VGG Net

DEEP LEARNING PAPER PRESENTATION

GROUP 3

Index

- Preface
- Paper Review
  - Introduction
  - ConvNet Configurations
  - Classification Framework
  - Classification Experiments
- Conclusion

Preface

- "Very Deep Convolutional Networks for Large-Scale Image Recognition"
- VGG: Visual Geometry Group, Department of Engineering Science, University of Oxford.
- Karen Simonyan & Andrew Zisserman
- ICLR (International Conference on Learning Representation) 2015

Preface

- Krizhevsky et al. (2012): AlexNet
- Zeiler and Fergus (2013): ZFNet
- Sermanet et al. (2014): OverFeat
- Szegedy et al. (2014): Inception

Paper Review: Introduction

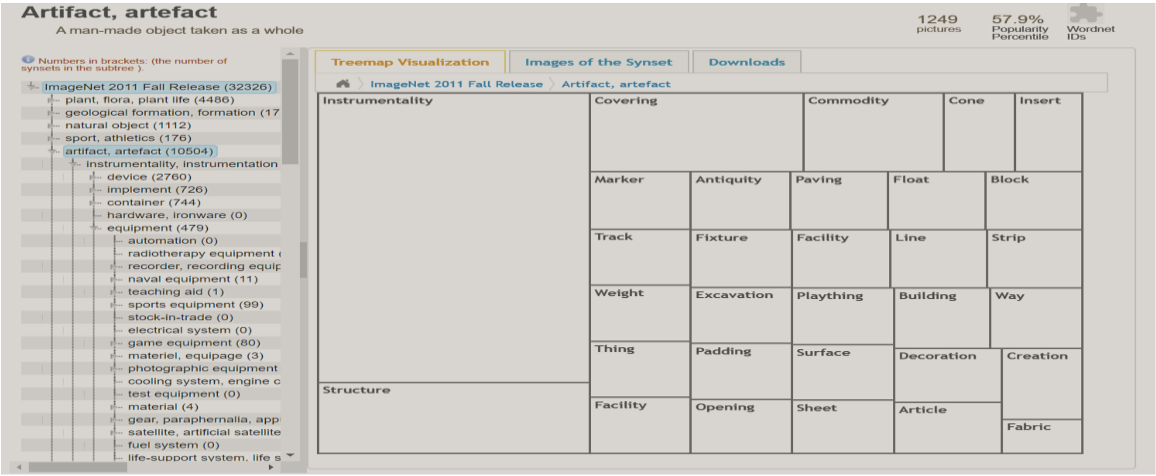
Introduction

- ConvNets rise in image and video recognition:
  - Large public databases: ImageNet
  - High-performance computing: GPUs

- ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) as testbed for image classification systems

ImageNet

- 14 million images
- 1 million images with bounding boxes annotations



Summary and Statistics (updated on April 30, 2010)

Overall

- Total number of non-empty synsets: 21841
- Total number of images: 14,197,122
- Number of images with bounding box annotations: 1,034,908
- Number of synsets with SIFT features: 1000
- Number of images with SIFT features: 1.2 million

Statistics of high level categories

High level category	# synset (subcategories)	Avg # images per synset	Total # images
amphibian	94	591	56K
animal	3822	732	2799K
appliance	51	1164	59K
bird	856	949	812K
covering	946	819	774K
device	2385	675	1610K
fabric	262	690	181K
fish	566	494	280K
flower	462	735	339K
food	1495	670	1001K

Introduction

- Previous improvements over AlexNet:
  - Smaller receptive window size and stride (OverFeat, ZFNet)
  - Training and testing over whole image on multiple scales (OverFeat)
- This paper improvement: network depth

ConvNet Configurations

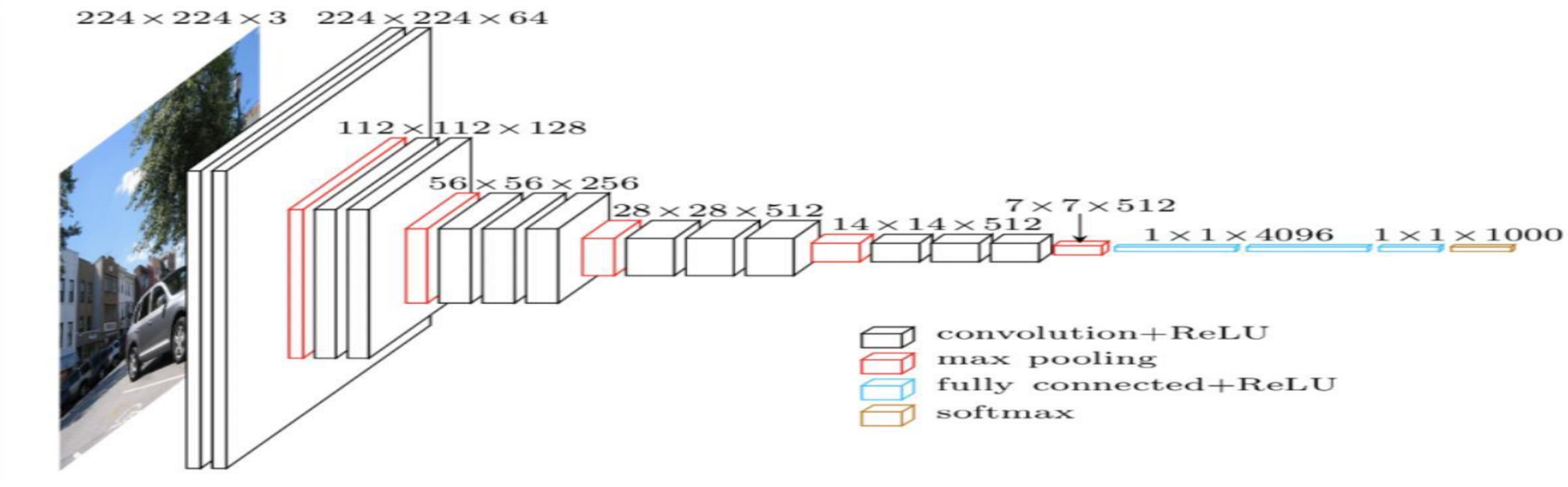
Architecture

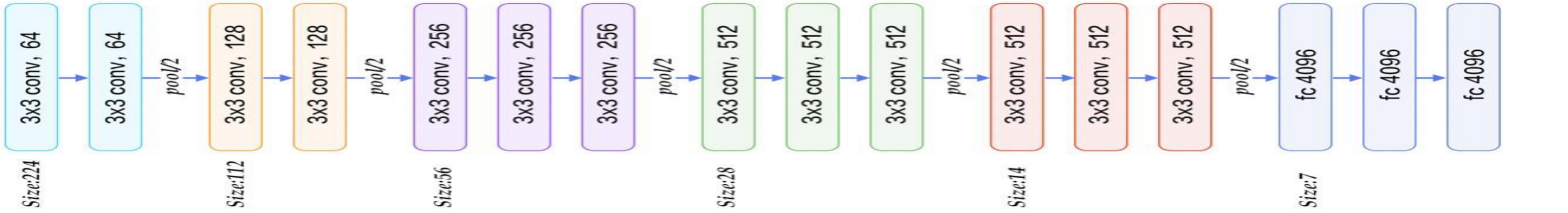
- Input: 224x224 RGB image
- Preprocessing: subtracting mean RGB value, computed on the training set, from each pixel
- Output: probability for each of the 1000 classes
- ReLU activation function on hidden layers

Architecture

- Building blocks:
  - Convolutional layer: 3x3 with 1-pixel padding or 1x1 filter, both with 1-pixel stride
  - Max-pool layer: 2x2 with 2-pixel stride
  - Fully-connected layers: 4096 channels and 1000 channels
  - Soft-max layer

Architecture





Configurations

- Same building blocks as stated in previous slides
- Only differ in number of conv. layers
- Width of conv. layers (no. of channels) starts

  from 64 and increases by 2 until reaching 512

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					

Configurations

- Less weights than in a more shallow net with larger conv. layer widths and receptive fields (144M weights in OverFeat)

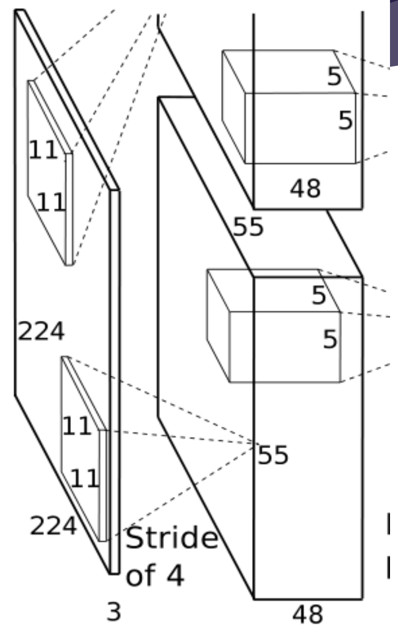
Table 2: **Number of parameters** (in millions).

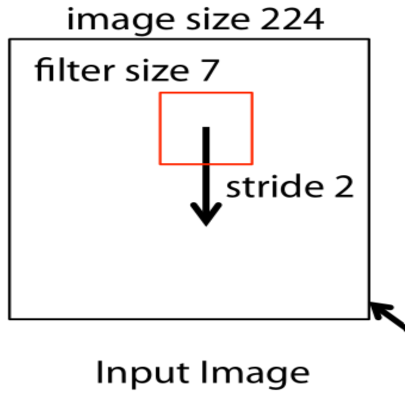
Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Layer	1	2	3	4	5	6	7	Output 8
Stage	conv + max	conv + max	conv	conv	conv + max	full	full	full
# channels	96	256	512	1024	1024	3072	4096	1000
Filter size	11x11	5x5	3x3	3x3	3x3	-	-	-
Conv. stride	4x4	1x1	1x1	1x1	1x1	-	-	-
Pooling size	2x2	2x2	-	-	2x2	-	-	-
Pooling stride	2x2	2x2	-	-	2x2	-	-	-
Zero-Padding size	-	-	1x1x1x1	1x1x1x1	1x1x1x1	-	-	-
Spatial input size	231x231	24x24	12x12	12x12	12x12	6x6	1x1	1x1

Discussion

- VGG quite different from previous top-performers:
  - ILSVRC-2012: 11x11 receptive field with stride 4 in AlexNet
  - ILSVRC-2013 : 7x7 receptive field with stride 2 in ZFNet and same as AlexNet and OverFeat



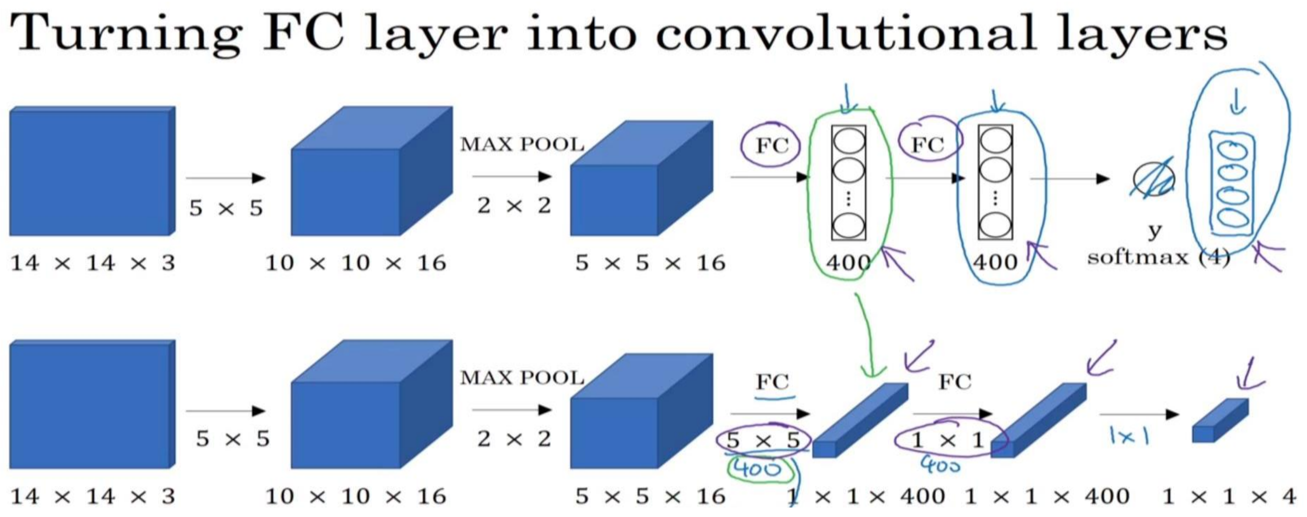


Discussion

- VGG uses very small 3 x 3 receptive fields with stride 1:
  - Stacks of layers with smaller fields have same effective receptive field as bigger fields
- Benefits:
  - More non-linear rectification layers => More discriminative decision function
  - Decrease the number of parameters:
    - Three 3x3 conv. layers: 3(3^2*C^2) = 27 C^2 weights
    - 7x7 conv. layer: 7^2*C^2 = 49 C^2 parameters
    - 81% less for 3x3 vs 7x7

Discussion

- Benefits of 1 × 1 conv. layers:
  - Increase non-linearity of decision function
  - Does not affect receptive field of conv. layers



Discussion

- Lin et al. (2014):
  - 1x1 convolutional filters in “Network in Network” architecture
- Ciresan et al. (2011):
  - Used small-size convolution filters
  - Significantly less deep nets.
  - Did not evaluate on the large-scale ILSVRC dataset
- Goodfellow et al. (2014):
  - Deep ConvNets (11 weight layers) in the task of street number recognition and showed increased depth led to better performance

Discussion

- GoogLeNet (Inception):
  - Top-performing entry of the ILSVRC-2014 classification task
  - Similarly based on very deep ConvNets(22 weight layers) and small convolution filters (1x1, 3x3 and 5x5 convolutions).
  - Network topology is more complex
  - Spatial resolution more reduced in first layers to decrease computation
- VGG outperforms Inception in single-network classification accuracy

Classification Framework

Training

- Mini-batch gradient descent with momentum
  - Batch size : 256
  - Momentum : 0.9
- Regularization
  - L2 penalty multiplier : $5 \cdot 10^{-4}$
- First two fully connected layers: dropout regularization with dropout ratio of 0.5

Training

- Learning rate: 0.01
- Decreased by a factor of 10 when the validation set accuracy stopped improving
- Learning rate decreases 3 times
- Stopped after 370K iterations (74 epochs)

Training

- Initialization of weights can be a problem
- Random initialization of weights with normal distribution (μ = 0, $\sigma^2 = 10^{-2}$) and biases = 0
- Train Configuration A with random initializationand use pre-trained weights for other configurations:
  - Initialize first 4 conv. layers and last 3 FC layers
  - No decreasing learning rate

Training image size

- Crop-size fixed at 224x224
- Training set augmentation:
  - Random RGB color shift
  - Random horizontal flipping

Training image size

- Rescale to training scale S ≧ 224. Crop to 224x224
- Single-scale training (Fixed S): S = 256 or S = 384
  - First, train S = 256 and then initialize S = 384 with pre-trained weights from S = 256 and smaller initial learning rate (10^-3)
- Multi-scale training: sampling from $[S_{min}, S_{max}]$ and $S_{min} = 256$, $S_{max} = 512$
  - Training set augmentation by scale jittering
  - Fine-tuning with pre-trained weights from fixed S = 384

Testing

- Rescale the image to a smallest side Q (not necessarily equal to S)
- Test-set augmentation: horizontal flipping of images = > Soft-max of original and flipped averaged to obtain final result

Testing

- Dense evaluation:
  - FC layers convert to convolutional layers
  - Variable resolution (depending on input)
  - Results: a class score map with no. of channels = no. of classes
- Multi-crop: 50 crops per scale for a total of 150 crops

Implementation

- C++ Caffe (Convolutional Architecture for Fast Feature Embedding) toolbox with some modification
- 4 NVIDIA Titan Black GPUs:
  - Speed-up of 3.75 times vs single GPU
  - Single-net training from to 2-3 weeks


GeForce GTX TITAN Black Specifications

CUDA Cores	2880
Base Clock	889 MHz
Boost Clock	980 MHz
Single Precision	5.1 Teraflops
Double Precision	1.3 Teraflops
Memory Config	6GB / 384-bit GDDR5
Memory Speed	7.0 Gbps
Power Connectors	6-pin + 8-pin
TDP	250W
Outputs	2x DL-DVI HDMI Displayport 1.2
Bus Interface	PCI Express 3.0

Classification Experiments

Data

- ILSVRC-2012 dataset
  - 1000 classes
  - 1.3 M training images
  - 50 K validation images
  - 100 K testing images
  - Two performance metrics: Top-1 error and Top-5 error

Single-Scale Evaluation

- Q = S for fixed S
- Q = 0.5(Smin + Smax) for jittered S ∈ [Smin, Smax]

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**

Single-Scale Evaluation

- Local Response Normalization doesn't help

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**

Single-Scale Evaluation

- Performance clearly favors depth (size matters!)

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**

Single-Scale Evaluation

- Prefers 3x3 to 1x1 filters

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**

Single-Scale Evaluation

- Scale jittering at training helps performance
- Performance starts to saturate with depth

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**

Multi-Scale Evaluation

- Multi-Scale Evaluation
  - Run model over several rescaled versions, or Q-values, and average resulting posteriors
  - For fixed S, $Q = \{S - 32, S, S + 32\}$
  - For jittered S, $S \in [S_{min}; S_{max}]$, $Q = \{S_{min}, 0.5(S_{min} + S_{max}), S_{max}\}$

Multi-Scale Evaluation

- Same pattern: depth and prefer jittering, performance starts to saturate with depth

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**

Multi-Crop Evaluation

- Does slightly better than dense
- Best result is averaging both posteriors

Table 5: **ConvNet evaluation techniques comparison.** In all experiments the training scale S was sampled from $[256; 512]$, and three test scales Q were considered: $\{256, 384, 512\}$.

ConvNet config. (Table 1)	Evaluation method	top-1 val. error (%)	top-5 val. error (%)
D	dense	24.8	7.5
	multi-crop	24.6	7.5
	multi-crop & dense	**24.4**	**7.2**
E	dense	24.8	7.5
	multi-crop	24.6	7.4
	multi-crop & dense	**24.4**	**7.1**

ConvNet Fusion

- Average soft-max class posteriors
  - Only got multi-crop results after submission

Table 6: **Multiple ConvNet fusion results.**

Combined ConvNet models	Error		
	top-1 val	top-5 val	top-5 test
ILSVRC submission			
(D/256/224,256,288), (D/384/352,384,416), (D/[256;512]/256,384,512) (C/256/224,256,288), (C/384/352,384,416) (E/256/224,256,288), (E/384/352,384,416)	24.7	7.5	7.3
post-submission			
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), dense eval.	24.0	7.1	7.0
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop	23.9	7.2	-
(D/[256;512]/256,384,512), (E/[256;512]/256,384,512), multi-crop & dense eval.	**23.7**	**6.8**	**6.8**

2-net post submission better than 7-net

ISLVRC-2014 Challenge

- 7-net submission got 2nd place classification
- 2-net post-submission even better!
- 1st place, Szegedy, uses 7-nets

Table 7: **Comparison with the state of the art in ILSVRC classification.** Our method is denoted as "VGG". Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-

Conclusion

- Main contribution: effect of depth on CNN performance
- VGG-16 and VGG-19 (and others) commonly found as pre-trained models as part of DL packages (TF, PyTorch)

VGG-11	30.98	11.37
VGG-13	30.07	10.75
VGG-16	28.41	9.62
VGG-19	27.62	9.12

References

- [1]“tf.keras.applications.VGG16 | TensorFlow Core r2.0,” *TensorFlow*. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/applications/VGG16. [Accessed: 04-Dec-2019].
- [2]“ImageNet.” [Online]. Available: http://www.image-net.org/. [Accessed: 04-Dec-2019].
- [3]“GeForce GTX TITAN Black Gaming Graphics Card | NVIDIA.” [Online]. Available: https://www.nvidia.com/gtx-700-graphics-cards/gtx-titan-black/. [Accessed: 05-Dec-2019].
- [4]“Deep Learning by deeplearning.ai,” *Coursera*. [Online]. Available: https://www.coursera.org/specializations/deep-learning. [Accessed: 05-Dec-2019].
- [5]“02b6266c608492d1007bbb560e762ab4.png (1356×1114).” [Online]. Available: http://images4.programmersought.com/948/02/02b6266c608492d1007bbb560e762ab4.png. [Accessed: 04-Dec-2019].
- [6]C. Szegedy *et al.*, “Going Deeper with Convolutions,” *arXiv:1409.4842 [cs]*, Sep. 2014.
- [7]P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks,” *arXiv:1312.6229 [cs]*, Feb. 2014.
- [8]M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” in *Computer Vision – ECCV 2014*, vol. 8689, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 818–833.
- [9]A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.