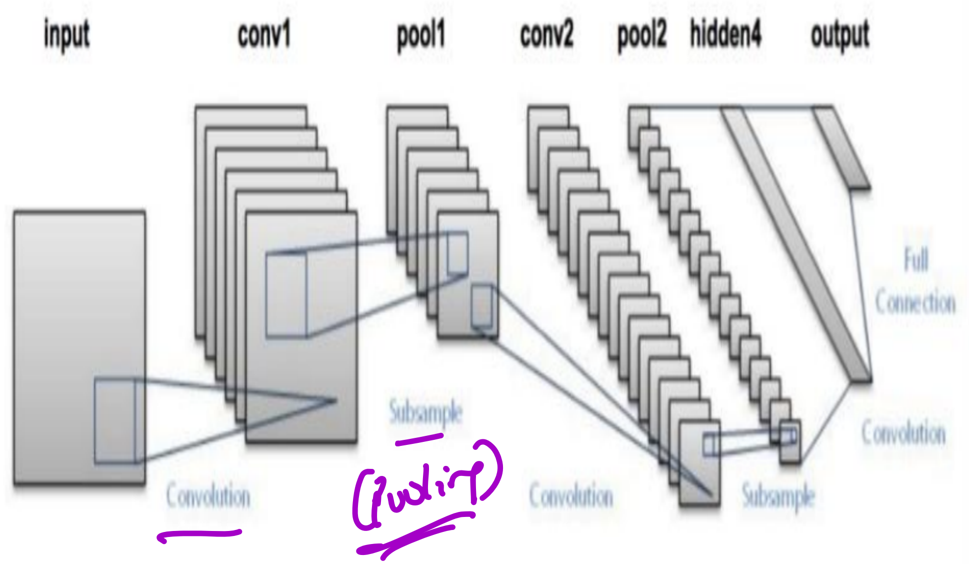
Top models on ImageNet

Popular CNN Architectures

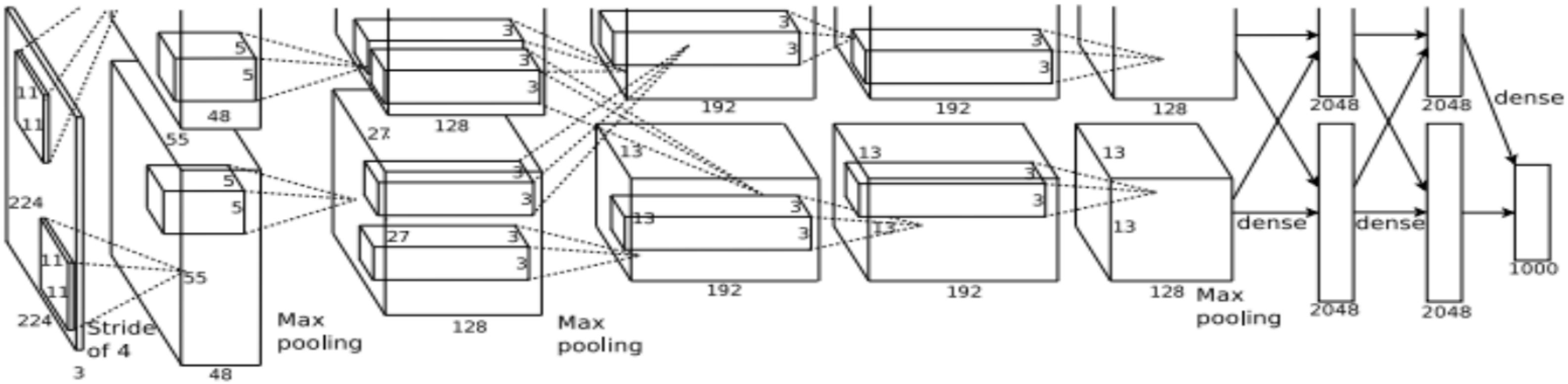
Year	CNN	Developed By	Error Rates	No. of Parameters	Dataset
1998	LeNet	Yann LeCun		60 Thousand	ImageNet
2012	AlexNet	Alex Krizhevsky, Geoffrey Hinton and Ilya Sutskever	15.3 %	60 Million	
2013	ZFNet	Matthew Zeiler, Rob Fergus	14.8 %		
2014	GoogleNet	Google	6.67 %	4 Million	
2014	VGGNet	Simonyan, Zisserman	7.3 %	138 Million	
2015	ResNet	Kaiming He	3.6 %		

LeNet

Yann LeCun

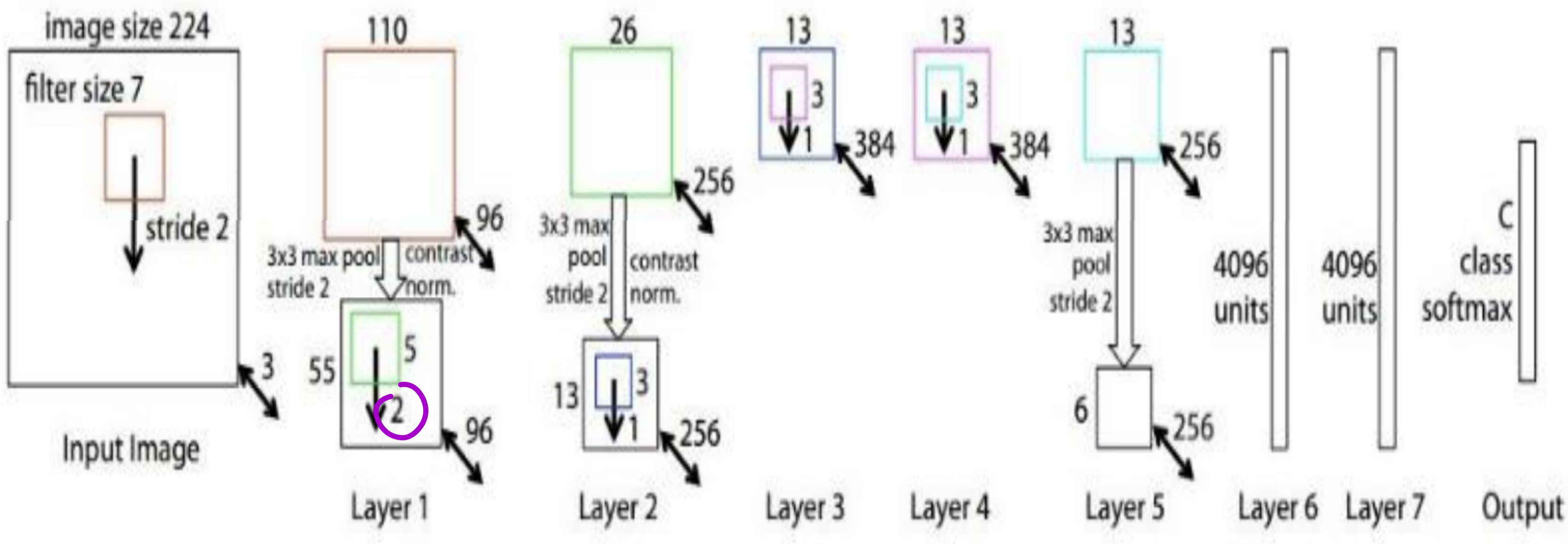


AlexNet



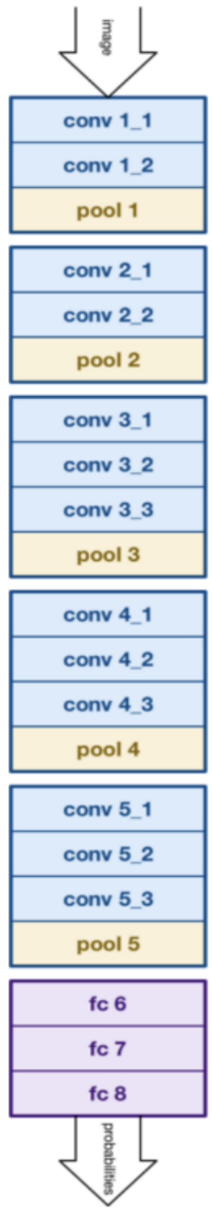
1. Incorporates RELU
2. Deeper layers than LeNet
3. Developed to measure lateral distance between vehicles

ZFNet



1. Hyper-parameter Tweaking in AlexNet
2. Small changes in structure
3. Number of params same as AlexNet: 60MM!
4. Top 5 Accuracy at 85.3% up from 84.6% of AlexNet

VGGNet



more stacking of conv layers as compared AlexNet

more FCs

VGGNet

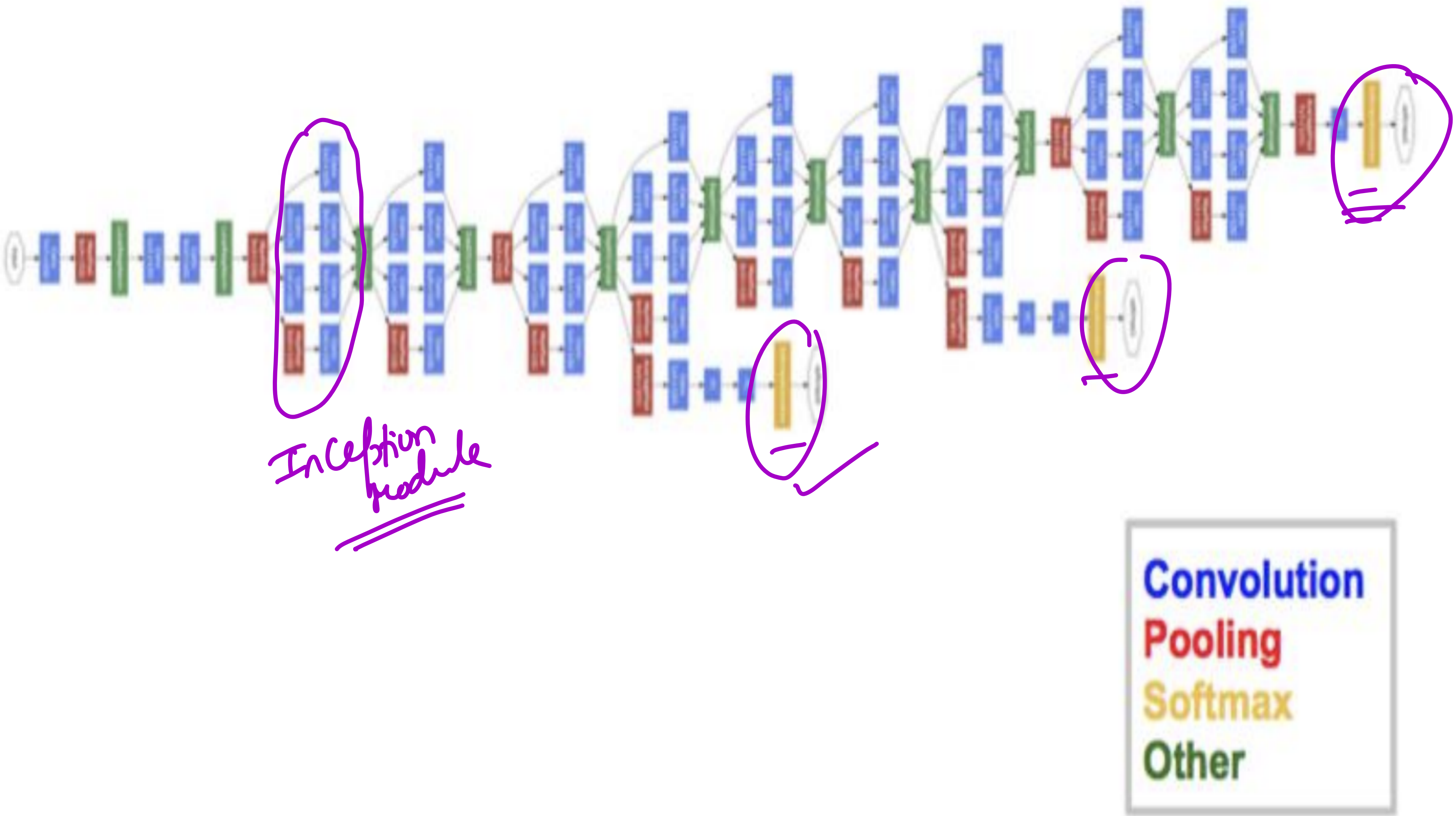
1. Top 5 Accuracy at 92% of VGGNet, up from 85.3% of ZFNet!
2. Runner up in the 2014 competition
3. Number of params: $138MM$, up from $60MM$ of ZFNet!
4. Quite popular for image embeddings and representations
5. Prone to over-fit - Obviously!
6. *Applications:* Finger-print biometric authentication, crack detection, object tracking.

Inception/GoogleNet Motivation

(a) Siberian husky

(b) Eskimo dog

Inception/GoogLeNet

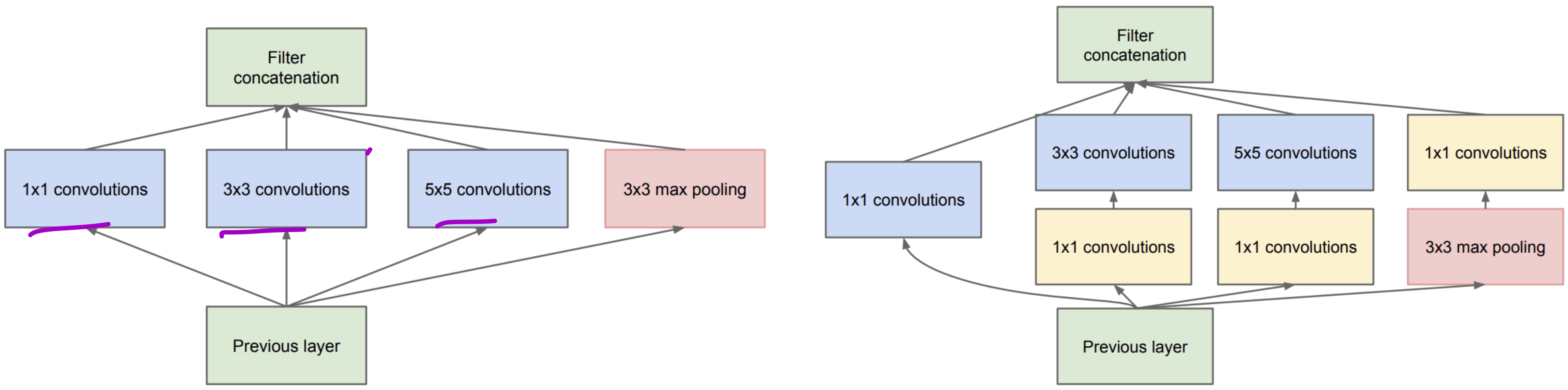


Inception module

Inception/GoogLeNet

1. Top 5 Accuracy at 94.4% up from 92% of VGGNet
2. Introduced an Inception Module
3. Has many more layers than AlexNet or ZFNet!
4. 22 layers deep!
5. Number of params: $4MM$, down from $60MM$ of ZFNet!

Inception Module



(a) Inception module, naïve version (b) Inception module with dimension reductions

1. Concatenates the depth from each of the convolutions
2. Allows for looking at the input at different scales (1x1, 3x3, 5x5, etc)
3. Let's the model use information from all scales

Inception/GoogleNet Breakdown

Time

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

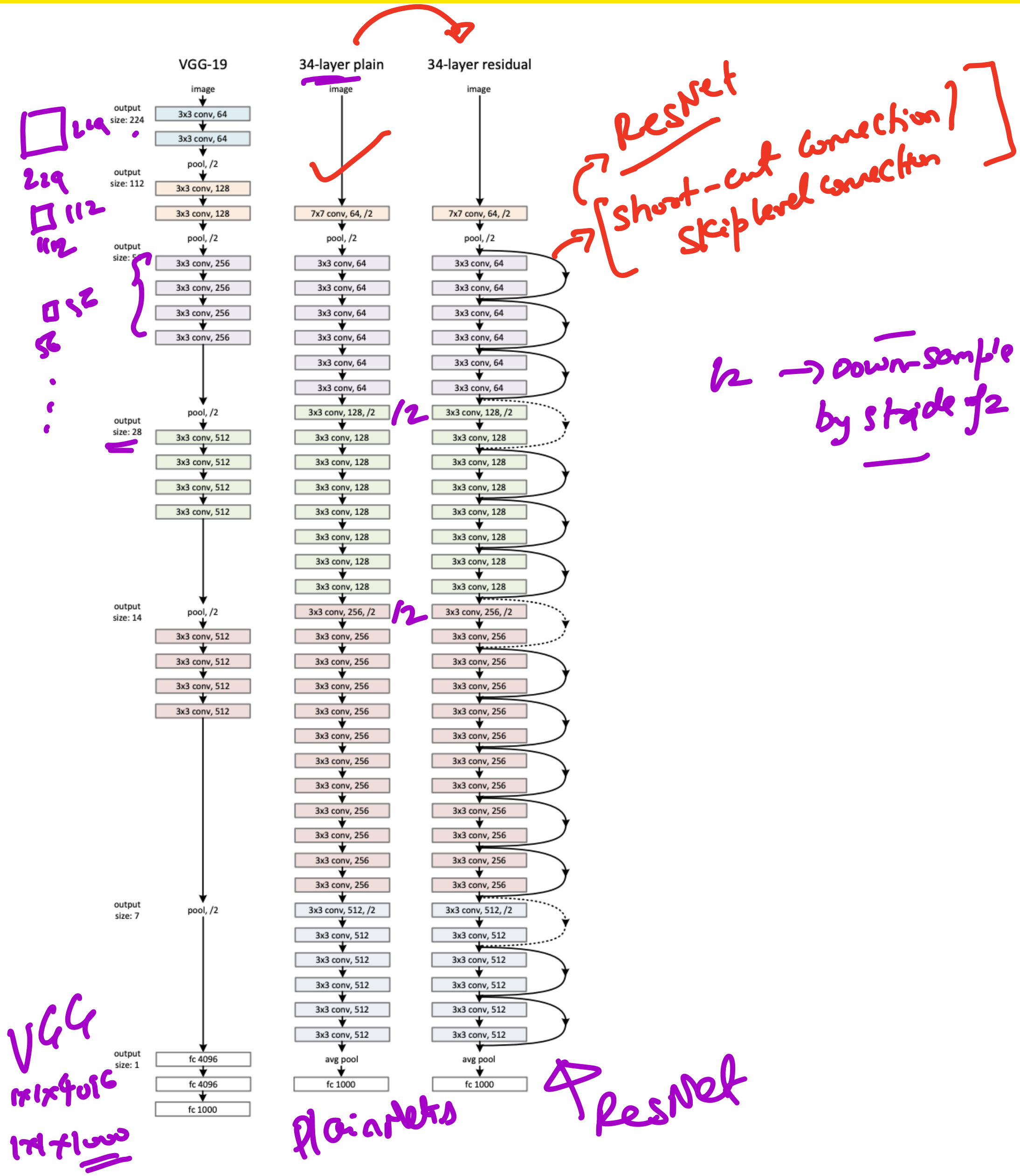
Popular CNN Architectures

Year	CNN	Developed By	Error Rates	No. of Parameters	Dataset
1998	LeNet	Yann LeCun		60 Thousand	ImageNet
2012	AlexNet	Alex Krizhevsky, Geoffrey Hinton and Ilya Sutskever	15.3 %	60 Million	
2013	ZFNet	Matthew Zeiler, Rob Fergus	14.8 %		
2014	GoogleNet	Google	6.67 %	4 Million	
2014	VGGNet	Simonyan, Zisserman	7.3 %	138 Million	
2015	ResNet	Kaiming He	3.6 %		

winner ILSVRC 2015

ResNet

ResNet Arch



ResNet ILSVRC paper

ResNet Motivation

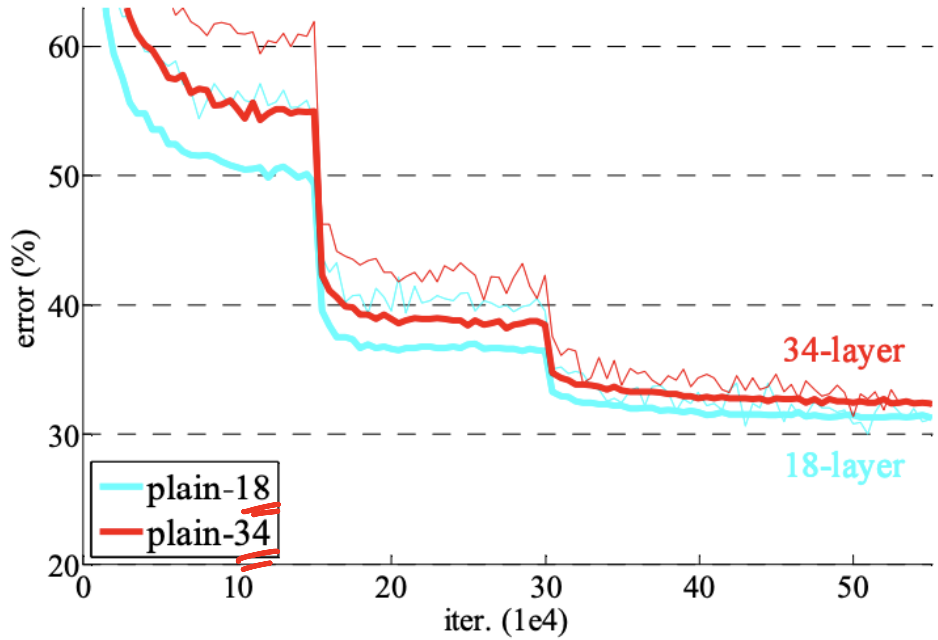
Res

Residue - Left over/

Difference

1. ResNet - Short for Residual Networks
2. Residual - Residue with respect to a reference
3. Ability to train “deeper networks” more effectively than plain nets

Plain Nets Degradation



Thin line
— Train Error

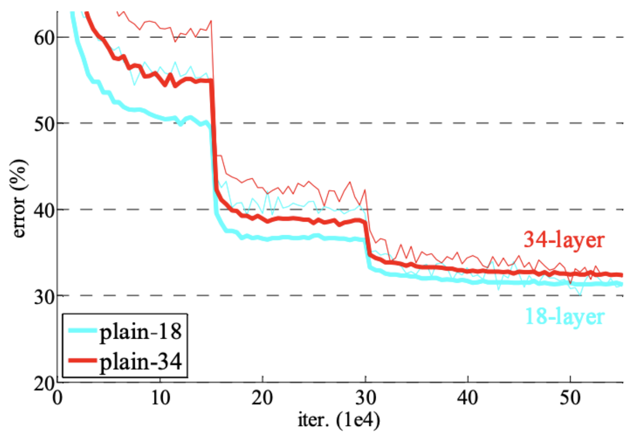
Thick line
— val Error

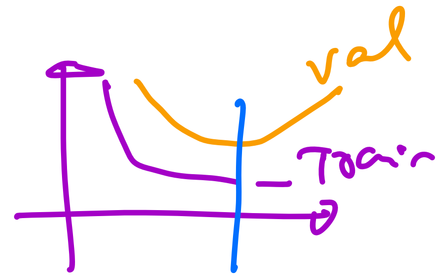
ResNet ILSVRC paper

ICE #1

Degradation or Over-fitting?

Thin Line - Train Error
ThickLine - val Error





The authors claim that the phenomenon we see above for plain networks is not over-fitting but a degradation in the network. What aspect of the graph hints at this?

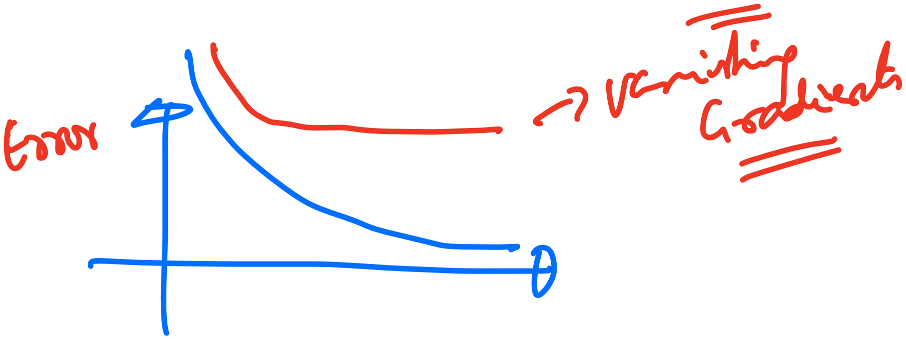
1. High train error for the 34 layer net vs the 18 layer } degradation
2. As the train error keeps going down, the validation error isn't going up at any point — No overfit!
3. Both a) and b)

d) None of the above

$0.1 \times 0.1 \times 0.1 \ldots \approx 0$ → Vanishing

$2 \times 2 \times 2 \ldots$ → Large → Exploding

1. *Vanishing or Exploding Gradients!* } DeepNet



RNN (Recurrent NN) → NLP

LSTM/Transformer

Issues with DeepNets

1. *Vanishing or Exploding Gradients!*
2. *Batch Normalization* - Normalization of layers ensures this doesn't happen (BN)

Issues with DeepNets

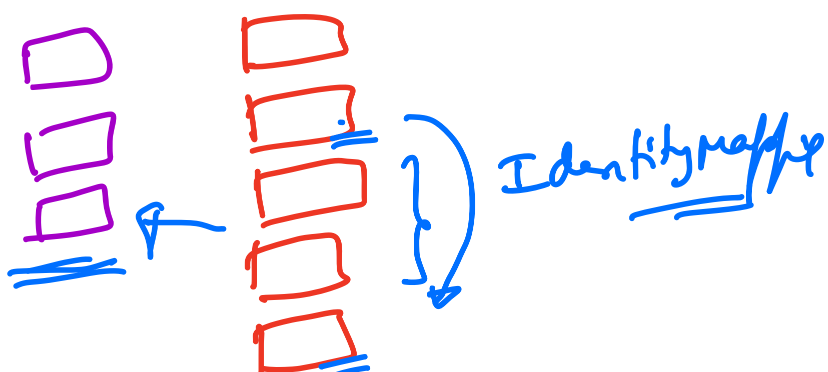
1. *Vanishing or Exploding Gradients!*
2. *Batch Normalization* - Normalization of layers ensures this doesn't happen
3. Despite Batch Normalization, authors saw a degradation with Plain Deep Nets

Issues with DeepNets

1. *Vanishing or Exploding Gradients!*
2. *Batch Normalization* - Normalization of layers ensures this doesn't happen
3. Despite Batch Normalization, authors saw a degradation with Plain Deep Nets
4. And this wasn't over-fitting!

Issues with DeepNets

1. *Vanishing or Exploding Gradients!*
2. *Batch Normalization* - Normalization of layers ensures this doesn't happen
3. Despite Batch Normalization, authors saw a degradation with Plain Deep Nets
4. And this wasn't over-fitting!
5. Ideally a DeeperNet should do at least as better as a shallow net if no over-fitting



ResNet vs Plain Nets

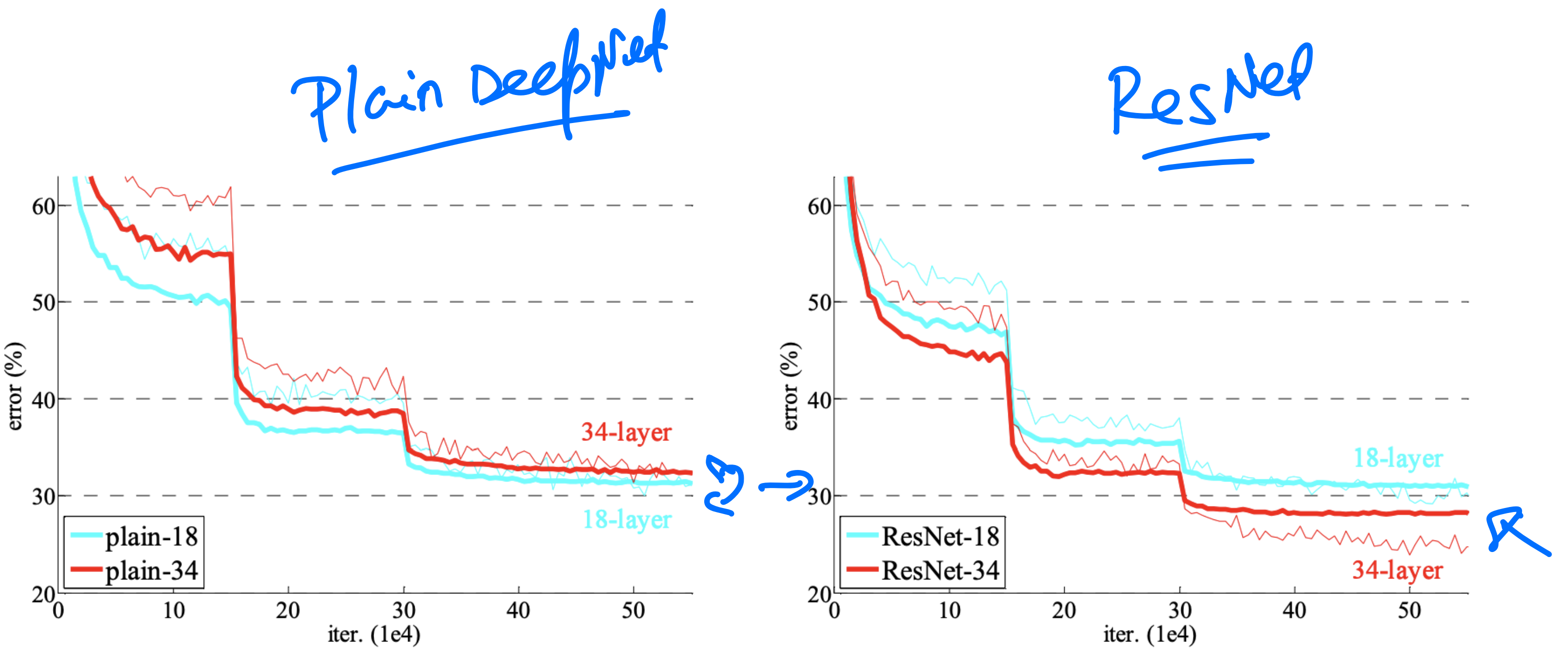
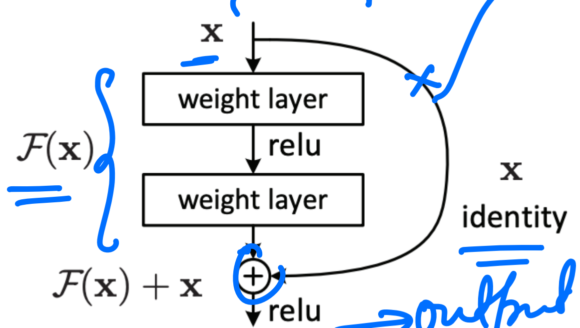


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

ResNet ILSVRC paper

shortcut connection

Input



output

Figure 2. Residual learning: a building block.

$\mathcal{F}(x)$ — Residue

x — Input

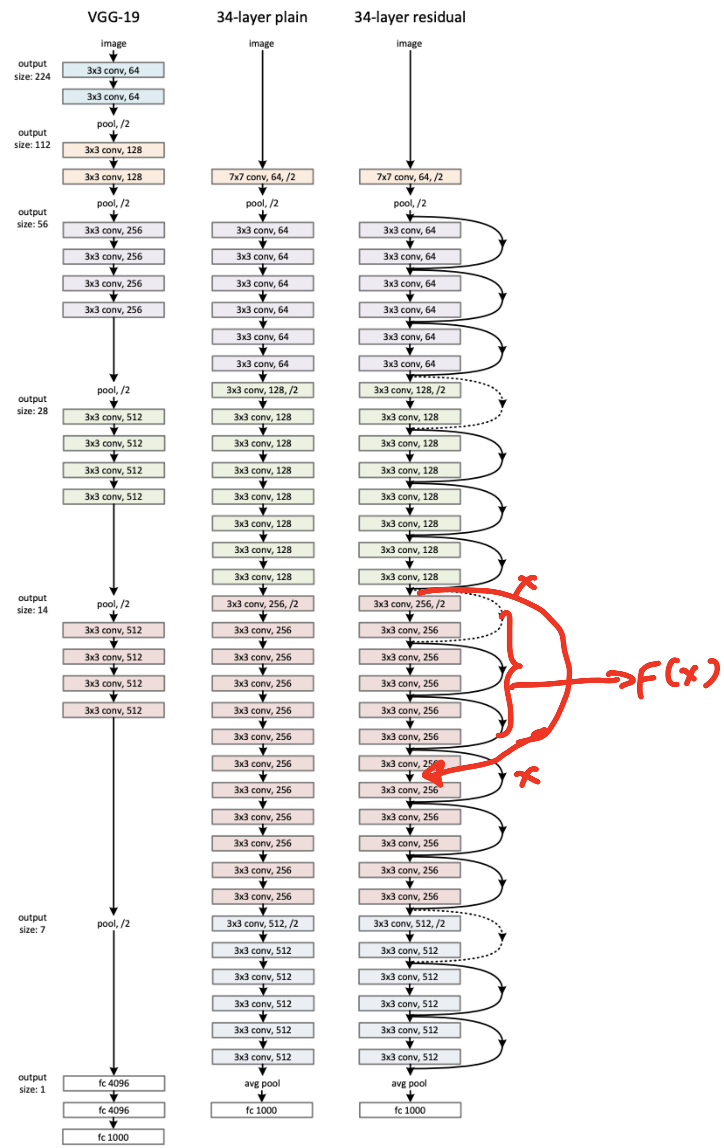
output $= x + \mathcal{F}(x)$

$H(x)$

Reference

Residue

ResNet ILSVRC paper



ResNet ILSVRC paper

ICE #2

ResNet Building Block

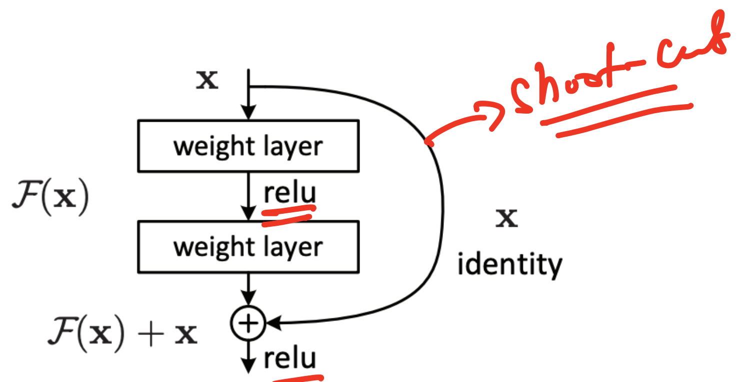


Figure 2. Residual learning: a building block.

Consider the ResNet building block as above. The only thing different from a plain-net is the short-cut connection. The output of this block is $F(x) + x$, where $F(x)$ refers to the "residual" from the Identity mapping x. If W_1, W_2 are the weights of the first and second layer and assume it's just a feedforward network and not a convNet layer and σ represents the non-linear RELU activation. How would you represent the output of this block?

ResNet ILSVRC paper

ICE #2

ResNet Building Block

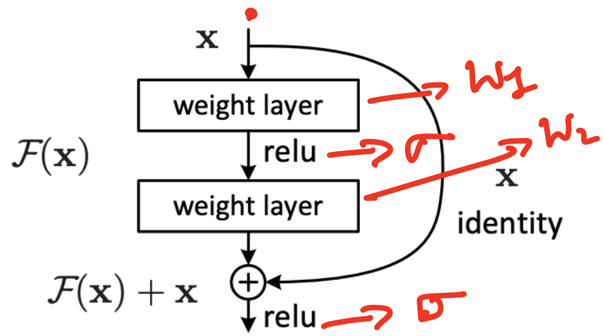


Figure 2. Residual learning: a building block.

1. $\sigma(W_2\sigma(W_1x)) + x$
2. $\sigma(W_2\sigma(W_1x) + x)$
3. $\sigma(W_1\sigma(W_2x)) + x$
4. $\sigma(W_1\sigma(W_2x) + x)$

$\mathcal{F}(x) = W_2\sigma(W_1(x))$

ResNet ILSVRC paper

ResNet Sizes

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1. Architectures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of blocks stacked. Down-sampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2.

ResNet ILSVRC paper

Resnet Results on Imagenet/Training

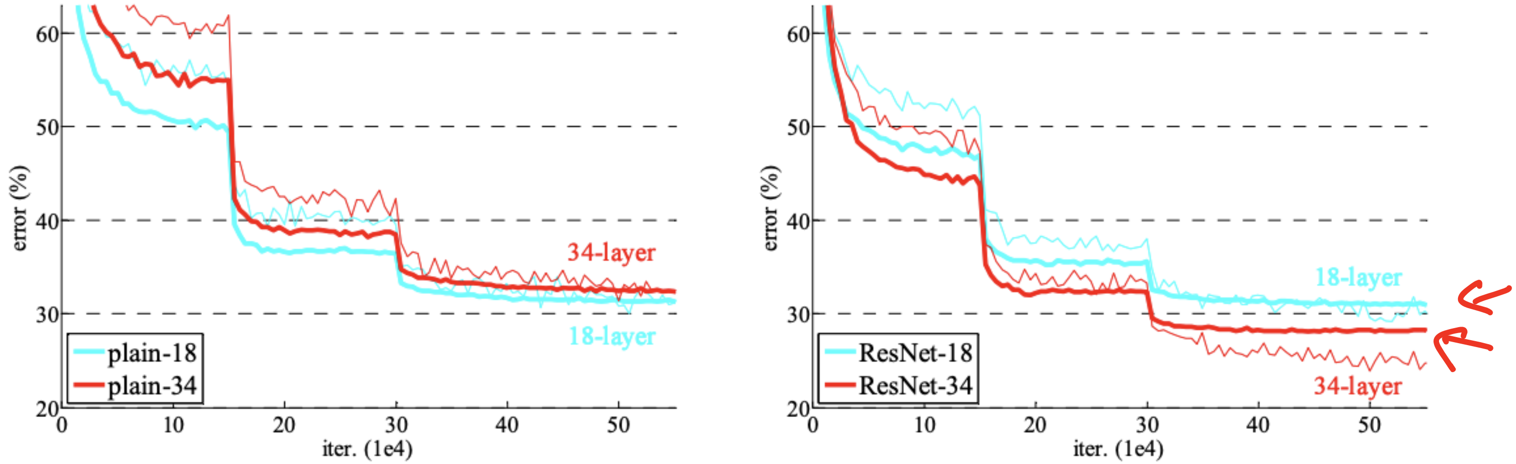


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

ResNet ILSVRC paper

Resnet Results on Imagenet/Validation

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except † reported on the test set).

ResNet

2

ResNet ILSVRC paper

Resnet Results on Imagenet/Test Set

method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Table 5. Error rates (%) of **ensembles**. The top-5 error is on the test set of ImageNet and reported by the test server.

ResNet ILSVRC paper

Resnet Results on CIFAR

#classes

CIFAR-10

60k

50k train

10k test

method	# layers	# params	error (%)
Maxout [10]			9.38
NIN [25]			8.81
DSN [24]			8.22
FitNet [35]	19	2.5M	8.39
Highway [42, 43]	19	2.3M	7.54 (7.72±0.16)
Highway [42, 43]	32	1.25M	8.80
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43** (6.61±0.16)
ResNet	1202	19.4M	7.93

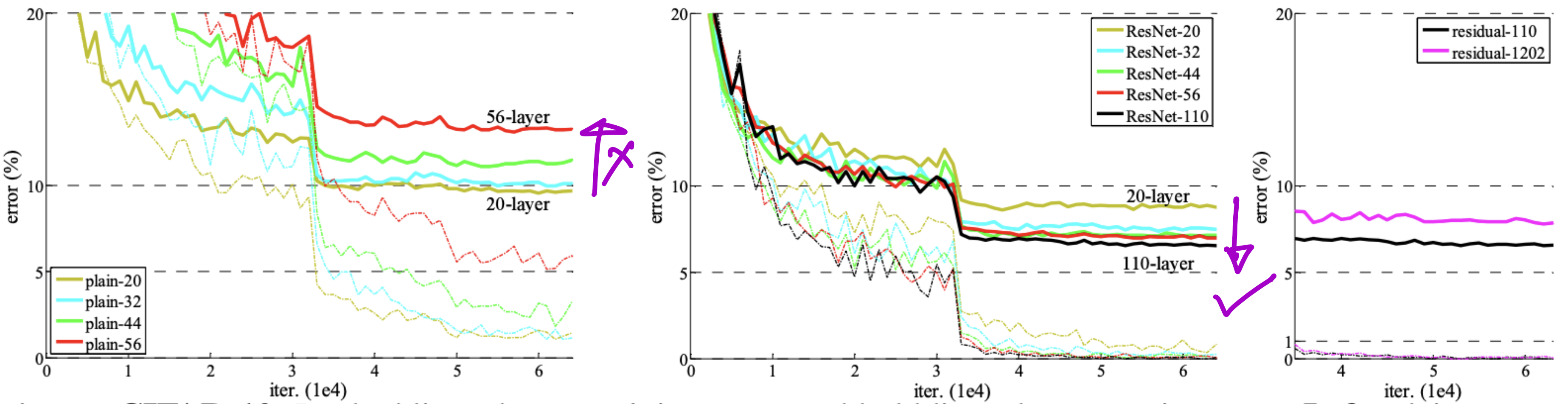
??

Table 6. Classification error on the **CIFAR-10** test set. All methods are with data augmentation. For ResNet-110, we run it 5 times and show "best (mean±std)" as in [43].

ResNet ILSVRC paper

Resnet Results on CIFAR

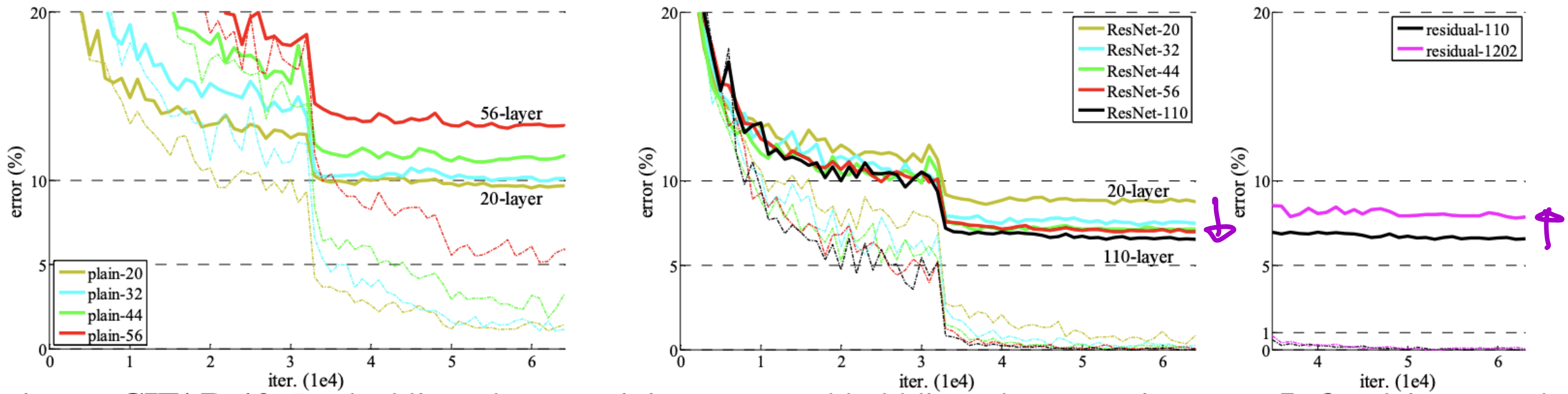
Generalizability



Training on CIFAR-10. Dashed lines denote training error, and bold lines denote testing error. Left: plain networks. The error of plain-110 is higher than 60

ResNet ILSVRC paper

ICE #3



What's going on?

What's going on with the right most figure? The 1000 layer ResNet actually has a worse validation error than the 100 layer ResNet. What's the likely explanation for this?

1. Degradation
2. Overfitting
3. Optimization issues
4. All of the above

Next Lecture

1. Pre-Training CV models
2. Object Detection and Image Segmentation