A Survey of topologies of the Deep Convolutional Neural Networks used for image classification

Yevgen Kuzmenko
Universität Konstanz
yevgen.kuzmenko@uni-konstanz.de

ABSTRACT

This survey provides a description of the evolution of convolution networks starting from the year 1960, when the scientists found convolution patterns in the cat's brain, till 2017 when Machine Learning (ML) researchers exploit the ideas of convolutions and build wide and deep artificial neural networks (NN).

The goal of this overview is to demonstrate the flow of the thought of the NN researchers and give a better understanding of the state-of-the-art in deep learning algorithms in computer vision.

This work gives an overview of the recent research papers and provides a comprehensive summary of the main relevant models and discuss the possible ways of development in the area of computer vision and neural networks.

1. INTRODUCTION

Neither in science nor technology, there isn't one area that would experience as many ups and downs as the creation and development of artificial intelligence. Sometimes this area was overestimated, sometimes completely disappointing.

Until the second half of the 2000s, the term "artificial intelligence" in the scientific environment was considered indecent. Scientists have found an elegant way out of the situation, using the term "AI winter."[1] The AI winter was the time in which the research was not well funded into AI. It emerged out of the DARPA report on machine translation. Machine translation got canceled, and the funding for it nearly got dumped completely. This era ended with the advent of deep learning algorithms. Access to large amounts of data, cheaper and better GPU and improved algorithms stimulated the growth of deep learning (DL).

In 2012 happens an epoch-making event - deep convolutional neural network wins ImageNet challenge[2]. With the result two times better than the algorithm on the second place (16,4% vs. 25,8% top5 error)[3].

Five Years passed since that event. It may look like five years is not a big period, but a lot has changed in the area of artificial intelligence since then. To show the development of the field - the current top-end result on ImageNet is 2,3%[3]. NNs have got the attention of the scientist all over the world and found implementation in almost every industry.

This paper describes the evolution of convolutional neural networks (CNN), shows the early stages of research of Visual AI and discusses possible future of NN. This Survey aims to be a good entry point for researchers and developers who are thinking about implementing deep learning algorithms in their work. It helps to get general understanding of how the CNN work and gives an idea of it's possible future development.

1.1 Structure overview

The structure of this survey represents the timeline, where each section describes single paper and single topology. This structure should help a reader to follow the thoughts of the researchers in their pursuit of increasing the accuracy and decreasing the number of parameters needed to be calculated.

The focus of this survey is on the convolutional networks, because they are most popular example of the feed forward networks and serve as the foundation for understanding other techniques like recurrent networks and reinforcement learning.

The models also represent a logical development of CNN by correcting weaknesses and adding new features to the existing architectures.

For this survey was taken the papers to every model which showed outstanding results in ImageNet competition .

2. SURVEY

2.1 The transition from neurophysiology to computer vision

The review should begin with the pioneers of the field of convolutional neural networks (not only artificial ones) and their contribution: David Hubel and Thorsten Wiesel, the Nobel laureates of 1981. They received a prize for their

[1]AI Expert Newsletter: W is for Winter; 9.11.2013 at the Wayback Machine. http://www.ainewsletter.com/newsletters/aix_0501.htm#w

[2]The ImageNet project is a large visual database designed to be used in visual object recognition software research. Since 2010, the ImageNet project runs an annual software contest, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) http://image-net.org/about-publication

[3]http://image-net.org/challenges/LSVRC/2017/results

work done in 1959. The prize was awarded for "for their discoveries concerning information processing in the visual system."[4]

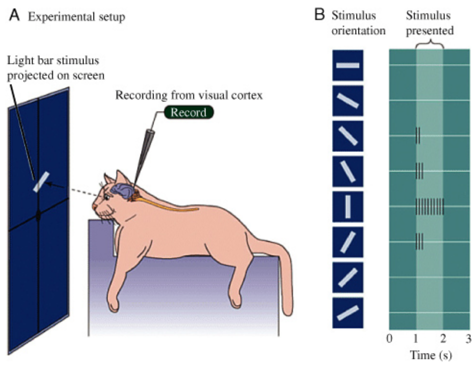


Figure 1: Hubel and Wiesel experiment illustration.[6]

The model of the experiment is shown on the figure 1. On a dark screen at different angles demonstrated a bright elongated moving rectangle; The oscilloscope electrode is connected to the occipital part of the brain, where the mammal has a visual information processing center. During the experiment, scientists observed the following effects (analogies with modern convolutional neural networks can be found.):

- Certain areas of the visual cortex are activated only when the line is projected onto a particular part of the retina;
- The level of activity of neurons in the region changes with the angle of inclination of the rectangle;
- Some areas are activated only when the object is moving in a certain direction.

One of the results of the study was a model of the visual system, or a topographic map, with the following properties [10]:

- Neighboring neurons process signals from neighboring regions of the retina;
- Neurons form a hierarchical structure, where each next level highlights more and more high-level signs;
- The neurons are organized in so-called columns - the computational blocks that transform and transfer information from level to level.

The first who tried to shift the ideas of Hubel and Weisel to the program code was Kunihiko Fukushima, who in the period from 1975 to 1980 proposed two models: a cognitron [4] and a neocognitron [5]. These models almost repeated the biological model, today we call simple cells convolutions, and complex cells (pool cells) are called pulling: they are the building blocks of modern convolutional neural networks. The model was not trained with backpropagation of the error, but by the original heuristic algorithm without a teacher. We can assume that this work was the beginning of neural network computer vision.

[4] http://www.nobelprize.org/nobel_prizes/medicine/laureates/1981/
[6] http://www.informit.com/articles/article.aspx?p=1431818

2.2 LeNet. First working convolutional network (1998)

Many years later in 1998, after so-called "AI winter" has passed. Yann LeCun, who was a post-doc of Geoffrey Everest Hinton, the author of the article on the algorithm for backpropagation of an error[17], publishes the paper **"Gradient-based learning applied to document recognition"** [13]. In this article he mixes the ideas of convolutions and pooling with backpropagation, eventually obtaining the first working convolutional neural network. US Post service used it for recognition of post zip indexes. This architecture was the standard template for building convolutional networks until recently: the convolution layer alternates with the pulling layer a few times, then several fully connected layers. (See figure 2)

This network contains 340 908 connections and 60 000 trainable parameters. The basic building blocks are 5×5 convolutions with stride 1 and 2×2 with stride 2. Convolutions play the role of feature detectors, and pooling (or Subsampling, as it's called in the paper) is used to reduce the dimension by exploiting the fact that neighboring pixels do not differ much from each other; thus, the information loss will be insignificant.

2.3 AlexNet. The network which has got all the attention (2012)

Based on the paper **"ImageNet Classification with Deep Convolutional Neural Networks"**[12]

Another 14 years have passed. Alex Krizhevsky, a student of Hinton, from the same laboratory, where LeCun was a postdoc, added the last ingredients to the formula. Deep training = model + learning theory + large data + computational power. With new GPU it became possible to increase the number of learning parameters significantly. This model has eight levels: five convolutional and three fully-connected and contains 60 million parameters. Two graphics accelerators were used to train this model.

From network topology (see figure 3), this is almost the same LeNet, just increased a thousand times. Several more convolution layers were added, and the size of convolution kernels decreases from the input of the network to the output. This is explained by the fact that at the beginning the pixels are strongly correlated. Next, they apply pooling, thereby increasing the density of uncorrelated regions. At the next level, they take a slightly smaller receptor region. As a result, the authors obtained a pyramid of the convolutions $11 \times 11 \to 5 \times 5 \to 3 \times 3$.

To avoid overfitting they applied new techniques, some of which are now standard for deep networks:

- DropOut[8] (The key idea is to randomly drop neurons with their connections from the network during training.) (Batch Normalization [11] is used in the modern networks in addition to, or instead of DropOut)
- Data Augmentation [18] (Image translations, horizontal reflections, and patch extractions)
- ReLu [15] (They found that using ReLU as activation functions decrease training time several times comparing the conventional tanh function)

AlexNet wins an ImageNet challenge in 2012. And not just wins, but shows an error of almost half the second place

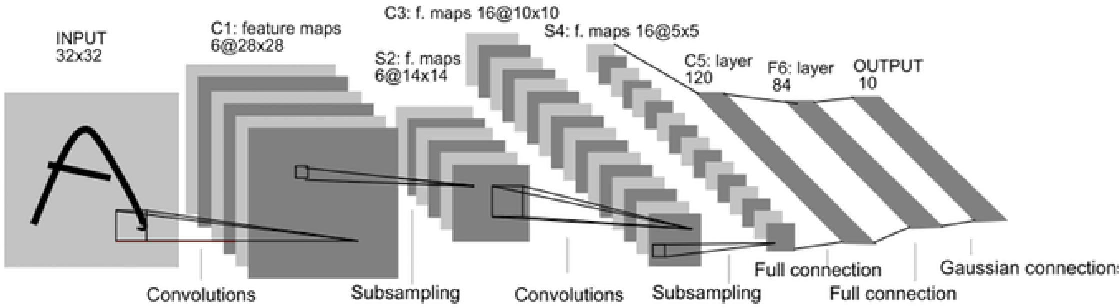


Figure 2: Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane us a feature map, i.e. a set of units whose weights are constrained to be identical.

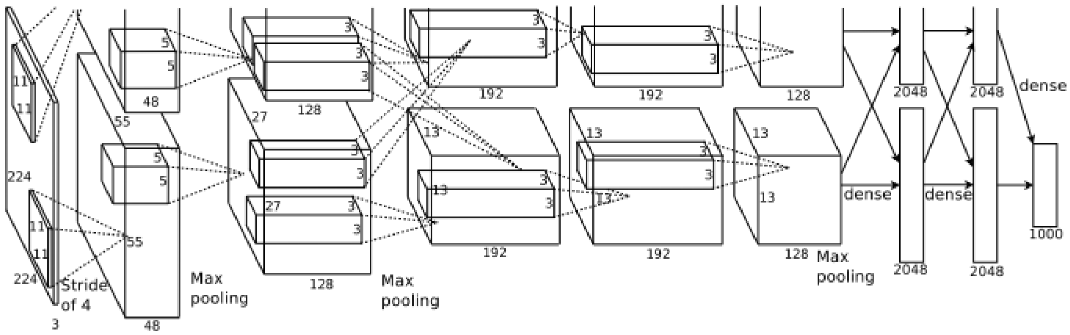


Figure 3: An illustration of the architecture of AlexNet, explicitly showing the delineation of responsibilities between the two GPUs.

(16,4% vs. 25.8% top5 error[7]). Since 2012, not one non-neural network model have won in the ImageNet competition.

Alex net has a simple to understand architecture and very popular even nowadays. It has 60 mln parameters, so it is computationally expensive to train it from scratch, but it is easy to find trained model for every existing framework and then just fine tune it to fit the specific needs.

2.4 VGG. New Convolution filters (12 Apr 2014)

In 2014 there were two interesting papers, this one: **"Very Deep Convolutional Networks for Large-Scale Image Recognition"** [19] and Google Inception, which we will consider below. VGG is the last model, which use the pattern of topology laid by LeCune. Their model VGG-19 consists of 19 layers, 144 million parameters and adds to the architecture one simple idea. Let's take, for example, the convolution of 5×5, this is a mapping: $f : \mathbb{R}^{25} \to \mathbb{R}$. It contains 25 parameters. If we replace it with a stack of two layers with 3×3 convolutions, then we get the same mapping, but the number of parameters will be less: $3 \times 3 + 3 \times 3 = 18$, which is 22% less. If we replace 11×11 with four 3×3 convolutions, this is 70% fewer parameters.

It worked well on both image classification and localization tasks.

Although this network doesn't deliver outstanding results (comparing to the new very deep networks), many researchers use it in their work, because the architecture of the network is simple to understand and therefore it's easier to change according to the purpose.

2.5 GoogLeNet. Inception model (17 Sep 2014)

Based on the paper **"Going Deeper with Convolutions"** [21]

An arbitrary increase in the width (the number of neurons in the layers) and the depth (the number of layers) has many drawbacks. First, an increase in the number of parameters leads to overfitting, and an increase in the number of layers also adds a vanishing gradient problem[8] [2]. Secondly, an increase in the number of convolutions in the layer leads to a quadratic increase in the computations in this layer. If new model parameters are used ineffectively, for example, many of them become close to zero, then we only waste computer processing power. Despite these problems, the scientists from Google wanted to experiment with deep networks. For this, they turned to the theoretical work "Provable Bounds for Learning Some Deep Representations" by Arora et al. [1], in which it is proved that if the probability distribution of data can be represented as a sparse, deep and wide neural network, then it is possible to construct an optimal neural network for a given dataset, analyzing neuron correlations of the previous layer and combining the correlated neurons into groups that will be the neurons of

[7] http://image-net.org/challenges/LSVRC/2012/results.html

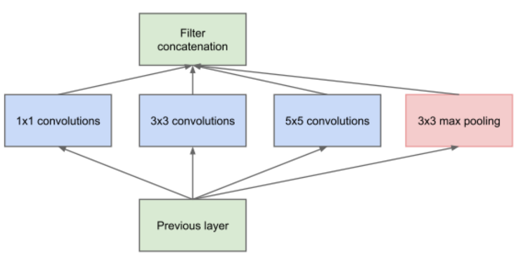
[8] This problem makes it really to learn the parameters of the earlier layers in the network. This problem becomes worse as the number of layers in the architecture increases.

the next layer. Thus, the neurons of the late layers "look" at the strongly intersecting regions of the original image. The main ideas they have are:

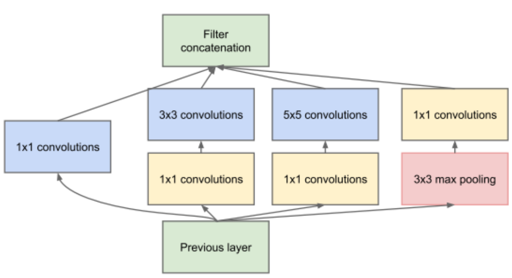
- The original AlexNet did large convolutions that require a lot of parameters, so they tried to make smaller convolutions with more layers.
- Then they will aggressively reduce the number of dimensions to compensate for thicker layers. It is possible to do this with the help of 1x1 convolutions - it is a linear filter which applied throughout the picture to take the current number of dimensions, and linearly mix them into smaller ones. Since it also learns, it turns out very efficiently.
- At each level, they run several convolution kernels of different sizes to get features of different scale. If the scale is too large for the current level, it is recognized in the next.
- It was found that a move from fully connected layers to average pooling improved the top-1 accuracy by about 0.6%.

In order not to lose the original image from the previous layer, in addition to the convolutions, authors also added pulling operations. The entire group of several operations is called the Inception block or Inception module.

> "Additionally, since pooling operations have been essential for the success in current state of the art convolutional networks, it suggests that adding an alternative parallel pooling path in each such stage should have additional beneficial effect, too."[21]



(a)



(b)

Figure 4: Incepton module: (a) Naive version; (b) With dimension reduction;

To equalize the size of output tensors, it is also suggested to use a 1×1 convolution. You can see it in the figure 4 to the right after the operation of the pooling. Besides, 1×1 convolutions are also used to reduce the dimension before energy-intensive convolution operations. Google uses 1×1 convolution to achieve such goals (the next generation of networks will also exploit these techniques):

- Reduce the dimension *before the operation*;
- Increase in dimension *after operation*;
- Grouping of correlated values (the first operation in the block).

The resulting model is on the figure 5.

Also they put a few additional classifiers at different levels. The initial idea was that such classifiers would allow to "push" the gradients to the early layers and thereby reduce the effect of vanishing gradient problem. Later Google will give up on them, because the network itself gradually increases the depth of the tensor and reduces the spatial dimension.

The authors claim that GoogLeNet actually uses $12\times$ fewer parameters than the winning architecture of Krizhevsky et al [12] from two years ago, while being significantly more accurate.

At the end of the article, the authors leave open the question of the effectiveness of such a model, as well as hinting that they will explore the possibility of automatic generation of network topologies, using the above principles.

2.6 Inception v2. Rethinking the Architecture of Inception Block. (11 Dec 2015)

In the new paper **"Rethinking the Inception Architecture for Computer Vision"** [22], the authors explored in practice different architectures and developed **four principles for constructing deep convolutional neural networks:**

1. Avoid representational bottlenecks: do not drastically reduce the dimensionality of data representation. This should be done gently from the beginning of the network and to the classifier at the output.
2. High-dimensional representations should be processed locally within a network. Increasing the activations per tile in a convolutional network allows for more disentangled features. The resulting networks will train faster.
3. Spatial aggregation can and should be factored into lower dimensional embeddings without much or any loss in representational power. This will save resources which they use to increase the size of the network. Given that these signals should be easily compressible, the dimension reduction even promotes faster learning.
4. It is necessary to balance the depth and width of the network. Do not dramatically increase the depth of the network separately from the width, and vice versa; Evenly increase or decrease both dimensions.

The authors of VGG model in 2014 showed that the big convolutions could be factored into a stack of convolutions 3×3. Google exploit this idea and factored all convolutions into $N \times 1$ and $1 \times N$ (Figure 6).

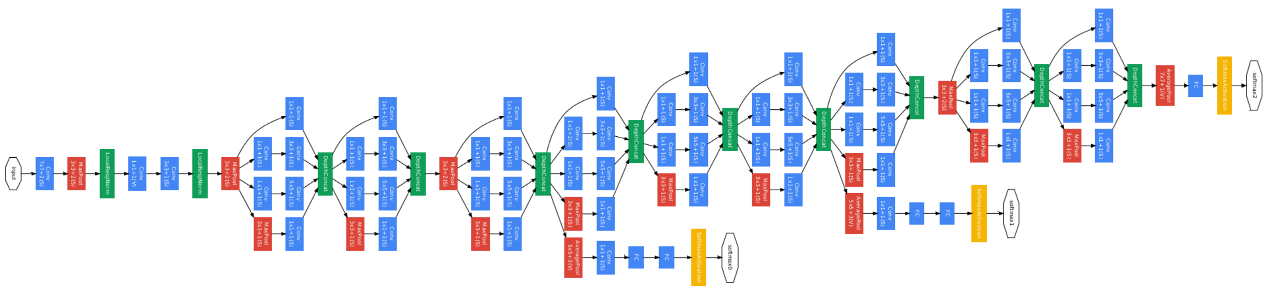
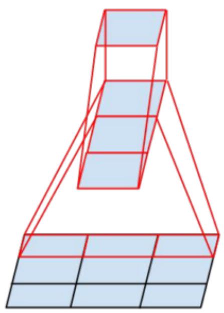
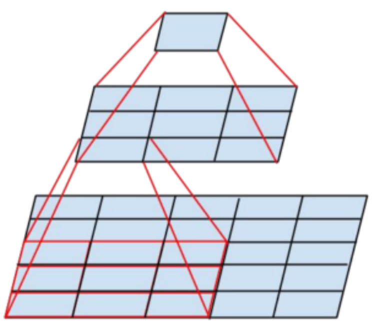


Figure 5: GoogleNet.



(a)



(b)

Figure 6: (a)Mini-network replacing the 5×5 convolutions;(b) Mini-network replacing the 3×3 convolutions. The lower layer of this network consists of a 3×3 convolutions with 3 output units.

Concerning efficient grid size reduction, to avoid a representational bottleneck, before applying maximum or average pooling the activation dimension of the network filters is expanded. For example, if the input dimensionality is

$$d \times d \times k$$

(k is the number of filters) and we want to get

$$\frac{d}{2} \times \frac{d}{2} \times 2k$$

We first compute a stride-1 convolution with $2k$ filters and then apply an additional pooling step. The complexity of such operation is:

$$2d^2k^2$$

If we do first pooling and then convolution - the complexity will drop to:

$$2\left(\frac{d}{2}\right)^2 k^2$$

but then the first principle will be violated resulting in less expressive networks.

Illustration to this two alternatives is on the figure 7.

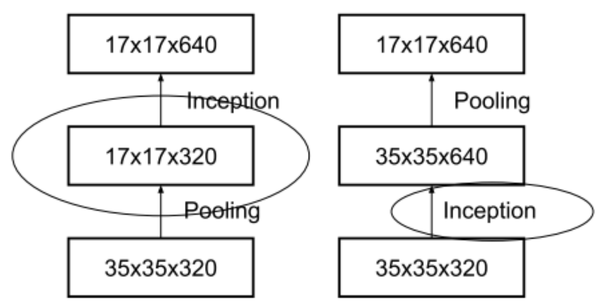


Figure 7: Two alternative ways of reducing the grid size. The solution on the left violates the principle 1 of not introducing an representational bottleneck. The version on the right is 3 times more expensive computationally.

It is proposed to increase the number of parallel branches the following way: do the convolutions with stride 2, but at the same time increase the number of channels twice, then the representative power of the representation decreases ”smoother.” And to manipulate the depth of the tensor they use convolutions 1×1.

Finally, they slightly modify the blocks for the last layers, so that they are wider, though less deep.

The network has a few convolutions at the beginning and then 11 inception layers concatenated to each other.

The network is 42 Layers deep, but computation cost is only about 2.5 higher than that of GoogLeNet(Inception v1), and it is still much more efficient than VGGNet.

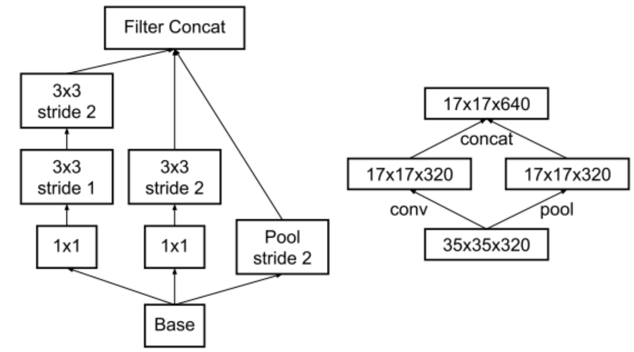


Figure 8: Inception module that reduces the grid-size while expands the filter banks. It is both cheap and avoids the representational bottleneck as is suggested by principle 1. The diagram on the right represents the same solution but from the perspective of grid sizes rather than the operations.

They call the main architecture Inception v2 (Figure 9), and the version where additional classifiers work with Batch Normalization [11] - Inception v3.

Inception v3 reaches 4.2% top5 classification error on ImageNet, and the ensemble of four models - 3.58%.

2.7 ResNet. More Layers = Better quality (10 Dec 2015)

It has long been noted that if more layers simply stack to each other, then the quality of such a model grows to a certain limit (see VGG-19 [19]), and then begins to fall. This problem is called **degradation problem** [9] , and the networks obtained by the concatenation of many layers are plain networks. Kaiming He and his partners from Microsoft Research Asia, the authors of **"Deep Residual Learning for Image Recognition"** [7] were able to find a topology in which the quality of the model grows with the addition of new layers (See Figure 10).

The trivial way to get a deeper network, with the not worse result than before is to add more *identity* layers. This observation, that you can always do not worse then *identity*, is the main idea of ResNets. Let $H(x)$ be the true function we want to learn, but make the network to learn residual function $F(x) := H(x) - x$. The original function is then $H(x) = F(x) + x$. (See Figure 11).
If we take a network, for example, VGG-19, and attach twenty more layers to it, then we would like the new deep network to behave at least as good as its shallow counterpart. The **problem of degradation** implies that a complex nonlinear function $F(x)$, obtained by the stacking of several layers, must learn the *identity* transformation, if the previous layer has reached the quality limit, but this does not happen for some reason. The authors want to help it by adding a shortcut connection because they consider, it will be easier for the optimizer to make all weights close to zero, rather than fit an *identity* mapping. (See Figure 11).

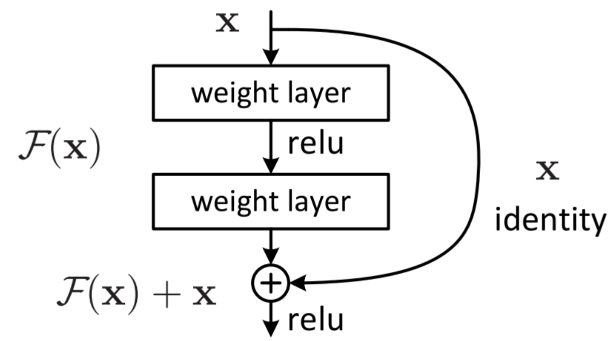


Figure 11: Residual learning: a building block.

To prove their theory, the authors build a simplified VGG model with 34 layers with fewer filters, lower complexity and added shortcut connections. And it appears to give even better results than original VGG-34!(See Figure 12)

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Figure 12: Top-1 error (%, 10-crop testing) on ImageNet validation. Here the ResNets have no extra parameter compared to their plain counterparts.

TO exploit this success they added more layers. To be able to use more layers they must be lighter, so they use instead of two convolutional layers - only one and thinner, see figure 13.

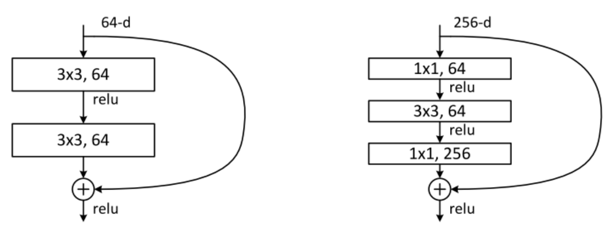


Figure 13: A deeper residual function F for ImageNet. Left: a building block (on 56×56 feature maps) Right: a "bottleneck" building block for ResNet-50/101/152

This way the number of parameters will decrease drastically and the authors were able to create networks with 101 and 152 layers, but still with fewer parameters than in VGG! The ensemble of six models of different depth got the result of impressive 3.57% top-5 error on the test set. This entry won the 1st place in ILSVRC 2015.

[9] https://github.com/tensorflow/models/tree/master/inception

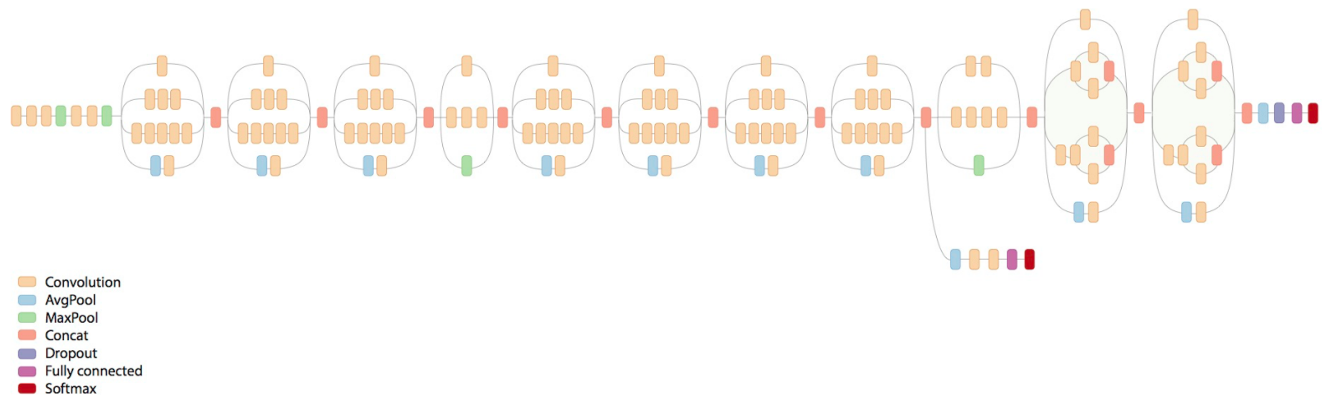


Figure 9: Inception-v2. [9]

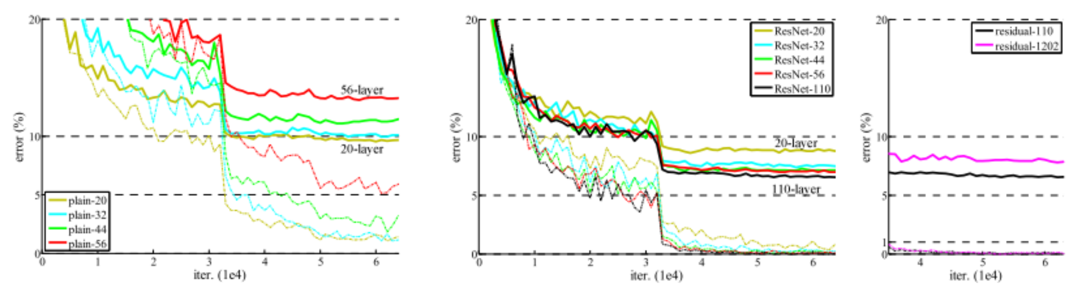


Figure 10: Training on CIFAR-10. Dashed lines denote training error, and bold lines denote testing error. Left: plain networks. The error of plain-110 is higher than 60% and not displayed. Middle: ResNets. Right:ResNets with 110 and 1202 layers.

method	top-5 err. (**test**)
VGG [40] (ILSVRC'14)	7.32
GoogLeNet [43] (ILSVRC'14)	6.66
VGG [40] (v5)	6.8
PReLU-net [12]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Figure 14: Error rates (%) of *ensembles*. The top-5 error is on the test set of ImageNet.

After the publication of this paper, there was a lot of research and discussion of the ResNet mechanics.

In two months after ResNet paper Google publishes a paper **"Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning"** by Szegedy at al [20] in which they show that the Inception model will work better if they add *identity* connections. In this paper, they developed two architectures: Inception-v4 (upgraded Inception model without residual connections) and Inception-ResNet-2 (Inception-v4 with residual connections) The ensemble of one Inception-v4 and three ResNet-2 achieves new record 3.08% top-5 error the test set of the ImageNet Classification (CLS) challenge.

In **"Wide Residual Networks"** by Zagoruyko et al. [25] the authors use the idea of ResNets to train very wide and not deep networks. They report that their model outperform ResNet-152, having three times fewer layers.

Veit et al. did interesting research in **"Residual Networks are Exponential Ensembles of Relatively Shallow Networks"** [23] were the authors perform a couple of interesting experiments and claimed that the ResNet is an ensemble by its design.

3. CONCLUSIONS

Since the last deep learning algorithms made such significant progress in the field of computer vision, it becomes much more challenging to move on top of that. One of the possible ways of developing more powerful models is to implement the methods of fusion and use ensembles. In ImageNet competition in 2016 won the ensemble model, which didn't bring anything radically new to the progress of NN architectures and the result is only a half percent better than the last year.

There are other new promising directions in NN, for example, Generative adversarial networks[6], Attention networks[24], and Evolving Deep Neural Networks[14]. Maybe, providing more **understanding** of the object is the next step in computer vision.

The main disadvantage which can stop one of using NN in computer vision is that they are working as a black box. It's hard to determine how exactly they make a decision, which also makes them not useful for understanding the problem. This makes NN vulnerable to so-called "Black-Box Attacks" [16], because of this, it's hard to trust NN and use them for such task as autopilot or health-care. The training of the networks very depends on the choice of initial parameters, which make them difficult to train.

Despite all the problems and the black box problem of the neural networks - there is no going back to using algorithms for computer vision. No algorithm could give such an outstanding result in object recognition and segmentation.

4. REFERENCES

[1] S. Arora, A. Bhaskara, R. Ge, and T. Ma. Provable Bounds for Learning Some Deep Representations. 32, 2013.

[2] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, Mar 1994.

[3] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, F. Li, S. Stony, B. Stanford, and S. Mit. Large Scale Visual Recogni1on Challenge 2012. *Lsvr*, 2012.

[4] K. Fukushima. Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, 20(3-4):121–136, 1975.

[5] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

[6] I. Goodfellow, J. Pouget-Abadie, and M. Mirza. Generative Adversarial Networks. *arXiv preprint arXiv: . . .*, pages 1–9, 2014.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[8] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. pages 1–18, 2012.

[9] S. Hochreiter. *Untersuchungen zu dynamischen neuronalen Netzen*. PhD thesis, diploma thesis, institut für informatik, lehrstuhl prof. brauer, technische universität münchen, 1991.

[10] D. H. Hubel and T. N. Wiesel. Receptive Fields and Functional Architecture of monkey striate cortex. *Journal of Physiology*, 195:215–243, 1968.

[11] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015.

[12] A. Krizhevsky, I. Sutskever, and H. Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25 (NIPS2012)*, pages 1–9, 2012.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998.

[14] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, and B. Hodjat. Evolving Deep Neural Networks. 2017.

[15] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[16] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical Black-Box Attacks against Machine Learning. 2016.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, oct 1986.

[18] P. Simard, D. Steinkraus, and J. Platt. Best practices for convolutional neural networks applied to visual document analysis. *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, 1(Icdar):958–963, 2003.

[19] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. 2016.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:1–9, 2015.

[22] C. Szegedy, V. Vanhoucke, J. Shlens, and Z. Wojna. Rethinking the Inception Architecture for Computer Vision. 2014.

[23] A. Veit, M. Wilber, and S. Belongie. Residual Networks Behave Like Ensembles of Relatively Shallow Networks. pages 1–9, 2016.

[24] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical Attention Networks for Document Classification. *Proceedings of NAACL-HLT 2016*, pages 1480–1489, 2016.

[25] S. Zagoruyko and N. Komodakis. Wide Residual Networks. may 2016.

APPENDIX

A. TABLE

Name	Type of Network	Description	Year	Link to paper	Authors	ImageNet Resultat	LInk to GitHub	Number of Parameters	Main Points	Computational Power	Notes	Discussion
AlexNet	CNN (Basic model) Image Classification	Five convolutional layers + three fully connected layers	2012	ImageNet Classification with Deep Convolutional Neural Networks	Alex Krizhevsky	16.4% (Ensemble of 5 nets)	https://github.com/tensorflow/models/blob/master/slim/nets/alexnet.py	60 Mln	Trained the network on ImageNet data, which contained over 15 million annotated images from a total of over 22,000 categories. Used ReLU for the nonlinearity functions (Found to decrease training time as ReLUs are several times faster than the conventional tanh function). Used data augmentation techniques that consisted of image translations, horizontal reflections, and patch extractions. Implemented dropout layers in order to combat the problem of overfitting to the training data. Trained the model using batch stochastic gradient descent, with specific values for momentum and weight decay. 60 million parameters.	Trained on two GTX 580 GPUs for five to six days.	Won ILSVRC classification competitions in 2012.	
VGG	CNN (Basic model) Image Classification	Thirteen/Fifteen convolutional layers + three fully connected layers	2014	Very Deep Convolutional Networks for Large-Scale Image Recognition	Karen Simonyan, Andrew Zisserman	6.8% (2 nets)	https://github.com/machrisaa/tensorflow-vgg	144 Mln	The use of only 3x3 sized filters is quite different from AlexNet's 11x11 filters in the first layer. The authors' reasoning is that the combination of two 3x3 conv layers has an effective receptive field of 5x5. This in turn simulates a larger filter while keeping the benefits of smaller filter sizes. One of the benefits is a decrease in the number of parameters. Also, with two conv layers, we're able to use two ReLU layers instead of one. 3 conv layers back to back have an effective receptive field of 7x7. As the spatial size of the input volumes at each layer decrease (result of the conv and pool layers), the depth of the volumes increase due to the increased number of filters as you go down the network. Interesting to notice that the number of filters doubles after each maxpool layer. This reinforces the idea of shrinking spatial dimensions, but growing depth. Worked well on both image classification and localization tasks. The authors used a form of localization as regression (see page 10 of the paper for all details). Built model with the Caffe toolbox. Used scale jittering as one data augmentation technique during training. Used ReLU layers after each conv layer and trained with batch gradient descent.	Trained on 4 Nvidia Titan Black GPUs for two to three weeks.	Second Place in ILSVRC classification competitions in 2014.	
GoogLeNet	CNN (Basic model) Image Classification	Twenty-one convolutional layers + one fully connect- ed layer	2014	Going Deeper with Convolutions	Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich	6.7% (7 nets)	https://github.com/google/inception	5Mln	Used 9 Inception modules in the whole architecture, with over 100 layers in total! Now that is deep… No use of fully connected layers! They use an average pool instead, to go from a 7x7x1024 volume to a 1x1x1024 volume. This saves a huge number of parameters. Uses 12x fewer parameters than AlexNet. During testing, multiple crops of the same image were created, fed into the network, and the softmax probabilities were averaged to give us the final solution. Utilized concepts from R-CNN for their detection model. There are updated versions to the Inception module (Versions 6 and 7).	Trained on "a few high-end GPUs within a week". GoogLeNet ~3GFLOP (2x Inception v1)	Won ILSVRC classification competitions in 2014. Increased the depth and width without raising the computational requirements	
ResNet (Residual Networks)	CNN	Many more layers than traditional Deep Networks(18, 34, 152)	2015	Deep Residual Learning for Image Recognition	Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Microsoft Research	3.57% (Ensemble)	https://github.com/KaimingHe/deep-residual-networks	110 layers - 1.7M 1202 layers - 19.4M	"Ultra-deep" – Yann LeCun. 152 layers… Interesting note that after only the first 2 layers, the spatial size gets compressed from an input volume of 224x224 to a 56x56 volume. Authors claim that a naïve increase of layers in plain nets result in higher training and test error (Figure 1 in the paper). The group tried a 1202-layer network, but got a lower test accuracy, presumably due to overfitting.	Trained on an 8 GPU machine for two to three weeks. ResNet-152 ~12 GFLOP	Won ILSVRC 2015 with an incredible 3.57% top-5 error (Depending on their skill and expertise, humans generally hover around a 5-10% error rate)	
WRNs (Wide Residual Networks)	CNN, ResNet;	decrease depth and increase width of residual networks	2016	Wide Residual Networks	Sergey Zagoruyko, Nikos Komodakis	5.79% (Singel network)	wide-residual-networks	on the widest ropology - 36.5M	novel widened architecture for ResNet blocks that significantly improved performance of residual networks			

Figure 15: Artificial Neural Networks Zoo.