State-of-the-art Convolutional Neural Networks

Francesco Ciompi, Radboudumc

NFBIA Summer School 2015

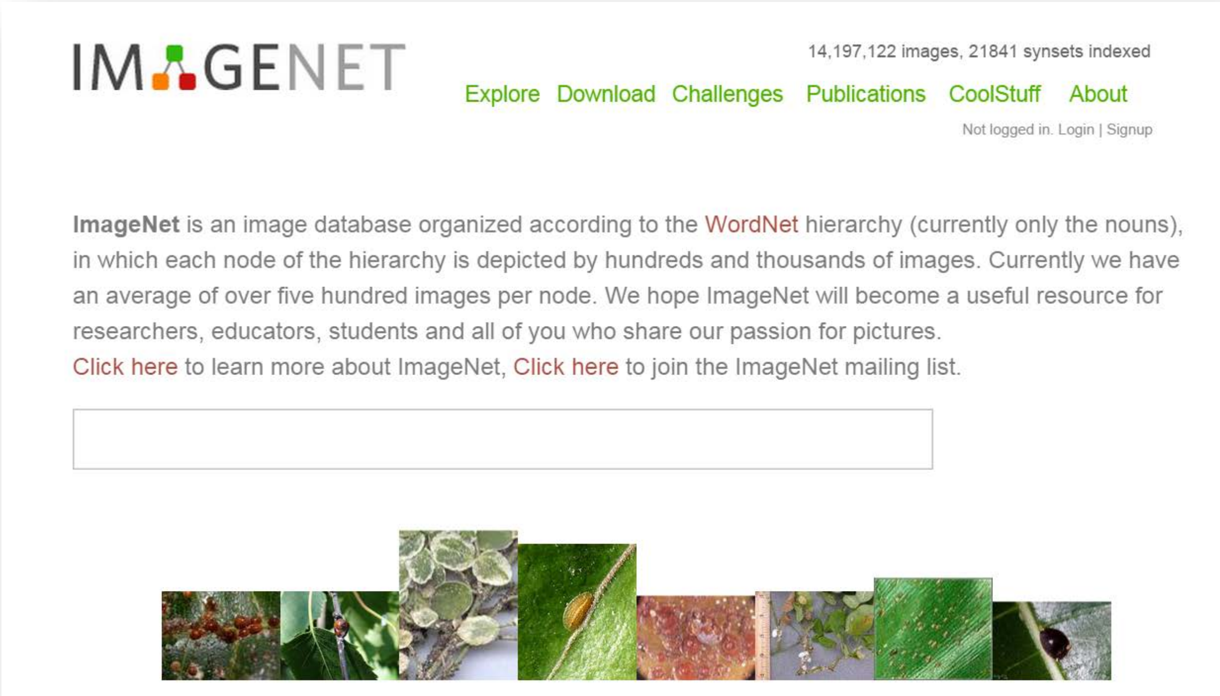
September 15, 2015, Nijmegen

Advances in convolutional networks

- Several factors contribute to the effectiveness of convolutional networks:
  - Implementation on GPUs, which allows fast training of deep networks
  - Design and training techniques: Dropout, ReLU, momentum, etc.

- Most of the innovations in the field of convolutional networks come from the **computer vision** community, especially in **natural image** analysis

- Our mission is to transfer this to the field of **medical image** analysis



More than 14 million images
More than 21,000 classes labeled

1.2 million images
1,000 classes labeled

Challenges like ILSVRC2015 really motivates researcher in computer vision and deep learning to improve and compare their approaches.

This helps to move research forward, and to fairly evaluate new methods.

AlexNet, 2012

https://code.google.com/p/cuda-convnet/

ImageNet Classification with Deep Convolutional Neural Networks

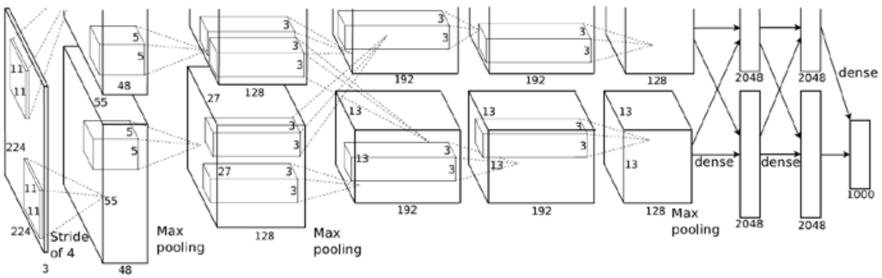
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

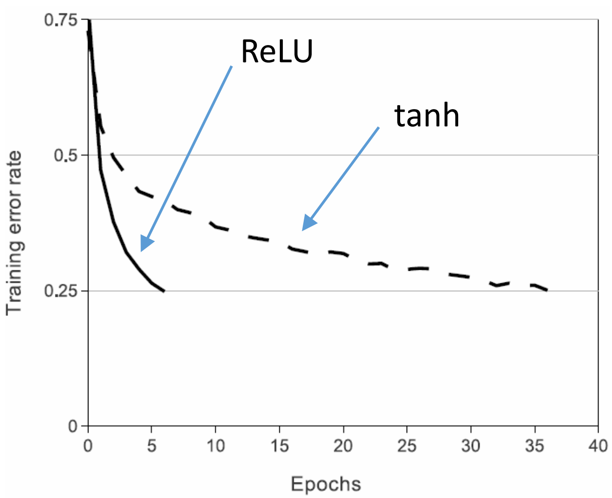
We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.



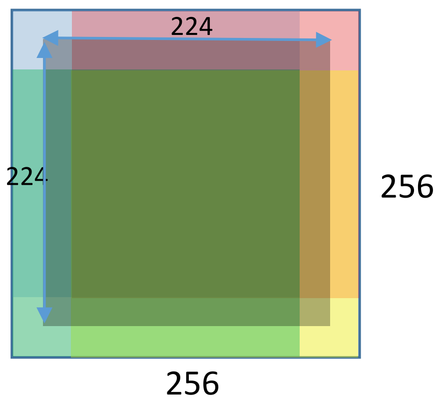
- Trained on **ILSVRC_train**, 1.2M images, 1000 classes
- **8 layers** (5 convolutional, 3 fully-connected)
- Trained on **2 GPUs** with 3GB of RAM each
- **Local response normalization**
- **Overlapping pooling**

AlexNet, 2012

- **ReLU and Dropout**



- **Data augmentation**
  - **Training time**
    - Translations + reflections
    - Increased by a factor 2048
  - **Test time**



5 patches of size 224x224

Four corners + center

Each patch + horizontal reflection

-> 10 test patches

AlexNet, 2012

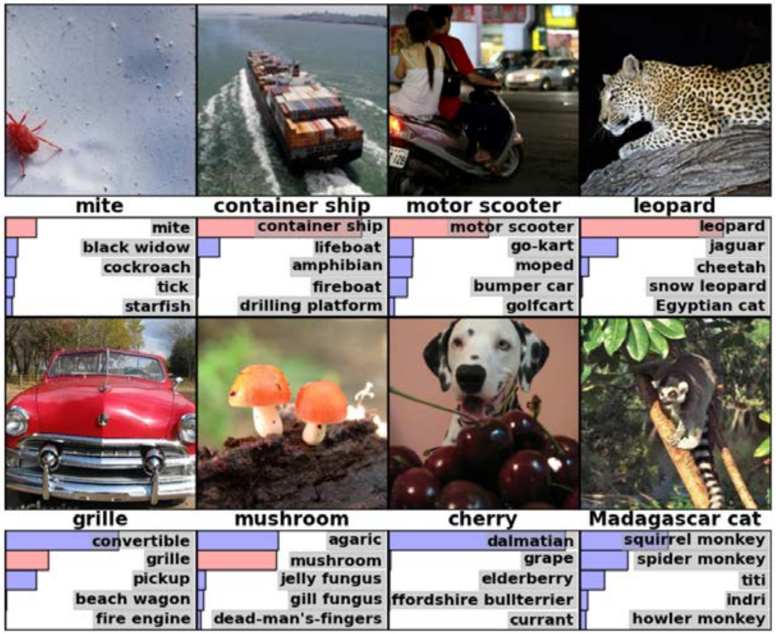
- **Results**

ILSVRC-2010

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

ILSVRC-2012

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**



VDD (aka OxfordNet)

http://www.robots.ox.ac.uk/~vgg/research/very_deep/

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.

- 19 layers
- Trained on 4 GPUs
- Weight decay
- Dropout
- ReLU
- Data augmentation
- **Fixed filter size: 3x3**

VDD (aka OxfordNet)

ConvNet Configuration
A
11 weight layers
conv3-64
conv3-128
conv3-256 conv3-256
conv3-512 conv3-512
conv3-512 conv3-512
maxpool
FC-4096
FC-4096
FC-1000
soft-max

VDD (aka OxfordNet)

ConvNet Configuration	
A	A-LRN
11 weight layers	11 weight layers
	i
conv3-64	conv3-64 **LRN**
conv3-128	conv3-128
conv3-256 conv3-256	conv3-256 conv3-256
conv3-512 conv3-512	conv3-512 conv3-512
conv3-512 conv3-512	conv3-512 conv3-512
maxpool	
FC-4096	
FC-4096	
FC-1000	
soft-max	

Contrarily to AlexNet, **Local Response Normalization** does not help improve the performance

VDD (aka OxfordNet)

ConvNet Configuration		
A	A-LRN	B
11 weight layers	11 weight layers	13 weight layers
input (224 × 2		
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**
max		
conv3-128	conv3-128	conv3-128 **conv3-128**
max		
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256
max		
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512
max		
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512
maxpool		
FC-4096		
FC-4096		
FC-1000		
soft-max		

VDD (aka OxfordNet)

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					

Results on ILSVRC-2014

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**

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**

Test done with fully-convolutional network

GoogLeNet

Going deeper with convolutions

Christian Szegedy
Google Inc.

Wei Liu
University of North Carolina, Chapel Hill

Yangqing Jia
Google Inc.

Pierre Sermanet
Google Inc.

Scott Reed
University of Michigan

Dragomir Anguelov
Google Inc.

Dumitru Erhan
Google Inc.

Vincent Vanhoucke
Google Inc.

Andrew Rabinovich
Google Inc.

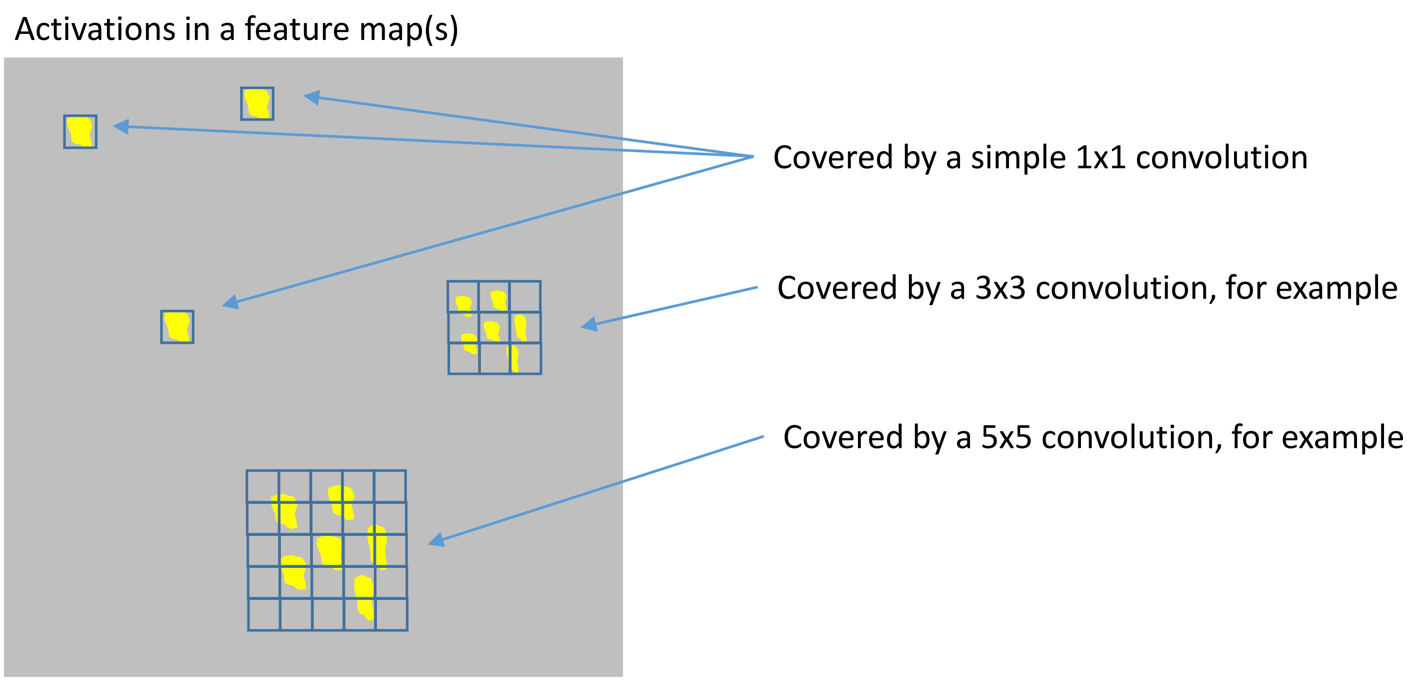
Abstract

We propose a deep convolutional neural network architecture codenamed Inception, which was responsible for setting the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.

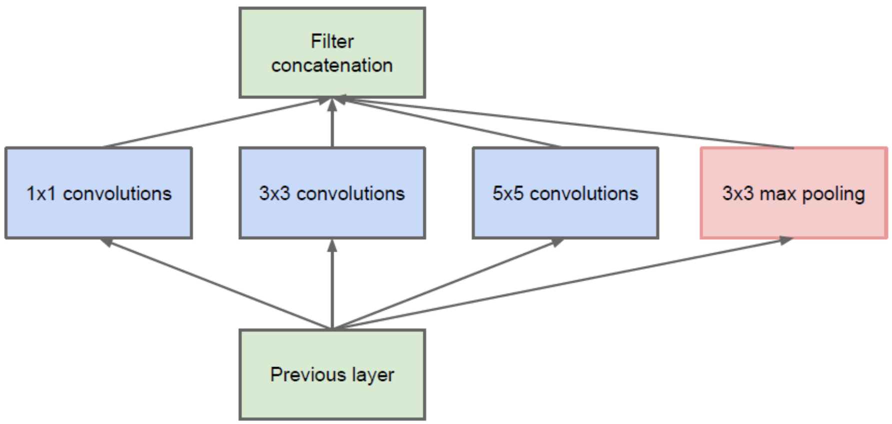
- 22 layers
- Trained on DistBelief
- Introduces a new layer architecture
- Inspired by Hebbian principle

GoogLeNet

- Hebbian principle: *“neurons that fire together, wire together”.*
- We expect that information in hidden layers has a **sparse** structure (using ReLU helps in that sense though…)



GoogLeNet



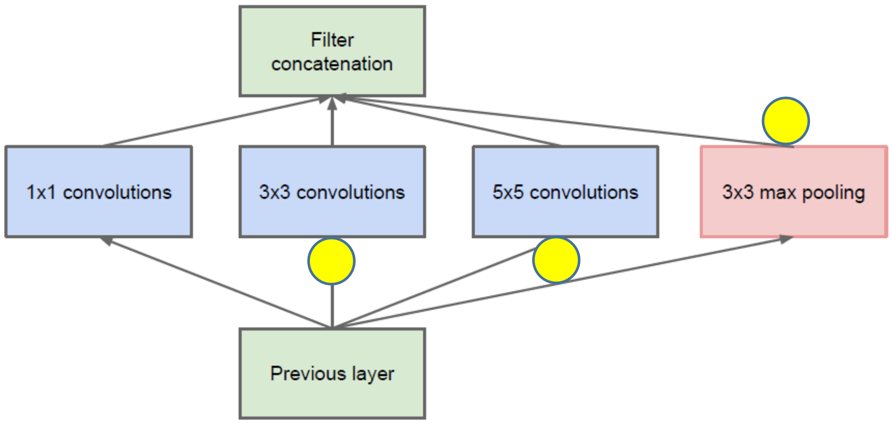
This is the new layer architecture

The outputs are **concatenated**

Problems

- Concatenation makes the matrix very big
- max-pooling does not change the number of feature maps, only their size

GoogLeNet



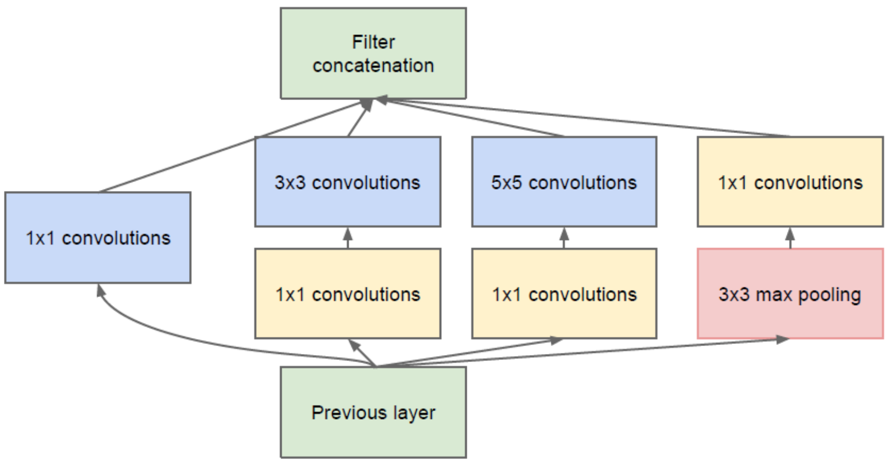
This is the new layer architecture

The outputs are **concatenated**

Problems

- Concatenation makes the matrix very big
- max-pooling does not change the number of feature maps, only their size
- **We should reduce the feature maps at some locations in the architecture**

GoogLeNet



The solution is using additional 1x1 convolutions!

It does not change the feature map size

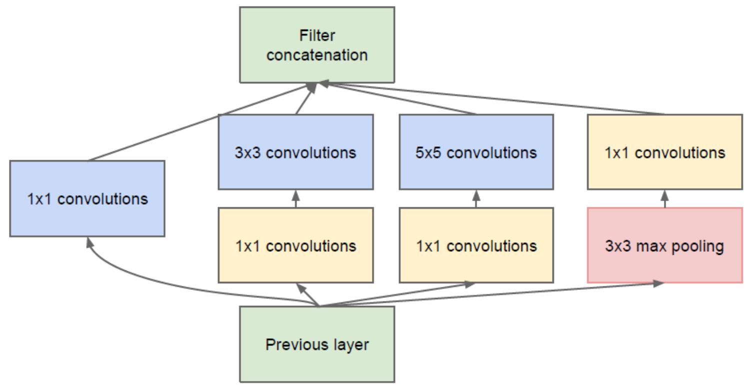
We can tune the number of output feature maps: **dimension reduction**

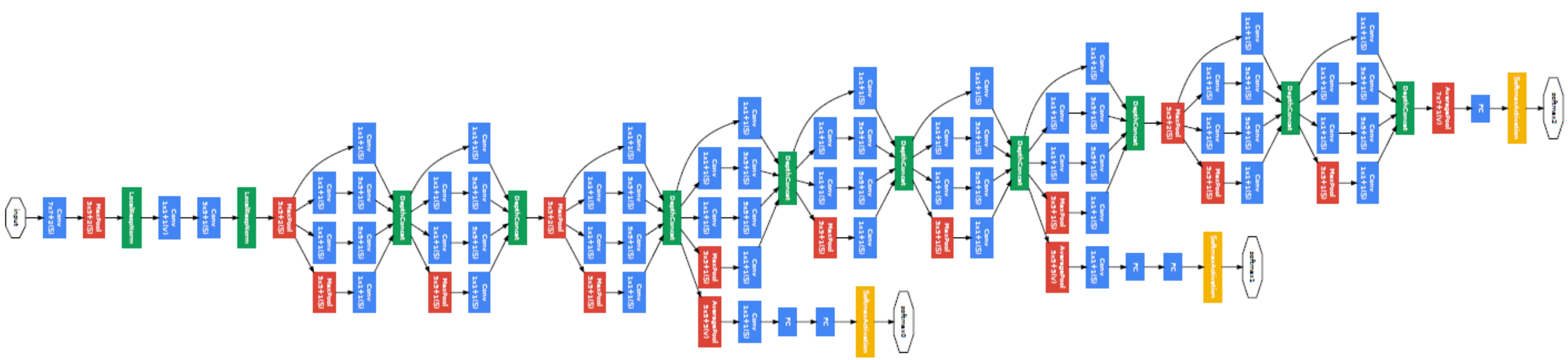
It combines (+ non-linearity) all activations across feature maps and makes a smaller set: this can be seen as an **embedding** procedure

Approximate the expected sparse structure of data

GoogLeNet

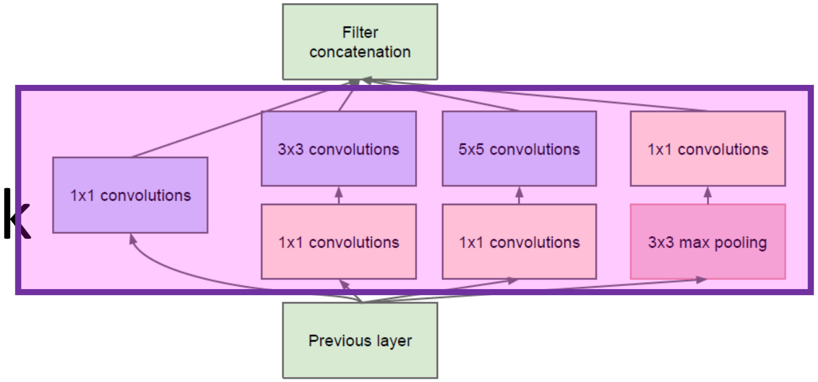
- **Inception** network

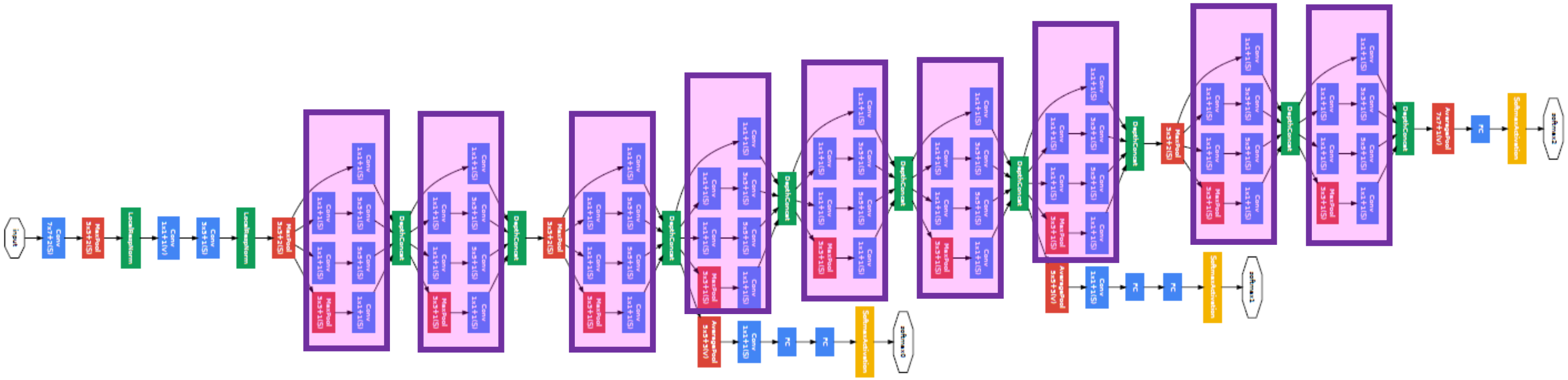




GoogLeNet

- **Inception** network





GoogLeNet

- Results on ILSVRC-2014

Team	Year	Place	Error (top-5)	Uses external data
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Table 2: Classification performance

architecture

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Table 1: GoogLeNet incarnation of the Inception architecture

Diabetic Retinopathy Kaggle challenge

Completed • $100,000 • 661 teams

Diabetic Retinopathy Detection

Tue 17 Feb 2015 – Mon 27 Jul 2015 (46 days ago)

Dashboard

Home
- Data
- Make a submission

Information
- Description
- Evaluation
- Rules
- Prizes
- References
- Timeline

Forum

Scripts
- New Script

Leaderboard
- Public
- Private

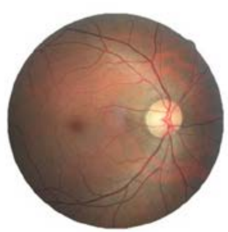
Leaderboard

1. Min-Pooling
2. o_O
3. Reformed Gamblers
4. Julian de Wit & Daniel Hammack
5. Jeffrey De Fauw
6. DeepSense.io
7. Zhanpeng Zhang
8. Plankton
9. Halla Yang
10. Sungbin Choi

Competition Details » Get the Data » Make a submission

Identify signs of diabetic retinopathy in eye images

Diabetic retinopathy is the leading cause of blindness in the working-age population of the developed world. It is estimated to affect over 93 million people.



The US Center for Disease Control and Prevention estimates that 29.1 million people in the US have diabetes and the World Health Organization estimates that 347 million people have the disease worldwide. Diabetic Retinopathy (DR) is an eye disease associated with long-standing diabetes. Around 40% to 45% of Americans with diabetes have some stage of the disease. Progression to vision impairment can be slowed or averted if DR is detected in time, however this can be difficult as the disease often shows few symptoms until it is too late to provide effective treatment.

Currently, detecting DR is a time-consuming and manual process that requires a trained clinician to examine and evaluate digital color fundus photographs of the retina. By the time human readers submit their reviews, often a day or two later, the delayed results lead to lost follow up, miscommunication, and delayed treatment.

Clinicians can identify DR by the presence of lesions associated with the vascular abnormalities caused by the disease. While this approach is effective, its resource demands are high. The expertise and equipment required are often lacking in areas where the rate of diabetes in local populations is high and DR detection is most

FRACTIONAL MAX-POOLING

Ben Graham
Department of Statistics
University of Warwick CV4 7AL, UK
b.graham@warwick.ac.uk

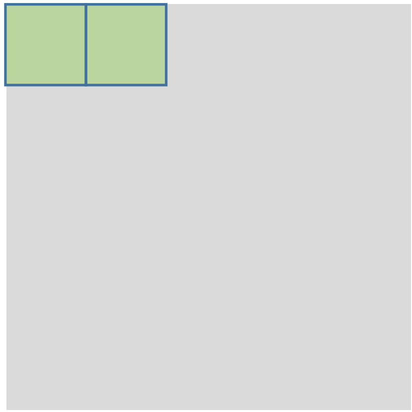
ABSTRACT

Convolutional networks almost always incorporate some form of spatial pooling, and very often it is $\alpha \times \alpha$ max-pooling with $\alpha = 2$. Max-pooling act on the hidden layers of the network, reducing their size by an integer multiplicative factor α. The amazing by product of discarding 75% of your data is that you build into the network a degree of invariance with respect to translations and elastic distortions. However, if you simply alternate convolutional layers with max-pooling layers, performance is limited due to the rapid reduction in spatial size, and the disjoint nature of the pooling regions. We have formulated a *fractional* version of max-pooling where α is allowed to take non-integer values. Our version of max-pooling is *stochastic* as there are lots of different ways of constructing suitable pooling regions. We find that our form of fractional max-pooling reduces overfitting on a variety of datasets: for instance, we improve on the state of the art for CIFAR-100 without even using dropout.

Fractional Max-pooling

- In classical max-pooling, the ratio between input dimension(s) and output dimension(s) is an integer
  - Max-pooling (2,2): N_{in} = 36, N_{out} = 18

We take the maximum within a 2x2 window with a step of 2



- The idea of fractional pooling is that we can have $N_{in}/N_{out} \in (1,2)$

Fractional Max-pooling

- Example, if we want N_{in} = 25 and N_{out} = 18, we can take the max considering the following steps:

18 max values

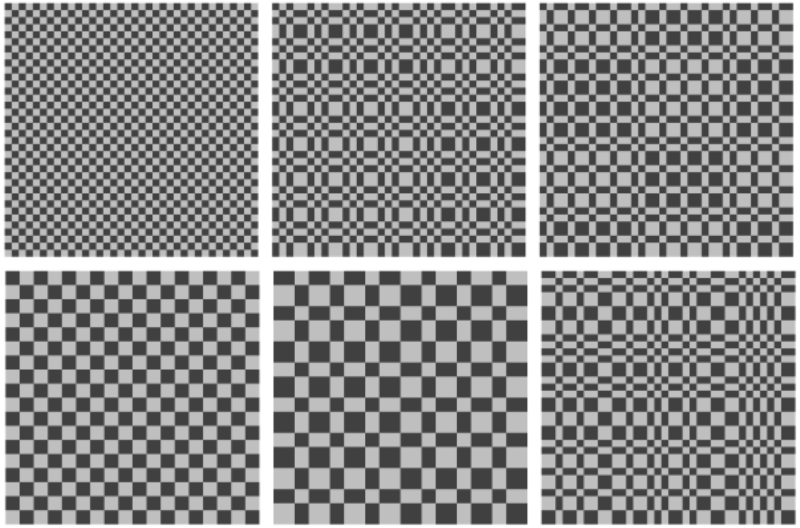
Example 1	211112112211112122	Covers 25 pixels
Example 2	111222121121112121	
Example 3	121122112111211212	

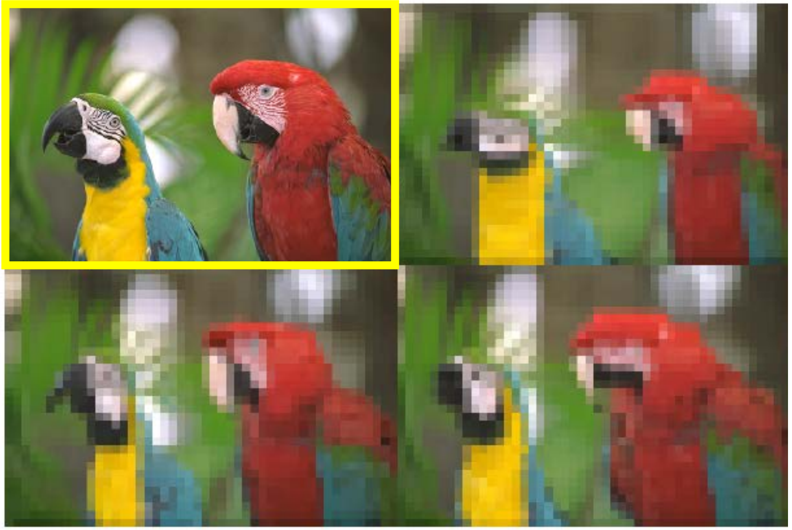
The 2s and 1s can be generated **randomly**

$N_{in}/N_{out} = 1.38$

Fractional Max-pooling

- Examples of pooling





Fractional Max-pooling

- Results

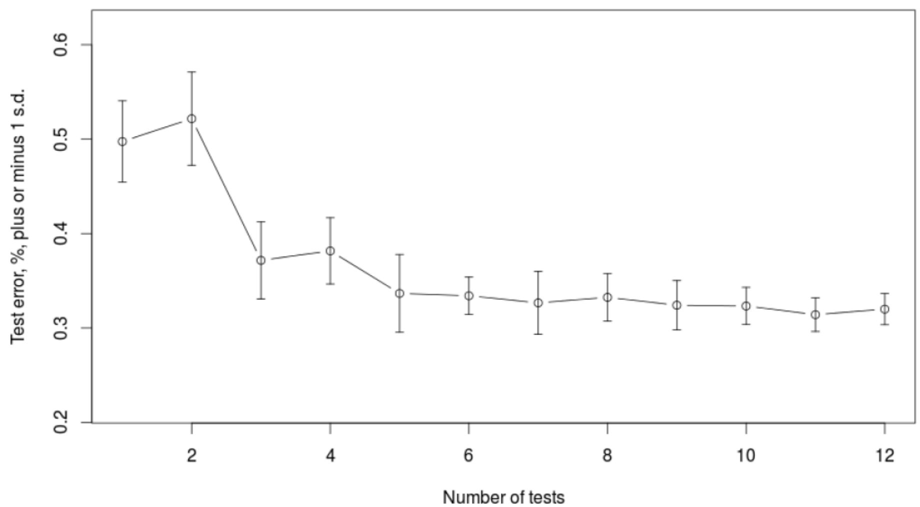


Figure 4: The effect of repeat testing for a single MNIST trained FMP network.

Several random poolings are generated, therefore several networks are obtained

The final result is the **average** of all the predictions

Improves on state of the art on **CIFAR-100 without using dropout**