

VGGNet is a deep Convolutional Neural Network (CNN) architecture introduced by the Visual Geometry Group (VGG) at the University of Oxford in 2014. It became widely popular due to its simplicity and effectiveness in image classification tasks, particularly in the ILSVRC 2014 (ImageNet Large Scale Visual Recognition Challenge) where it secured one of the top positions.

Agenda

- 1. Network Architecture
- 2. 3 x 3 convolution
- 3. VGG Variants

Published as a conference paper at ICLR 2015

VERY DEEP CONVOLUTIONAL NETWORKS  
FOR LARGE-SCALE IMAGE RECOGNITION

Karim Simonyan\* & Andrew Zisserman\*

Visual Geometry Group, Department of Engineering Science, University of Oxford  
{karim.simonyan, andrew.zisserman}@eng.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 varying depths on the standard ImageNet dataset. These findings were the basis of our ImageNet Challenge 2014 submission, where we won second place. We also show that our representations generalize well to other datasets, where they achieve state-of-the-art results. We have made our very best performing ConvNet models publicly available to facilitate further research on the use of deep visual representations in computer vision.

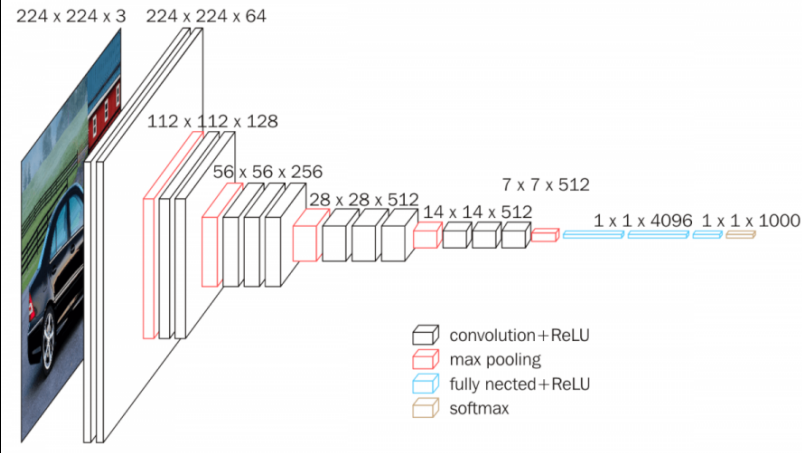
1 INTRODUCTION

Convolutional networks (ConvNets) have recently enjoyed a great success in large-scale image and video recognition (Krizhevsky et al., 2012; Zeiler & Fergus, 2013; Simonyan et al., 2014; Szegedy & Zisserman, 2014b) which has become possible due to the large public image recognition, such as ImageNet (Deng et al., 2009), and top performance competition, such as IAPR CVC (2009) and ILSVRC (2012). In this paper, we report on our investigation of the influence of deep visual recognition architectures has been played by the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015), which has acted as a catalyst for a new generation of large-scale image classification systems, from high-dimensional shallow feature embeddings (Krizhevsky et al., 2012) to the winner of ILSVRC 2015 (Zeiler & Fergus, 2015).

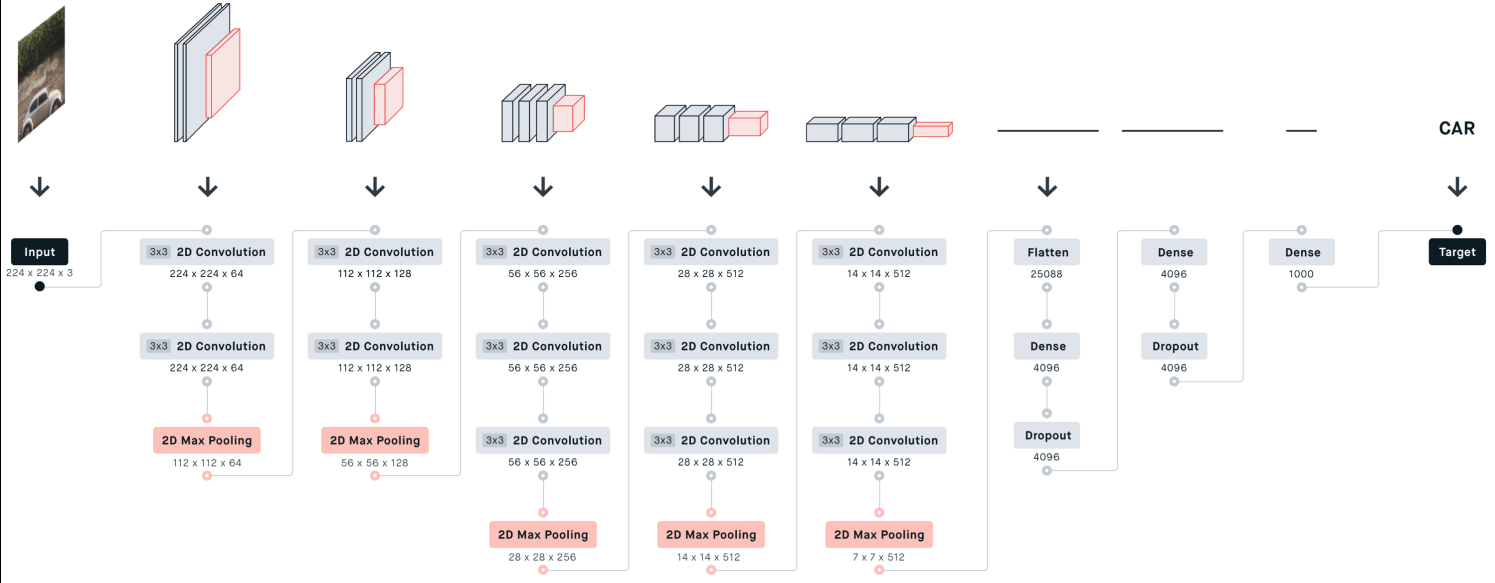
With ConvNets having made a mark in the computer vision field, a number of attempts have been made to improve the original architecture of Krizhevsky et al. (2012) in order to achieve better accuracy. For instance, the best performing submissions to the ILSVRC 2015 (Zeiler & Fergus, 2015; Szegedy et al., 2015) utilized smaller receptive fields, one and smaller stride of the first convolutional layer. Another line of improvements dealt with training and testing the networks directly over the whole image and area multiple times (Simonyan et al., 2014; Howard, 2014). In this paper, we address another important aspect of ConvNet architecture design – its depth. To this end, we fix other parameters of the architecture, and usually increase the depth of the network by adding more convolutional layers, which is feasible due to the use of crop-and-stitch (C & S) convolution filters in all layers.

As a result, we come up with significantly more accurate ConvNet architectures, which we will

arXiv:1409.1556v6 [cs.CV] 10 Apr 2015



	Layer	Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax



13 Conv Layers  
5 M.P  
3 FC  
VGG 16  
VGG 19

Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as “conv(receptive field size)-(number of channels)”. The ReLU activation function is not shown for brevity.

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 ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	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 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

V66616/19

Design of V666

1) Stacking of CNN  $\rightarrow$  Feature/Patterns

2)  $3 \times 3$  Filters

$$5 \times 5 = 25$$

$$2(3 \times 3) = 18$$

$$30 \times 30 \times ?$$

$$28 \times 28 \times ?$$

$$\text{Image: } \underline{32 \times 32 \times 3}$$

$$28 \times 28 \times ?$$

Smaller Receptive Fields are efficient

## Vanishing Gradient

Trainable Params :- 138M

Accuracy :- 92.7%

## Implementations

1) Pre Trained Version

2) Transfer Learning

16 layers

15 layers  $\rightarrow$  Freeze

1 layer  $\rightarrow$  Train