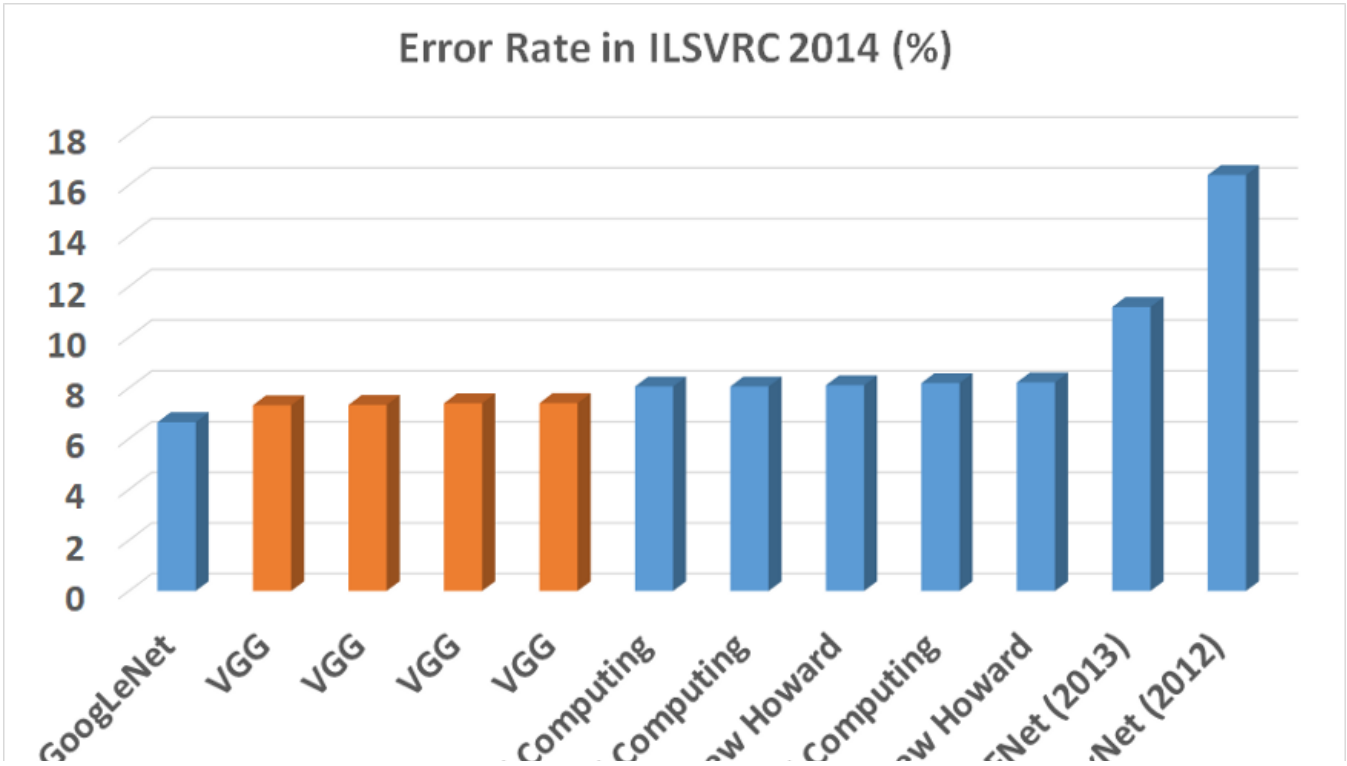
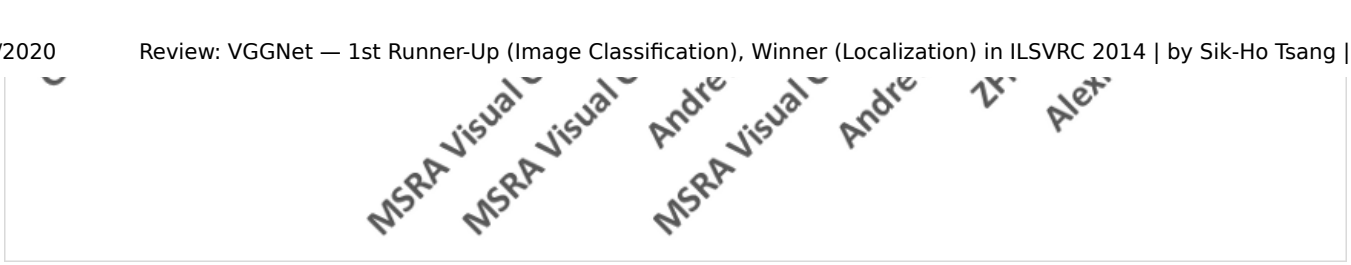
Review: VGGNet — 1st Runner-Up (Image Classification), Winner (Localization) in ILSVRC 2014

Sik-Ho Tsang
Aug 22, 2018 · 9 min read

In this story, **VGGNet [1]** is reviewed. VGGNet is invented by VGG (Visual Geometry Group) from University of Oxford, Though VGGNet is the **1st runner-up**, not the winner of the **ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2014 in the classification task**, which has significantly improvement over ZFNet (The winner in 2013) [2] and AlexNet (The winner in 2012) [3]. And GoogLeNet is the winner of ILSVLC 2014, I will also talk about it later.) Nevertheless, **VGGNet beats the GoogLeNet and won the localization task in ILSVRC 2014.**

And it is **the first year that there are deep learning models obtaining the error rate under 10%.** The most important is that **there are many other models built on top of VGGNet or based on the 3×3 conv idea of VGGNet** for other purposes or other domains. That's why we need to know about VGGNet! That is also why this is a **2015 ICLR paper** with **more than 14000 citations** when I was writing this story. (Sik-Ho Tsang @ Medium)





ILSVRC 2014 Ranking [4]

Usually, people only talked about VGG-16 and VGG-19. I will talk about **VGG-11, VGG-11 (LRN), VGG-13, VGG-16 (Conv1), VGG-16 and VGG-19** by ablation study in the paper.

Dense testing, usually ignored, **will also be covered.**

ImageNet, is a dataset of over 15 millions labeled high-resolution images with around 22,000 categories. ILSVRC uses a subset of ImageNet of around 1000 images in each of 1000 categories. In all, there are roughly 1.3 million training images, 50,000 validation images and 100,000 testing images.

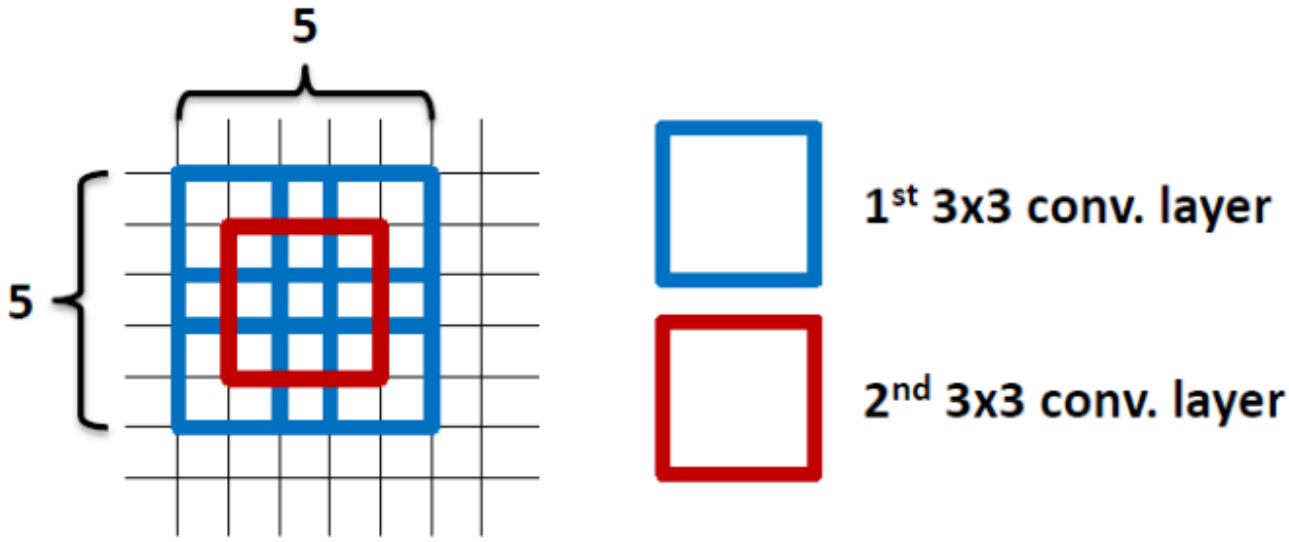
ILSVRC

What we'll cover:

1. **The Use of 3×3 Filters** instead of large-size filters (such as 11×11, 7×7)

2. **VGG-16** and **VGG-19** based on ablation study
   (VGG-11, VGG-11 (LRN), VGG-13, VGG-16 (Conv1) are also included.)
3. **Multi-Scale Training**
4. **Multi-Scale Testing**
5. **Dense Testing**
6. **Model Fusion**
7. **Comparison Between VGGNet and GoogLeNet**
8. **Localization Task**

1. The Use of 3×3 Filters



2 layers of 3×3 filters already covered the 5×5 area

By using 2 layers of 3×3 filters, it actually have already covered 5×5 area as in the above figure. **By using 3 layers of 3×3 filters, it actually have already covered 7×7 effective area.** Thus, large-size filters such as 11×11 in AlexNet [3] and 7×7 in ZFNet [2] indeed are not needed. (If interested, please go to my stories about the ZFNet[5] and AlexNet[6].)

Another reason is that **the number of parameters are fewer.** Suppose there is only 1 filter per layer, 1 layer at input, and exclude the bias:

1 layer of 11×11 filter, number of parameters = **11×11=121**
5 layer of 3×3 filter, number of parameters = **3×3×5=45**
Number of parameters is reduced by 63%

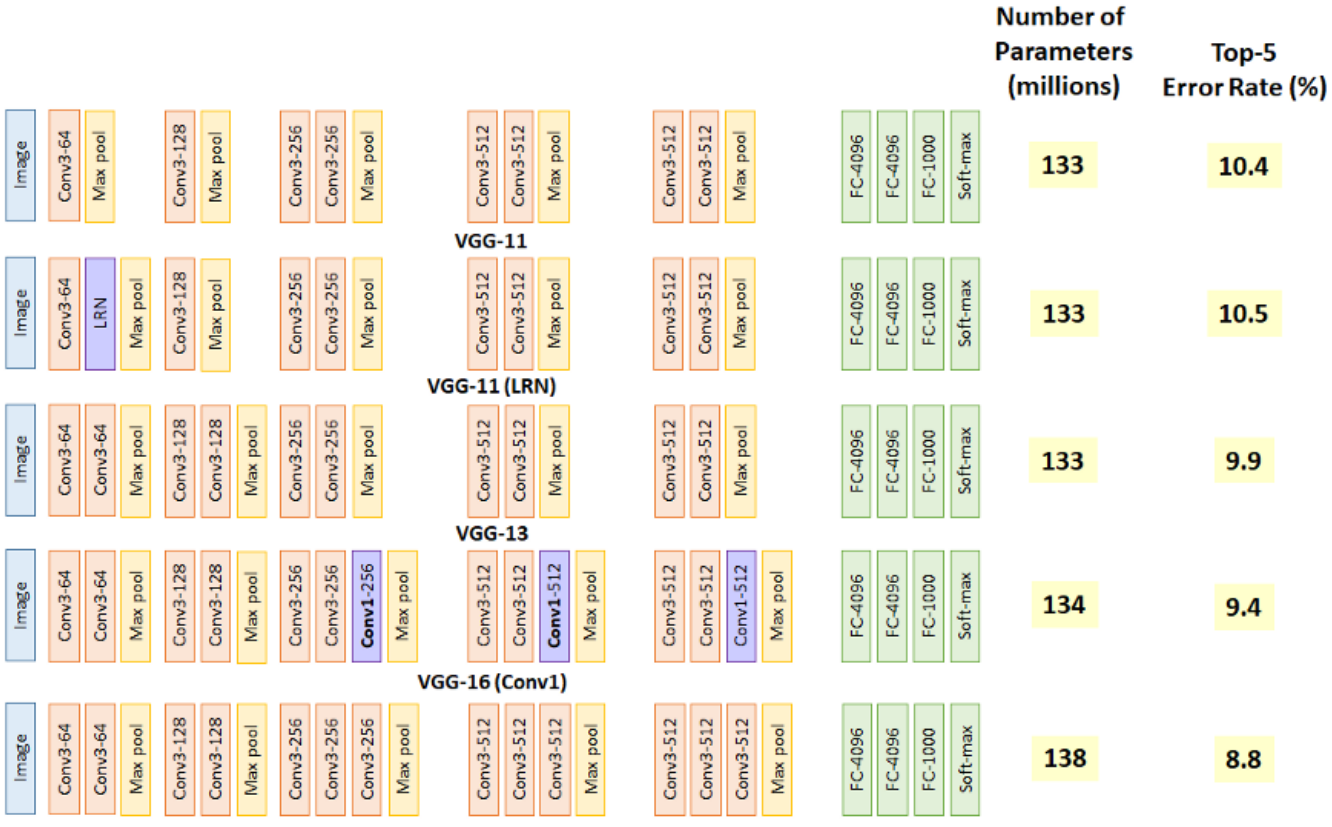
1 layer of 7×7 filter, number of parameters = **7×7=49**
3 layers of 3×3 filters, number of parameters = **3×3×3=27**
Number of parameters is reduced by 45%

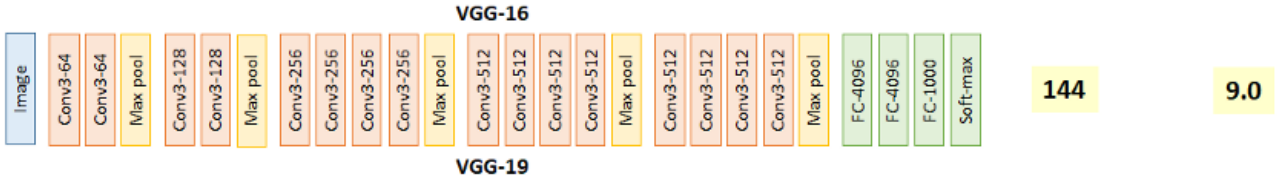
By using **1 layer of 5×5 filter**, number of parameters = 5**×5=25**
By using **2 layers of 3×3 filters**, number of parameters = **3×3+3×3=18**
Number of parameters is reduced by 28%

Larger network, hungrier the network for the training images. There are also vanishing gradient problem. But vanishing gradient problem has been kind of solved by skip connection in ResNet [9].

With **fewer parameters** to be learnt, it is better for **faster convergence**, and **reduced overfitting problem**.

2. VGG-16 and VGG-19 Based on Ablation Study





Different VGG Layer Structures Using Single Scale (256) Evaluation

To obtain the optimum deep learning layer structure, ablation study has been done as shown in the above figure.

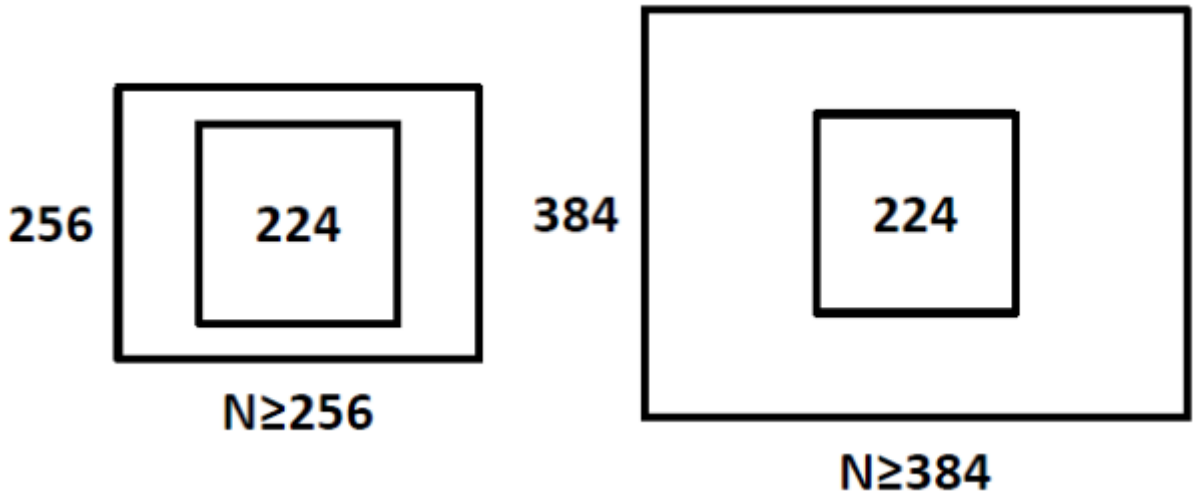
1. First of all, **VGG-11 already obtains 10.4% error rate**, which is similar to that of ZFNet in ILSVRC 2013. VGG-11 is set as benchmark.
2. **VGG-11 (LRN) obtains 10.5% error rate, is the one with additional local response normalization (LRN)** operation suggested by AlexNet. **By comparing VGG-11 and VGG-11 (LRN), the error rate doesn't improve which means LRN is not useful.** In fact, LRN is not used any more in later on deep learning network, instead, batch normalization (BN) is used.
3. **VGG-13 obtains 9.9% error rate, which means the additional conv helps the classification accuracy**.
4. **VGG-16 (Conv1) obtains 9.4% error rate, which means the additional three 1×1 conv layers help the classification accuracy**. **1×1 conv actually helps to increase non-linearlity of the decision function.** Without changing the dimensions of input and output, **1×1 conv is doing the projection mapping in the same high dimensionality.** This technique is essential in a paper called "Network in Network" [7] and also in the GoogLeNet [8] (the winner of ILSVRC 2014) and ResNet [9] (the winner of ILSVRC 2015). I will talk more about GoogLeNet and ResNet review stories in the coming future.
5. **VGG-16 obtains 8.8% error rate which means the deep learning network is still improving by adding number of layers.**
6. **VGG-19 obtains 9.0% error rate which means the deep learning network is NOT improving by adding number of layers.** Thus, authors stop adding layers.

By observing the addition of layers one by one, we can observe that **VGG-16 and VGG-19 start converging** and the accuracy improvement is slowing down. When people are talking about VGGNet, they usually mention VGG-16 and VGG-19.

3. Multi-Scale Training

As object has different scale within the image, **if we only train the network at the same scale, we might miss the detection or have the wrong classification for the objects with other scales.** To tackle this, authors propose multi-scale training.

For **single-scale training**, an image is scaled with smaller-size equal to 256 or 384, i.e. **S=256 or 384.** Since the network accepts 224×224 input images only. The scaled image will be cropped to 224×224. The concept is as follows:



Single-Scale Training with S=256 and S=384

For **multi-scale training**, an image is scaled with smaller-size equal to a range from 256 to 512, i.e. **S=[256;512]**, then cropped to 224×224. Therefore, **with a range of S, we are inputting different scaled objects into the network for training.**

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	VGG-13
	[256;512]	384	27.3	8.8	
D	256	256	27.0	8.8	
	384	384	26.8	8.7	VGG-16
	[256;512]	384	25.6	8.1	
E	256	256	27.3	9.0	
	384	384	26.9	8.7	VGG-19
	[256;512]	384	**25.5**	**8.0**	

Multi-Scale Training Results

By using multi-scale training, we can imagine that **it is more accurate for test image objects with different object sizes.**

VGG-13 reduced the error rate from 9.4%/9.3% to **8.8%.**
VGG-16 reduced the error rate from 8.8%/8.7% to **8.1%.**
VGG-19 reduced the error rate from 9.0%/8.7% to **8.0%.**

4. Multi-Scale Testing

Similar to multi-scale training, **multi-scale testing** can also reduce the error rate since we do not know the size of object in the test image. If we **scale the test image to different sizes**, we can **increase the chance of correct classification**.

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	VGG-13
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	VGG-16
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	VGG-19

Multi-Scale Testing Results

By using multi-scale testing but single-scale training, error rate is reduced.
Compared to single-scale training single-scale testing,

VGG-13 reduced the error rate from 9.4%/9.3% to **9.2%.**
VGG-16 reduced the error rate from 8.8%/8.7% to **8.6%.**
VGG-19 reduced the error rate from 9.0%/8.7% to 8.7/8.6**%.**

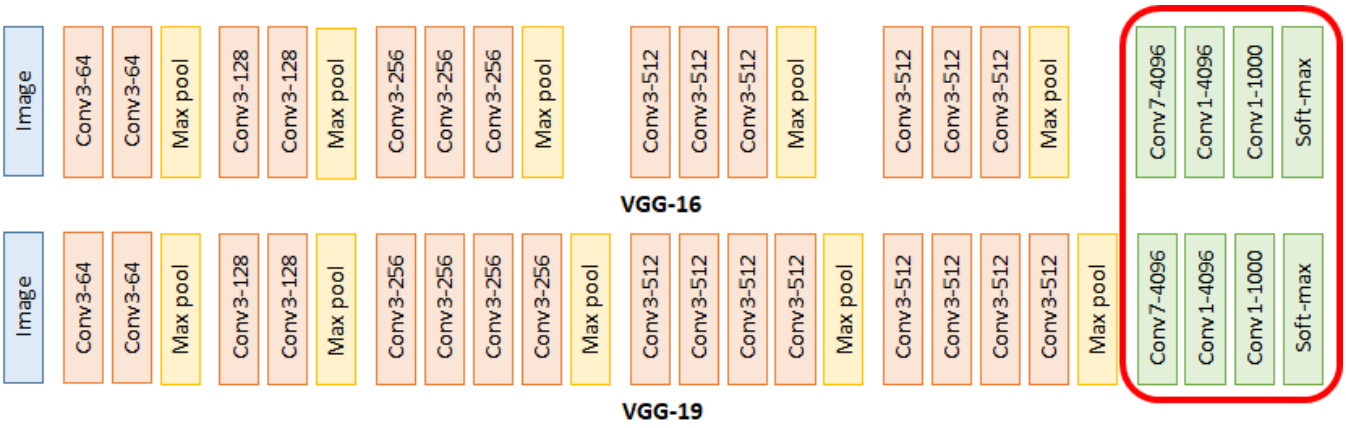
By using both multi-scale training and testing, error rate is reduced**.**
Compared with only multi-scale testing,

VGG-13 reduced the error rate from 9.2%/9.2% to **8.2%,**
VGG-16 reduced the error rate from 8.6%/8.6% to **7.5%,**
VGG-19 reduced the error rate from 8.7%/8.6% to 7**.5%,**

5. Dense (Convolutionalized) Testing

During testing, in **AlexNet**, the **4 corners and center of the image** as well as their **horizontal flips** are cropped for testing, i.e. **10 times of testing**. And the output probability vectors are added or averaged to get a better result.

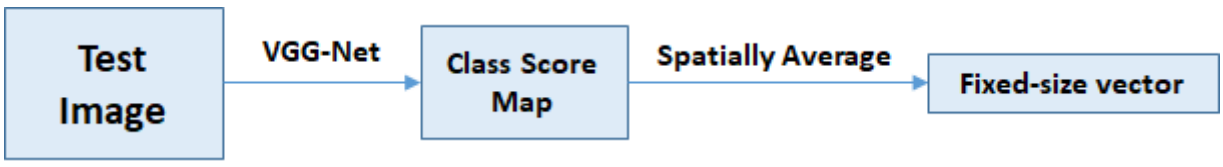
The VGGNet is different from the one in training as shown below:



VGGNet During Testing

The first FC is replaced by 7×7 conv.
The second and third FC are replaced by 1×1 conv.
Thus, all FC layers are replaced by conv layers.

During testing, in **VGGNet**, the test image is directly go through the VGGNet and obtain a class score map. This class score map is spatially averaged to be a fixed-size vector.



Workflow of VGGNet Testing

There are **only 2 times of testing** if we also include the horizontal flip as well.

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	VGG-16
	multi-crop & dense	**24.4**	7.2	
E	dense	24.8	7.5	
	multi-crop	24.6	7.4	VGG-19
	multi-crop & dense	24.4	7.1	

Dense (VGGNet), Multi-crop (Approach by AlexNet), Dense+Multi-crop (Both)

By average both dense and multi-crop results, VGG-16 and VGG-19 error rates are reduced to 7.2% and 7.1%.

6. Model Fusion

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

Fusion All Techniques Mentioned Above

By combining VGG-16 and VGG-19 plus multi-scale training, multi-scale testing, mutli-crop and dense, error rate is reduced to 6.8%.

7. Comparison Between VGGNet and GoogLeNet

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	-

VGGNet

GoogLeNet

Comparison Between VGGNet and GoogLeNet

Compared with **GoogLeNet using 7-nets** which has **error rate of 6.7%**, **VGGNet using 2-nets**, plus multi scale training, multi-scale testing, mutli-crop and dense has **error rate of 6.8%** which are competitive.

With only 1-net, **VGGNet has 7.0% error rate** which is better than **GoogLeNet, that has 7.9% error rate**.

However, **at the submission of ILSVRC 2014, VGGNet has 7.3% error rate only which got 1st runner up at the moment.**

8. Localization Task (Post Updated on 2nd Sept 2018)

For the localization task, a bounding box is represented by a 4-D vector storing its center coordinates, width, and height. Thus, the logistic regression objective is replaced with a Euclidean loss, which penalises the deviation of the predicted bounding box parameters from the ground-truth.

There is a choice of whether the bounding box prediction is shared across all classes (**single-class regression, SCR**) or is class-specific (**per-class regression, PCR**). In the former case, the last layer is **4-D**, while in the latter it is **4000-D** (since there are 1000 classes in the dataset).

Fine-tuned layers	regression type	GT class localisation error
1st and 2nd FC	SCR	36.4
	PCR	34.3
all	PCR	**33.1**

Localization Results

As shown above, **PCR is better than SCR.** And **fine-tuning all layers is better** than just fine-tuned the 1st and 2nd FC layers. The results above is only obtained by using just center crop.

smallest image side		top-5 localisation error (%)	
train (S)	test (Q)	val.	test.
256	256	29.5	-
384	384	28.2	26.7
384	352,384	27.5	-
fusion: 256/256 and 384/352,384		**26.9**	**25.3**

Multi-Scale Training and Testing

With **multiple training and testing** which just described in previous sections, the **top-5 localization error is reduced to 25.3%.**

Method	top-5 val. error (%)	top-5 test error (%)
VGG	**26.9**	**25.3**
GoogLeNet (Szegedy et al., 2014)	-	26.7
OverFeat (Sermanet et al., 2014)	30.0	29.9
Krizhevsky et al. (Krizhevsky et al., 2012)	-	34.2

Comparison with state-of-the-art results

VGGNet even outperforms GoogLeNet as shown above and **won the localization task in ILSVRC 2014**.

Method	VOC-2007 (mean AP)	VOC-2012 (mean AP)	Caltech-101 (mean class recall)	Caltech-256 (mean class recall)
Zeiler & Fergus (Zeiler & Fergus, 2013)	-	79.0	86.5 ± 0.5	74.2 ± 0.3
Chatfield et al. (Chatfield et al., 2014)	82.4	83.2	88.4 ± 0.6	77.6 ± 0.1
He et al. (He et al., 2014)	82.4	-	**93.4 ± 0.5**	-
Wei et al. (Wei et al., 2014)	81.5 (85.2*)	81.7 (**90.3***)	-	-
VGG Net-D (16 layers)	89.3	89.0	91.8 ± 1.0	85.0 ± 0.2
VGG Net-E (19 layers)	89.3	89.0	92.3 ± 0.5	85.1 ± 0.3
VGG Net-D & Net-E	**89.7**	**89.3**	92.7 ± 0.5	**86.2 ± 0.3**

VOC 2007, 2012 and Caltech 101 and 256 Dataset Results

VGGNet has the best results on VOC 2007, 2012 and Caltech 256 dataset. And it also has **competitive result on Caltech 101 dataset.**

I will cover GoogLeNet [8], ResNet [9], and so on for the image classification. Please stay tuned.

References

1. [2015 ICLR] [VGGNet]
   Very Deep Convolutional Networks for Large-Scale Image Recognition

2. [2014 ECCV] [ZFNet]
   Visualizing and Understanding Convolutional Networks

3. [2012 NIPS] [AlexNet]
   ImageNet Classification with Deep Convolutional Neural Networks

4. ILSVRC 2014 Ranking
   http://www.image-net.org/challenges/LSVRC/2014/results#clsloc

5. Review of ZFNet — Winner of ILSVRC 2013 (Image Classification)

6. Review of AlexNet, CaffeNet — Winner of ILSVRC 2012 (Image Classification)

7. [2014 ICLR] [NIN]
   Network in Network

8. [2015 CVPR] [GoogLeNet]
   Going Deeper With Convolutions

9. [2016 CVPR] [ResNet]
   Deep Residual Learning for Image Recognition