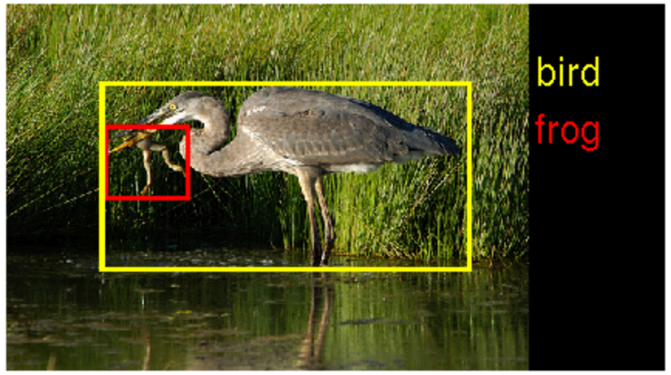
Depth Matters; VGGNet

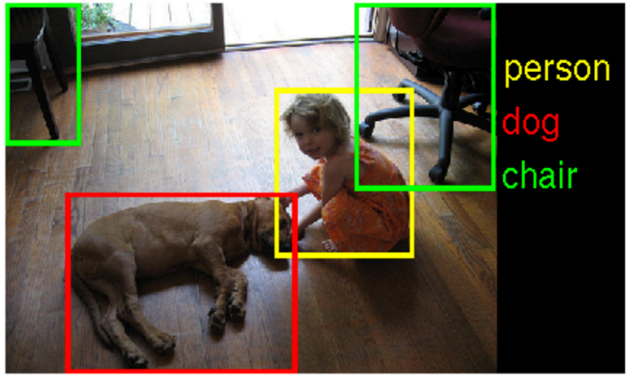
Gyubin Son

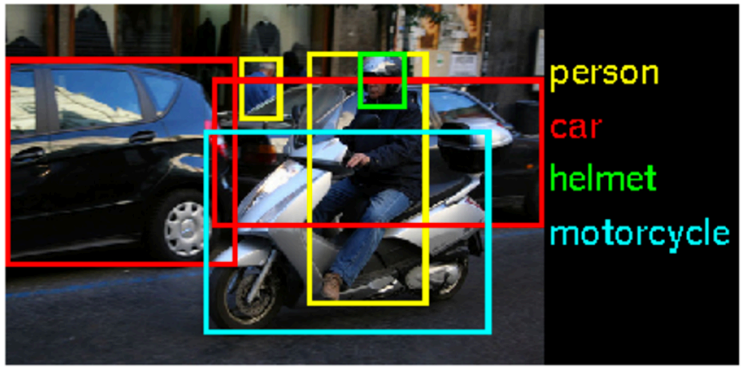
Index

Intro - Configuration - Train - Test

Intro - IMAGENET







1K Class

15M train images

22K categories

Intro - VGGNet

- Visual Geometry Group(**VGG**) of Oxford
- Result in ILSVRC 2014
  - 1st in Localization (25.3% error)
  - 2nd in Classification (7.3% error)
  - Better than GoogLeNet in Single network (7.1% vs 7.9%)

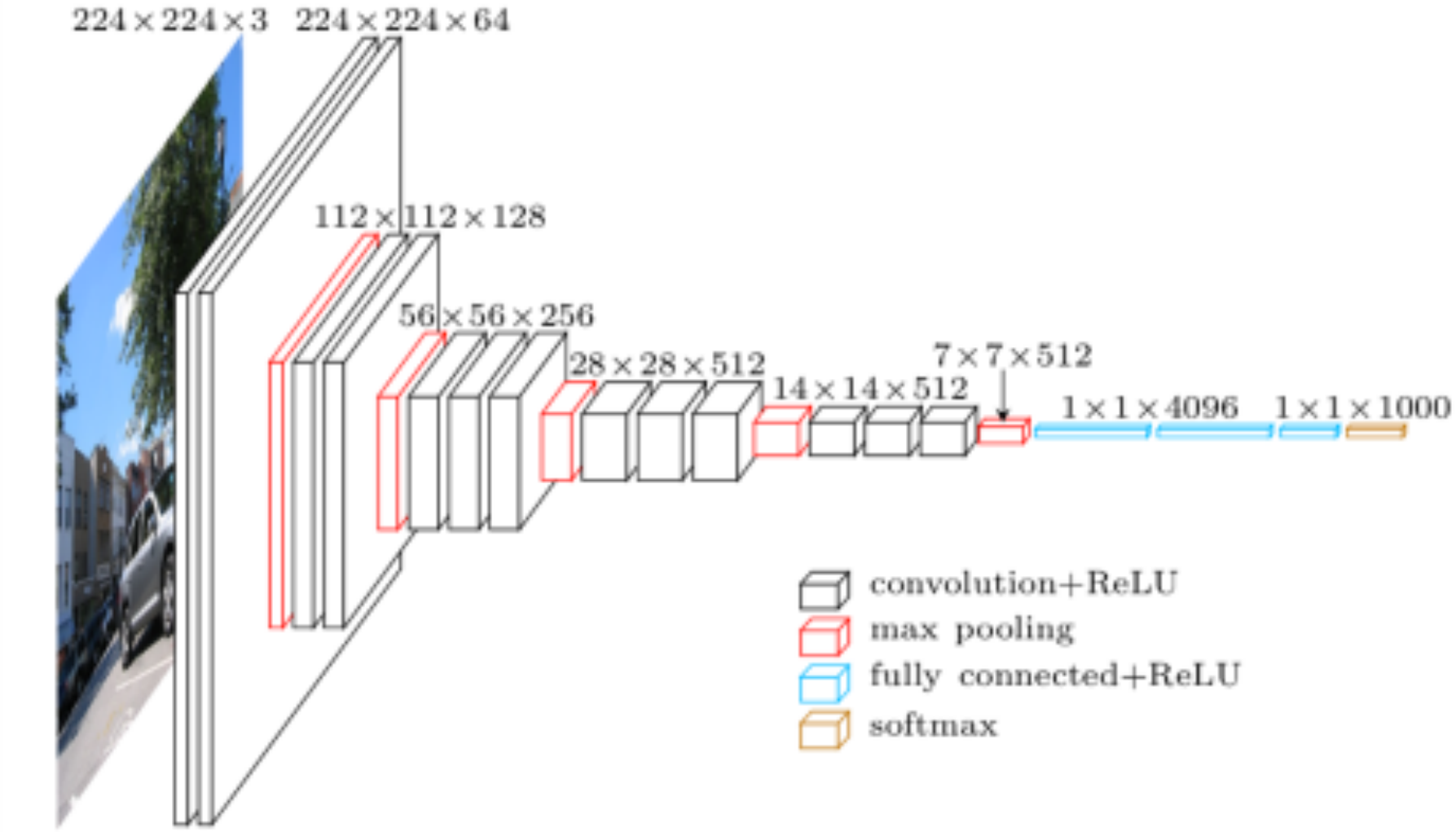
Intro - Goal

- CNN에서 **depth**가 어떤 영향을 미치는지 연구
  - 다른 parameter는 모두 fix
  - depth만 늘려가며 비교 (Layer 11, 13, 16, 19)
- depth를 늘리기 위해 **small filter** 사용
  - 3 x 3 filter

Index

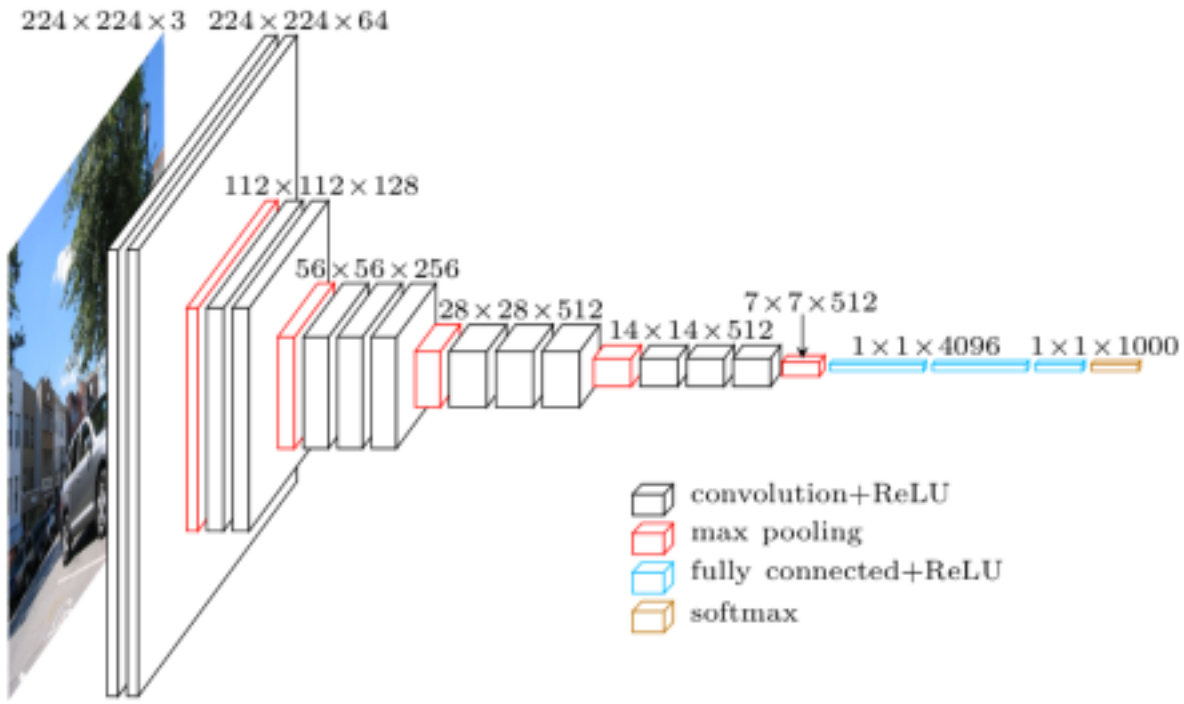
Intro - **Configuration** - Train - Test

Config - Architecture



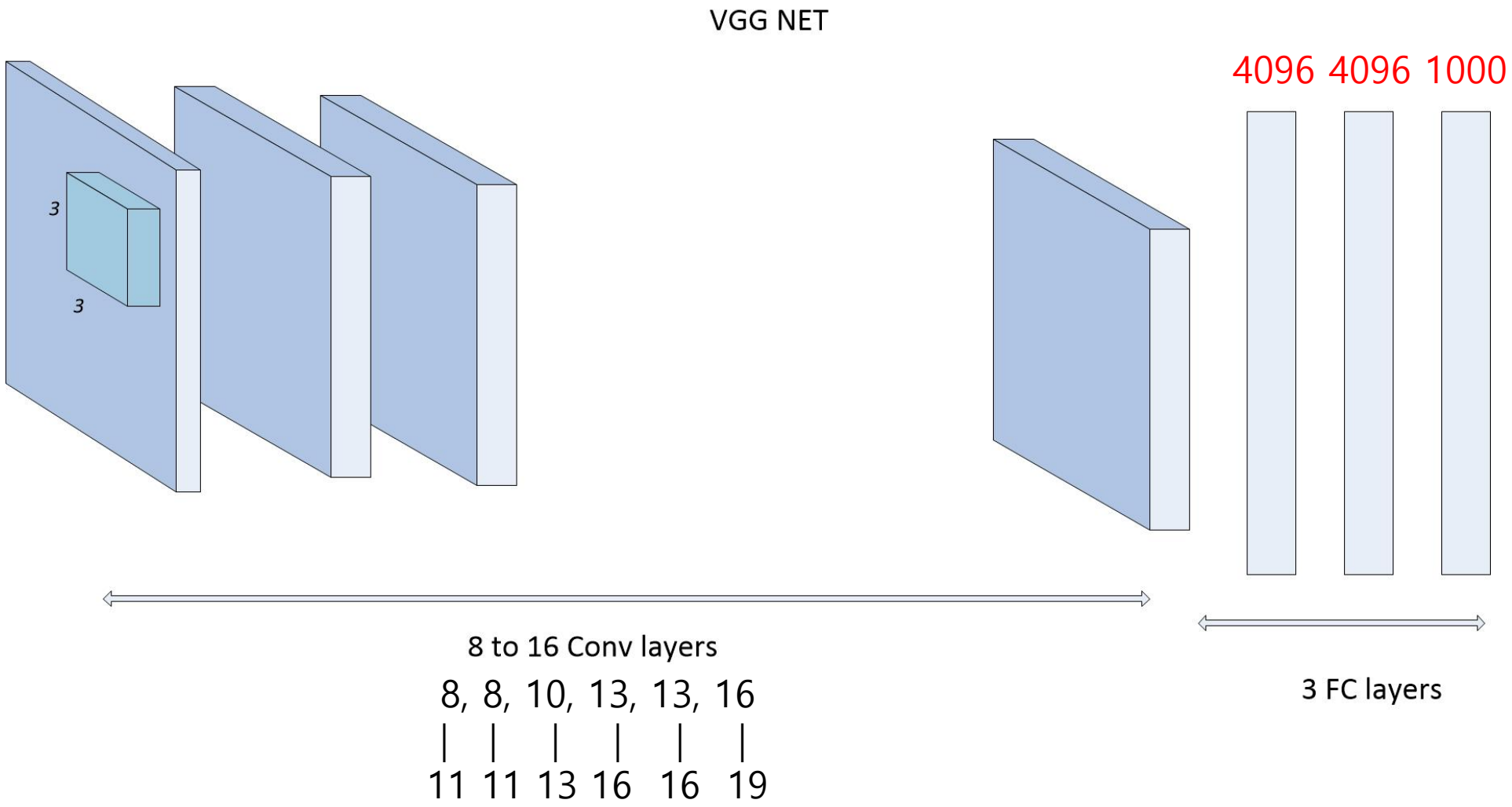
Config - Fixed configuration

- Fixed things
  - Image size: 224 x 224 x 3
  - Stride: 1 -> Preserve resolution
  - Padding: 1 -> Preserve resolution
  - Max-Pooling: 2x2 window, 2 stride
  - Filter sizes: 3x3, 1x1
- Layers
  - Convolutional Layers: 8 to 16
  - Fully Connected Layers: 3
  - ReLU: Follow all hidden layers
  - Convolutional filters: Starting from 64, double after each max-pooling layer until 512



No LRN(Local Response Normalization

Config - Architecture



Config - Table

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Config - Filter

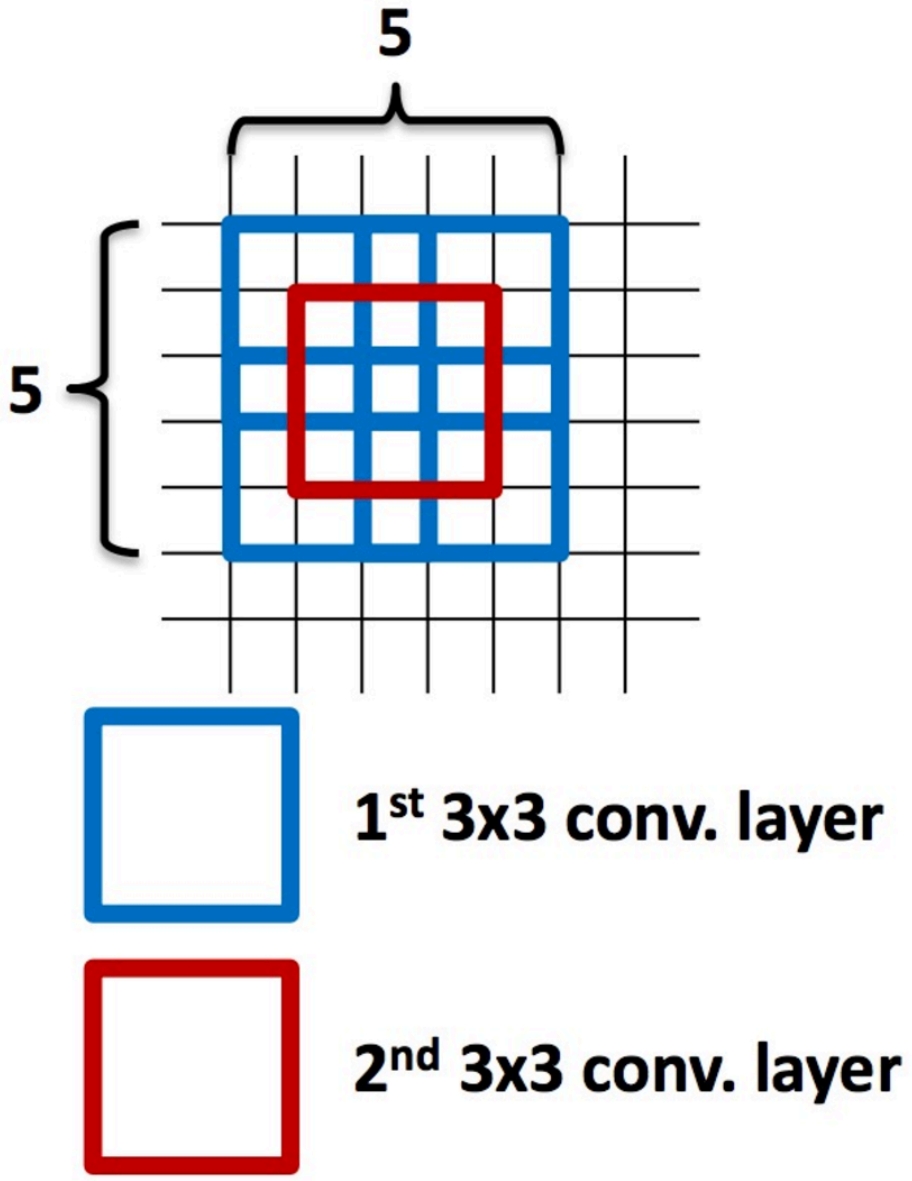
- **To enable deeper architectures**
- **Minimum size** required for learning concepts of horizontal, vertical, blob.
- **Less parameters** for same receptive field -> **Regularization**
  - Stack of 3 filters(3x3) = 1 filter(7x7)
  - Number of Parameters
    - 1 layer of 3x3 filter : #params = 3^2C^2 ; grows as $O(n^2)$
    - 3 layers of 3x3 filters : **#params = $3 * (3^2C^2) = 27C^2$**
    - 1 layer of 7x7 filter : **#params = $7^2C^2 = 49C^2$**
- **More ReLUs** -> More discriminative

0	-1	0
-1	4	-1
0	-1	0

Config - Filter sizes matters

- Why 3x3 layers?
- **Stacked conv. layers**

  ~= large receptive field
- Almost same
  - two 3x3 layers - 5x5 receptive field
  - three 3x3 layers - 7x7 receptive field



images from http://www.robots.ox.ac.uk/~karen/pdf/ILSVRC_2014.pdf

Config - 1x1 filter in C network

- **Increase non-linearity** without affecting receptive field
- When input channels == output channels
  - Projection onto space of same dimension
- Another perspective: Fully connected with weight sharing
- Used in Network In Network and GoogLeNet

Index

Intro - Configuration - **Train** - Test

Train - Configuration

- Multinomial logistic regression
- Mini-batch gradient descent : batch size 256
- Momentum: 0.9
- L2 Regularizer: $5 * 10^{-4}$
- Dropout only for the first 2 FC layers: 0.5
- **Learning rate**
  - Starting from **0.01**
  - **Decrease by factor of 10** when validation accuracy stops improving
- **Earlier convergence** than AlexNet: Regularization(3x3 filter), Pretrained W

Train - Weight initialization

- **Random initializations** for Configuration A
  - A is shallow enough to be trained as random
  - 11 Layers: 8 Conv, 3 FC
- Network B, C, D, E
  - **Use pretrained W of A** : First 4 conv layers, last 3 FC layers
  - Other layers are randomly initialized from $N(0, 0.01)$
- Bias initialized to 0

Train - Image preprocessing

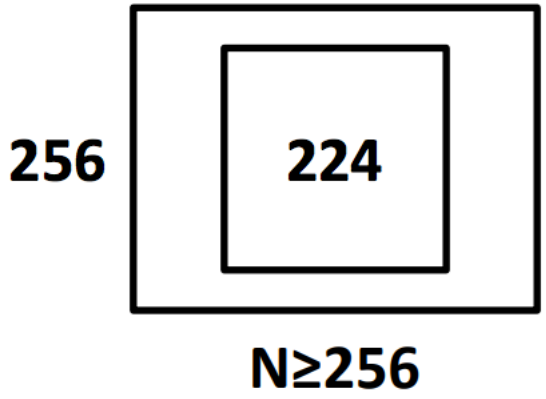
- **Fixed 224x224 size**

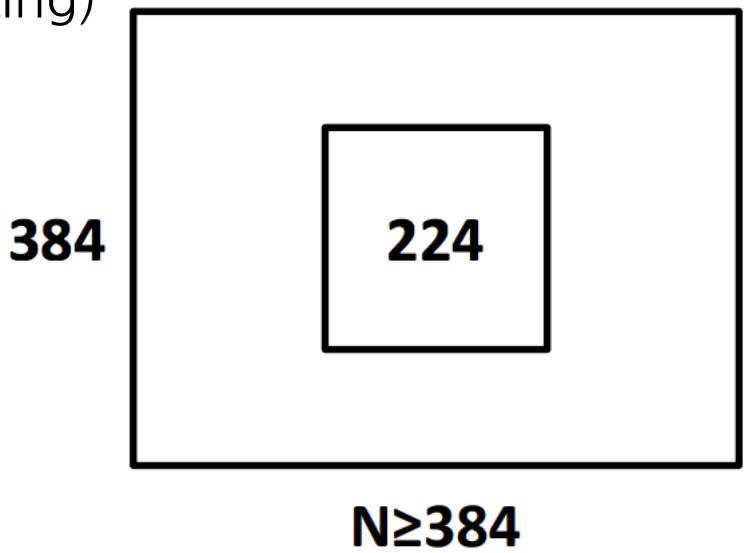
  (Randomly cropped from isotropically rescaled images)

- **Data augmentation**

  Similar with AlexNet (Rondom horizontal flipping, RGB shifting)

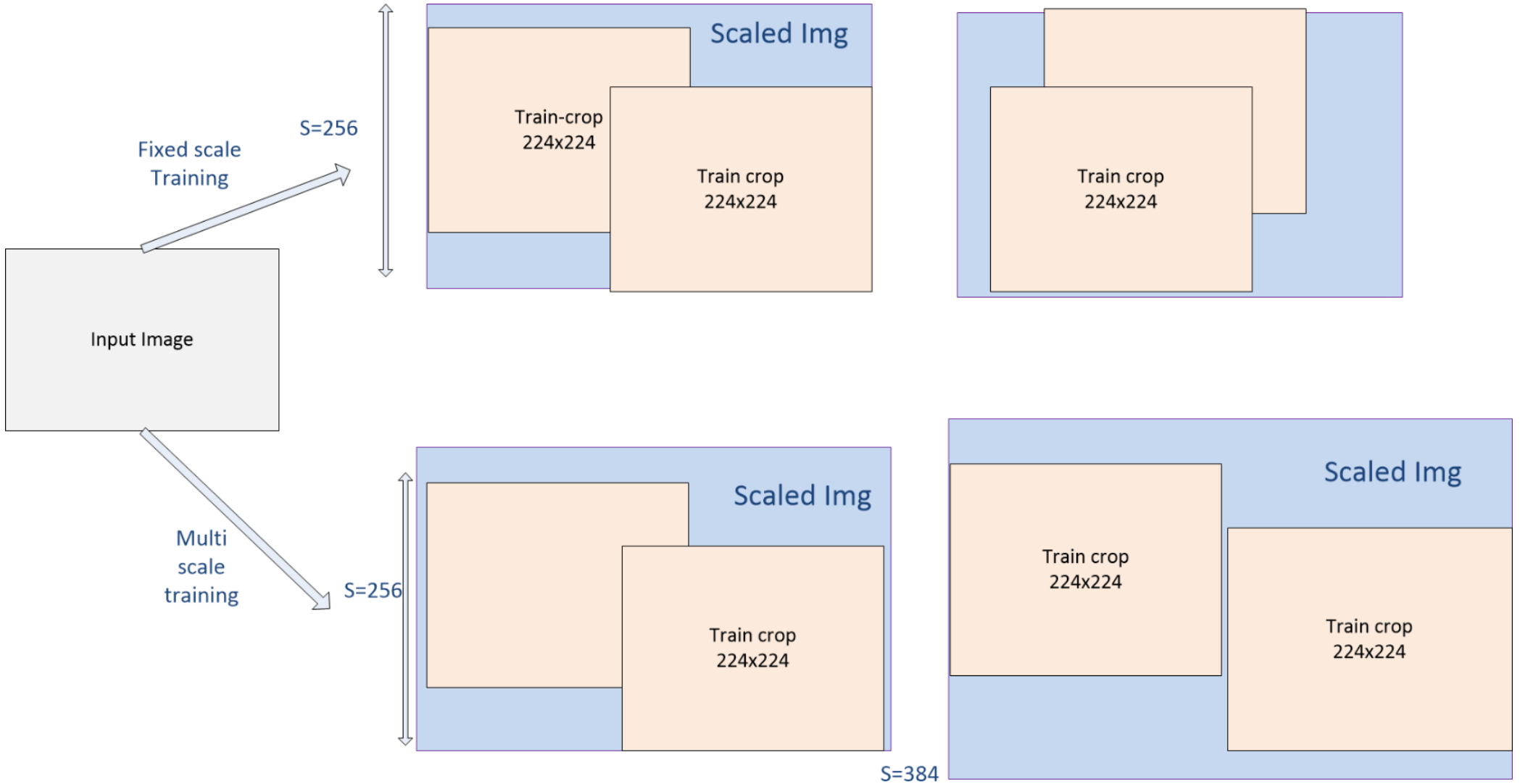
- **Image Scaling :** Let shortest dimension is **S**
  - **Single-scale** : S= 256, S=384(finetunning)
    - Random initialize at “256”
    - “384” uses pretrained weights of “256”
  - **Multi-scale (Scale Jittering)**
    - Sample S randomly in range [256, 512]





images from http://www.robots.ox.ac.uk/~karen/pdf/ILSVRC_2014.pdf

Train - Image scaling



Index

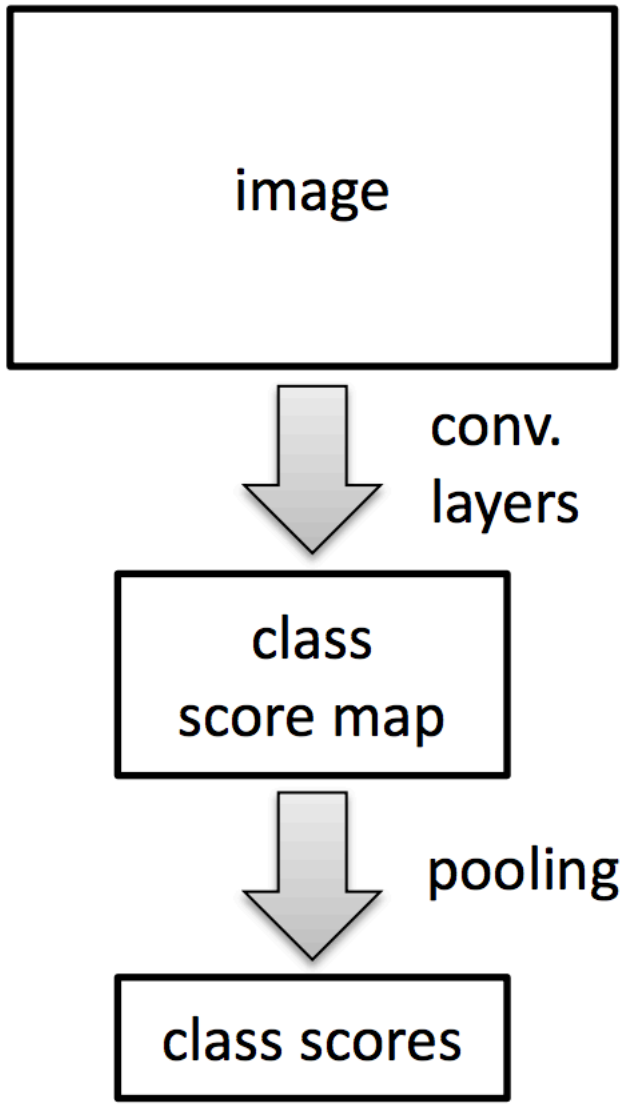
Intro - Configuration - Train - **Test**

Test - Configuration

- **Testing Scale: Q**
  - Each test image scaled isotropically smallest side is Q
- **Scaling**
  - Single scale evaluation: Q = S
  - Multiscale evaluation:Try different Qs for a single S
    - Multi-crop evaluation
    - Dense Evaluation

Test - Dense evaluation

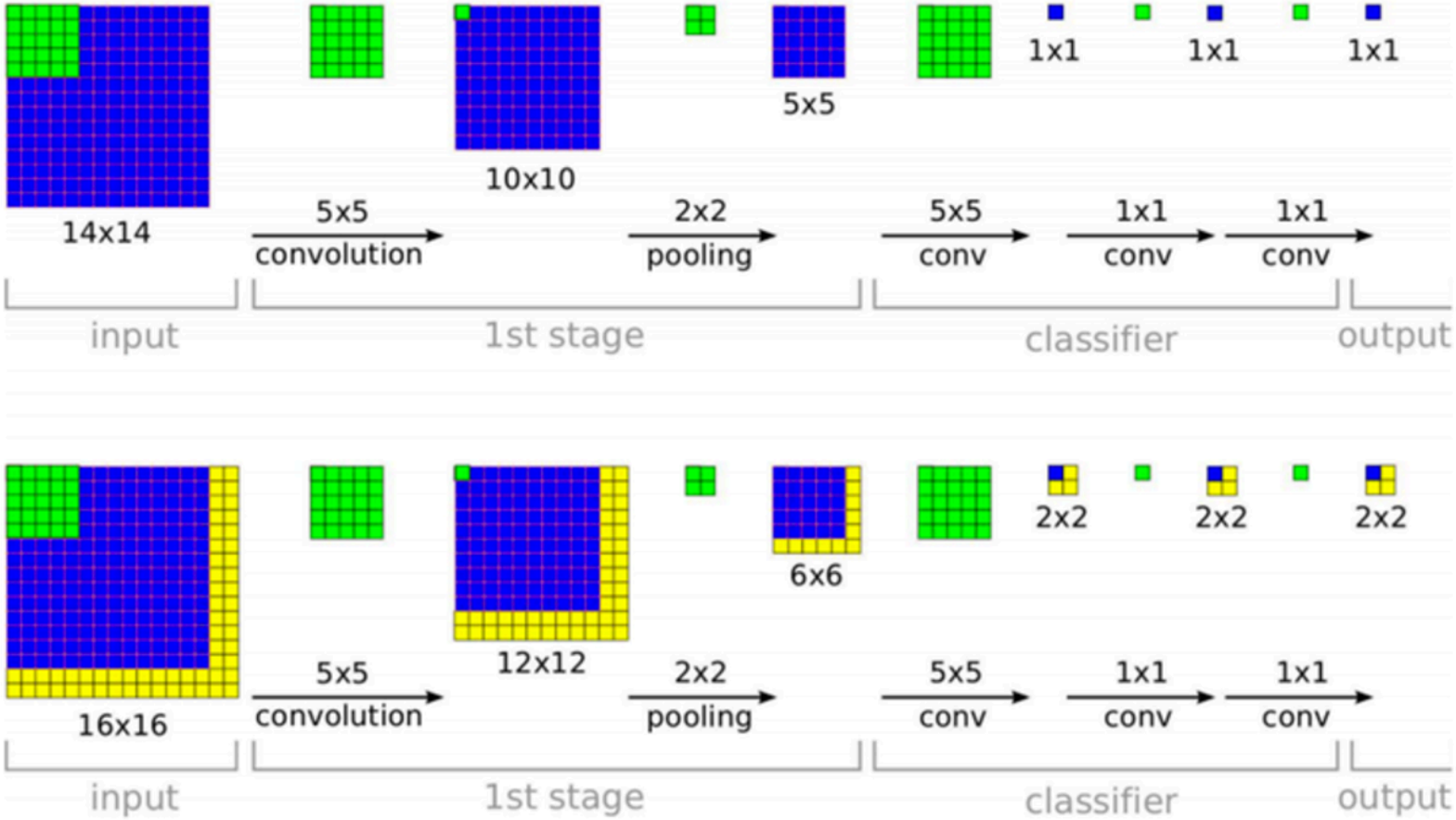
- No need to CROP, **Use whole images** as input.
- Any input dimensions can be changed to 1000 class softmax.
- Applying steps
  - Densely apply the network over the images
  - FC layers converted to convolutional layers
  - FC-1000: 4096x1000 params into 1000 filters size 1x1x4096
  - Sum-pooling of 1000 class score maps.
- **More efficient than applying the multiple crops**



images from http://www.robots.ox.ac.uk/~karen/pdf/ILSVRC_2014.pdf

Test - Dense evaluation

- Sliding windows
- Bigger filter than training filter
- Overcome oversize images.



images from 라온피플(주) 블로그

Test - Single scale evaluation

- Q=S for fixed scale training
- Q=0.5(Smin+Smax) for jittered scale

Table 3: **ConvNet performance at a single test scale.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
A	256	256	29.6	10.4
A-LRN	256	256	29.7	10.5
B	256	256	28.7	9.9
C	256	256	28.1	9.4
	384	384	28.1	9.3
	[256;512]	384	27.3	8.8
D	256	256	27.0	8.8
	384	384	26.8	8.7
	[256;512]	384	25.6	8.1
E	256	256	27.3	9.0
	384	384	26.9	8.7
	[256;512]	384	**25.5**	**8.0**

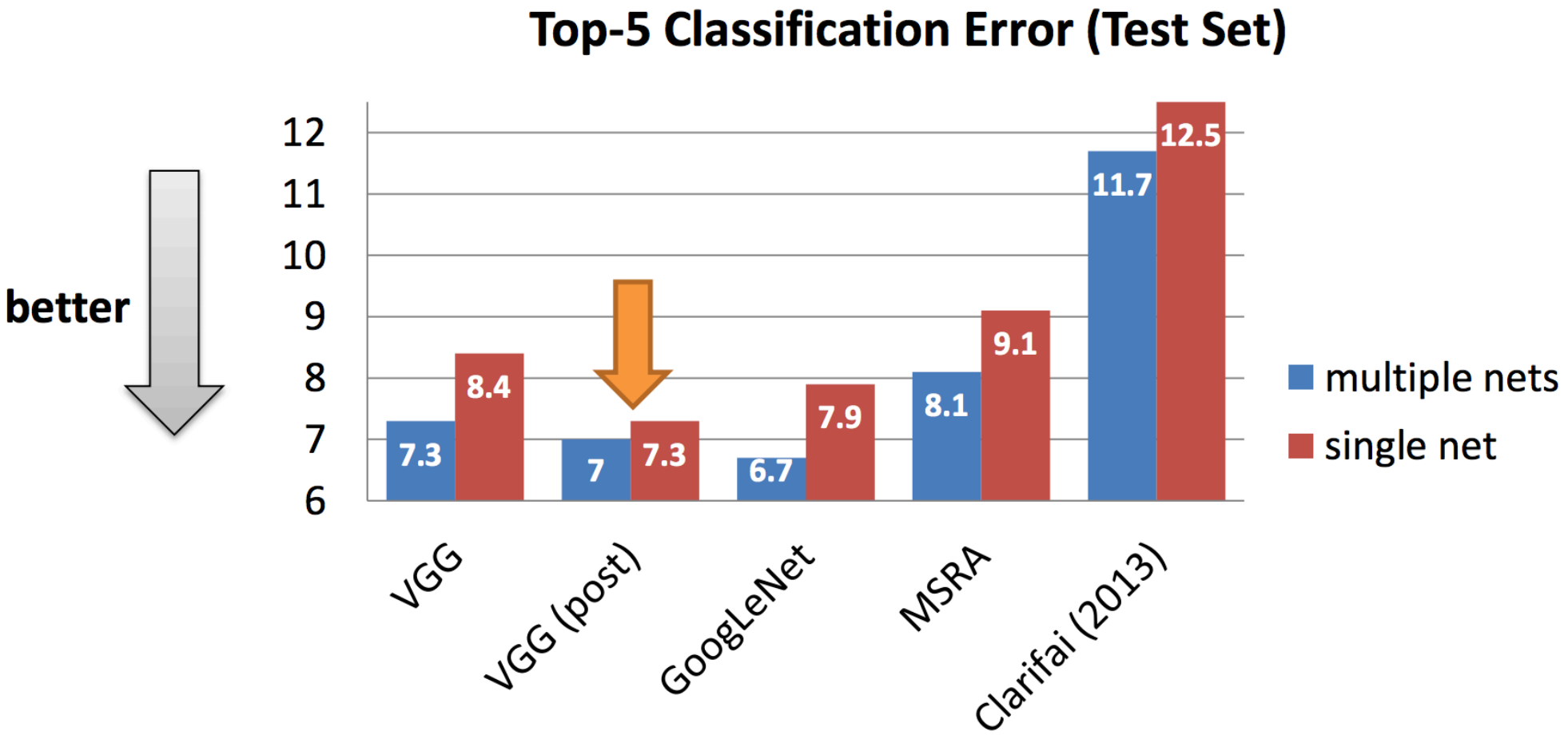
Test - Multi scale evaluation

- Scale jittering at test time
- Fixed S: Q= {S-32, S, S+32}
- S [Smin, Smax] : Q = {Smin, 0.5(Smin,Smax), Smax}
- Multi crop : 50 crops per scale(5x5 grid and 2 flips). 150 crops 3 scales

Table 4: **ConvNet performance at multiple test scales.**

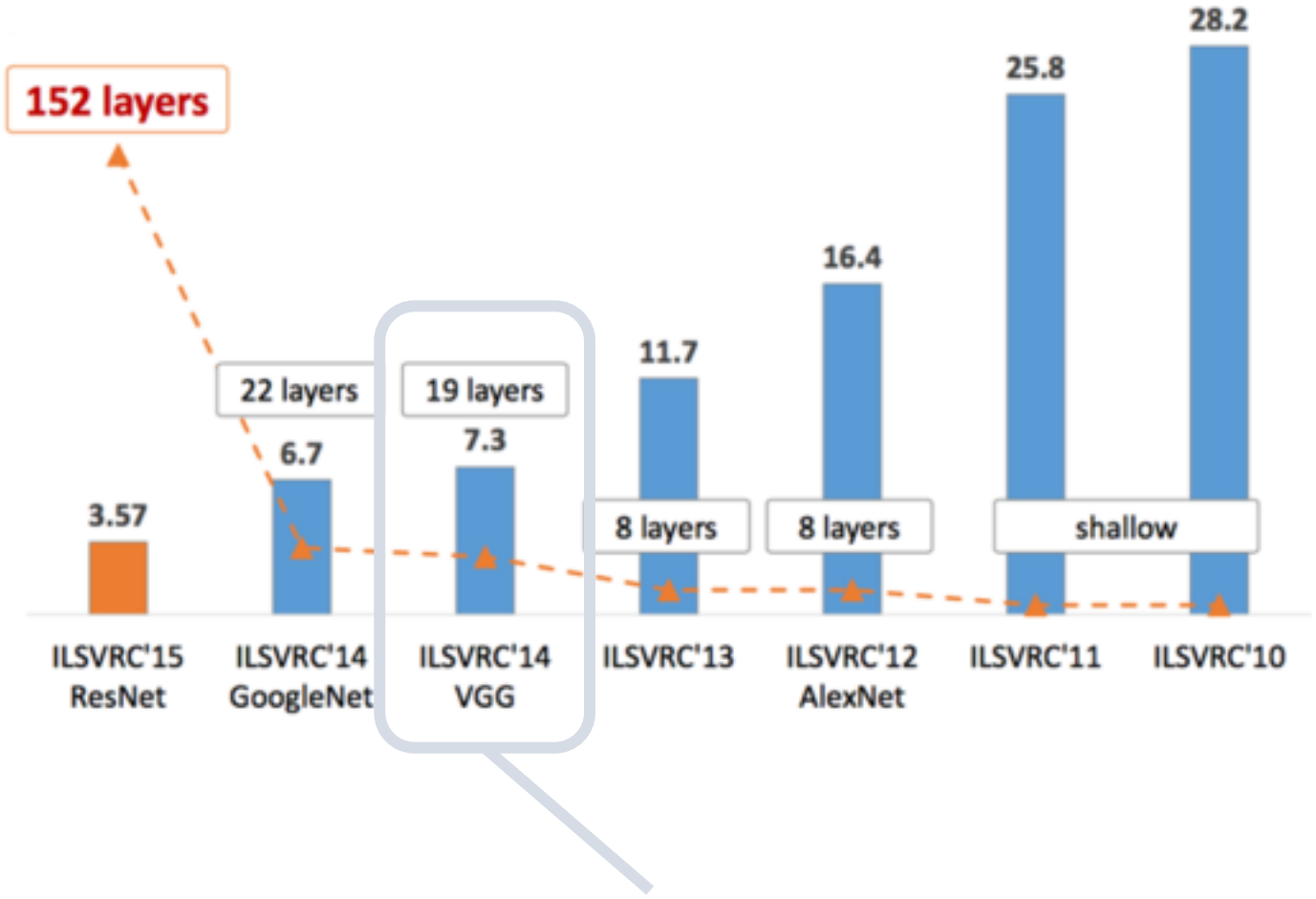
ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train (S)	test (Q)		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	**24.8**	**7.5**

Test - Best performance



Conclusion

- Classification error decreases with increases **ConvNet depth**
- Important to **capture more spatial context** : D(3x3) vs C(1x1)
- **DeepNets with small filters outperform shallow networks with large filters**
- Shallow version of B
  - 2 layers of 3x3 replaced with single 5x5
  - performs worse
- Error rate saturated at **19 layers**
- **Scale jittering** at training helps capturing multiscale statistics
  and leads to better performance


152 layers
22 layers
19 layers
8 layers
8 layers
shallow
3.57
6.7
7.3
11.7
16.4
25.8
28.2
ILSVRC'15 ResNet
ILSVRC'14 GoogleNet
ILSVRC'14 VGG
ILSVRC'13
ILSVRC'12 AlexNet
ILSVRC'11
ILSVRC'10

Thank you.