VGG-16

VGG-16

- Proposed in 2014

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan[*] **& Andrew Zisserman**[+]
Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk

ABSTRACT

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. These findings were the basis of our ImageNet Challenge 2014 submission, where our team secured the first and the second places in the localisation and classification tracks respectively. We also show that our representations generalise well to other datasets, where they achieve state-of-the-art results. We have made our two best-performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

Analytics Vidhya

VGG-16

- Proposed in 2014
- It has 16 layers with learnable parameters

VGG-16

- Proposed in 2014
- It has 16 layers with learnable parameters
- Simpler architecture, less number of hyperparameters

VGG-16

- Proposed in 2014
- It has 16 layers with learnable parameters
- Simpler architecture, less number of hyperparameters
- Architecture details:
    - Kernel size fixed to 3 X 3 for all the convolutional layers

VGG-16

- Proposed in 2014
- It has 16 layers with learnable parameters
- Simpler architecture, less number of hyperparameters
- Architecture details:
    - Kernel size fixed to 3 X 3 for all the convolutional layers
    - Stride - 1
    - Padding - 1

VGG-16

- Proposed in 2014
- It has 16 layers with learnable parameters
- Simpler architecture, less number of hyperparameters
- Architecture details:
  - Kernel size fixed to 3 X 3 for all the convolutional layers
  - Stride - 1
  - Padding - 1
  - Activation function - ReLU

VGG-16

- Proposed in 2014
- It has 16 layers with learnable parameters
- Simpler architecture, less number of hyperparameters
- Architecture details:
    - Kernel size fixed to 3 X 3 for all the convolutional layers
    - Stride - 1
    - Padding - 1
    - Activation function - ReLU
- Trained on imagenet dataset

Architecture: VGG-16

```
VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU(inplace=True)
    (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (3): ReLU(inplace=True)
    (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (6): ReLU(inplace=True)
    (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): ReLU(inplace=True)
    (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): ReLU(inplace=True)
    (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (13): ReLU(inplace=True)
    (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): ReLU(inplace=True)
    (16): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (17): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (18): ReLU(inplace=True)
    (19): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (20): ReLU(inplace=True)
    (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (22): ReLU(inplace=True)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (24): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (25): ReLU(inplace=True)
    (26): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (27): ReLU(inplace=True)
    (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (29): ReLU(inplace=True)
    (30): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
  (classifier): Sequential(
    (0): Linear(in_features=25088, out_features=4096, bias=True)
    (1): ReLU(inplace=True)
    (2): Dropout(p=0.5, inplace=False)
    (3): Linear(in_features=4096, out_features=4096, bias=True)
    (4): ReLU(inplace=True)
    (5): Dropout(p=0.5, inplace=False)
    (6): Linear(in_features=4096, out_features=1000, bias=True)
  )
)
```

Architecture: VGG-16

Layer	# filters	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-			224 X 224 X 3	

Architecture: VGG-16

Layer	# filters	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-			224 X 224 X 3	
Conv 1	64	3X3	1	1	224 X 224 X 64	ReLU
Conv 2	64	3X3	1	1	224 X 224 X 64	ReLU
Max Pooling 1	-	2X2	2		112 X 112 X 64	

Architecture: VGG-16

Layer	# filters	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-			224 X 224 X 3	
Conv 1	64	3X3	1	1	224 X 224 X 64	ReLU
Conv 2	64	3X3	1	1	224 X 224 X 64	ReLU
Max Pooling 1	-	2X2	2		112 X 112 X 64	
Conv 3	128	3X3	1	1	112 X 112 X 128	ReLU
Conv 4	128	3X3	1	1	112 X 112 X 128	ReLU
Max Pooling 2	-	2X2	2		56 X 56 X 128	

Architecture: VGG-16

Layer	# filters	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-			224 X 224 X 3	
Conv 1	64	3X3	1	1	224 X 224 X 64	ReLU
Conv 2	64	3X3	1	1	224 X 224 X 64	ReLU
Max Pooling 1	-	2X2	2		112 X 112 X 64	
Conv 3	128	3X3	1	1	112 X 112 X 128	ReLU
Conv 4	128	3X3	1	1	112 X 112 X 128	ReLU
Max Pooling 2	-	2X2	2		56 X 56 X 128	
Conv 5	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 6	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 7	256	3X3	1	1	56 X 56 X 256	ReLU
Max Pooling 3	-	2X2	2		28 X 28 X 256	

Architecture: VGG-16

Layer	# filters	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-			224 X 224 X 3	
Conv 1	64	3X3	1	1	224 X 224 X 64	ReLU
Conv 2	64	3X3	1	1	224 X 224 X 64	ReLU
Max Pooling 1	-	2X2	2		112 X 112 X 64	
Conv 3	128	3X3	1	1	112 X 112 X 128	ReLU
Conv 4	128	3X3	1	1	112 X 112 X 128	ReLU
Max Pooling 2	-	2X2	2		56 X 56 X 128	
Conv 5	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 6	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 7	256	3X3	1	1	56 X 56 X 256	ReLU
Max Pooling 3	-	2X2	2		28 X 28 X 256	
Conv 8	512	3X3	1	1	28 X 28 X 512	ReLU
Conv 9	512	3X3	1	1	28 X 28 X 512	ReLU
Conv 10	512	3X3	1	1	28 X 28 X 512	ReLU
Max Pooling 4	-	2X2	2		14 X 14 X 512	

Architecture: VGG-16

Layer	# filters	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-			224 X 224 X 3	
Conv 1	64	3X3	1	1	224 X 224 X 64	ReLU
Conv 2	64	3X3	1	1	224 X 224 X 64	ReLU
Max Pooling 1	-	2X2	2		112 X 112 X 64	
Conv 3	128	3X3	1	1	112 X 112 X 128	ReLU
Conv 4	128	3X3	1	1	112 X 112 X 128	ReLU
Max Pooling 2	-	2X2	2		56 X 56 X 128	
Conv 5	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 6	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 7	256	3X3	1	1	56 X 56 X 256	ReLU
Max Pooling 3	-	2X2	2		28 X 28 X 256	
Conv 8	512	3X3	1	1	28 X 28 X 512	ReLU
Conv 9	512	3X3	1	1	28 X 28 X 512	ReLU
Conv 10	512	3X3	1	1	28 X 28 X 512	ReLU
Max Pooling 4	-	2X2	2		14 X 14 X 512	
Conv 11	512	3X3	1	1	14 X 14 X 512	ReLU
Conv 12	512	3X3	1	1	14 X 14 X 512	ReLU
Conv 13	512	3X3	1	1	14 X 14 X 512	ReLU
Max Pooling 5	-	2X2	2		7 X 7 X 512	

Architecture: VGG-16

Layer	# filters	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-			224 X 224 X 3	
Conv 1	64	3X3	1	1	224 X 224 X 64	ReLU
Conv 2	64	3X3	1	1	224 X 224 X 64	ReLU
Max Pooling 1	-	2X2	2		112 X 112 X 64	
Conv 3	128	3X3	1	1	112 X 112 X 128	ReLU
Conv 4	128	3X3	1	1	112 X 112 X 128	ReLU
Max Pooling 2	-	2X2	2		56 X 56 X 128	
Conv 5	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 6	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 7	256	3X3	1	1	56 X 56 X 256	ReLU
Max Pooling 3	-	2X2	2		28 X 28 X 256	
Conv 8	512	3X3	1	1	28 X 28 X 512	ReLU
Conv 9	512	3X3	1	1	28 X 28 X 512	ReLU
Conv 10	512	3X3	1	1	28 X 28 X 512	ReLU
Max Pooling 4	-	2X2	2		14 X 14 X 512	
Conv 11	512	3X3	1	1	14 X 14 X 512	ReLU
Conv 12	512	3X3	1	1	14 X 14 X 512	ReLU
Conv 13	512	3X3	1	1	14 X 14 X 512	ReLU
Max Pooling 5	-	2X2	2		7 X 7 X 512	
Flatten	-				25088	

Architecture: VGG-16

Layer	# filters	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-			224 X 224 X 3	
Conv 1	64	3X3	1	1	224 X 224 X 64	ReLU
Conv 2	64	3X3	1	1	224 X 224 X 64	ReLU
Max Pooling 1	-	2X2	2		112 X 112 X 64	
Conv 3	128	3X3	1	1	112 X 112 X 128	ReLU
Conv 4	128	3X3	1	1	112 X 112 X 128	ReLU
Max Pooling 2	-	2X2	2		56 X 56 X 128	
Conv 5	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 6	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 7	256	3X3	1	1	56 X 56 X 256	ReLU
Max Pooling 3	-	2X2	2		28 X 28 X 256	
Conv 8	512	3X3	1	1	28 X 28 X 512	ReLU
Conv 9	512	3X3	1	1	28 X 28 X 512	ReLU
Conv 10	512	3X3	1	1	28 X 28 X 512	ReLU
Max Pooling 4	-	2X2	2		14 X 14 X 512	
Conv 11	512	3X3	1	1	14 X 14 X 512	ReLU
Conv 12	512	3X3	1	1	14 X 14 X 512	ReLU
Conv 13	512	3X3	1	1	14 X 14 X 512	ReLU
Max Pooling 5	-	2X2	2		7 X 7 X 512	
Flatten	-				25088	
Fully Connected 1					4096	ReLU
Fully Connected 2					4094	ReLU
Fully Connected 3					1000	Softmax

Analytics Vidhya

Architecture: VGG-16

Layer	# filters	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-			224 X 224 X 3	
Conv 1	64	3X3	1	1	224 X 224 X 64	ReLU
Conv 2	64	3X3	1	1	224 X 224 X 64	ReLU
Max Pooling 1	-	2X2	2		112 X 112 X 64	
Conv 3	128	3X3	1	1	112 X 112 X 128	ReLU
Conv 4	128	3X3	1	1	112 X 112 X 128	ReLU
Max Pooling 2	-	2X2	2		56 X 56 X 128	
Conv 5	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 6	256	3X3	1	1	56 X 56 X 256	ReLU
Conv 7	256	3X3	1	1	56 X 56 X 256	ReLU
Max Pooling 3	-	2X2	2		28 X 28 X 256	
Conv 8	512	3X3	1	1	28 X 28 X 512	ReLU
Conv 9	512	3X3	1	1	28 X 28 X 512	ReLU
Conv 10	512	3X3	1	1	28 X 28 X 512	ReLU
Max Pooling 4	-	2X2	2		14 X 14 X 512	
Conv 11	512	3X3	1	1	14 X 14 X 512	ReLU
Conv 12	512	3X3	1	1	14 X 14 X 512	ReLU
Conv 13	512	3X3	1	1	14 X 14 X 512	ReLU
Max Pooling 5	-	2X2	2		7 X 7 X 512	
Flatten	-				25088	
Fully Connected 1					4096	ReLU
Fully Connected 2					4094	ReLU
Fully Connected 3					1000	Softmax

Parameters =
138 million

Summary: VGG-16

Summary: VGG-16

- It has 16 layers with learnable parameters

Summary: VGG-16

- It has 16 layers with learnable parameters
- Takes RGB image as input

Summary: VGG-16

- It has 16 layers with learnable parameters
- Takes RGB image as input
- Architecture details:
    - 13 convolutional + 3 fully connected layers
    - Kernel size: 3 X 3
    - Stride - 1
    - Padding - 1
    - Activation function - ReLU

Summary: VGG-16

- It has 16 layers with learnable parameters
- Takes RGB image as input
- Architecture details:
    - 13 convolutional + 3 fully connected layers
    - Kernel size: 3 X 3
    - Stride - 1
    - Padding - 1
    - Activation function - ReLU
- Total number of parameters: 138 million

Thank You