Explaining AlexNet Convolutional Neural Network

AlexNet was a game changer in the world of Visual Machine Learning

Rishi Sidhu
Jun 7, 2019 · 6 min read ★

AlexNet is the name of a convolutional neural network that won the LSVRC competition in year 2012. It was designed by **Alex** Krizhevsky, Ilya Sutskever and Krizhevsky's PhD advisor Geoffrey Hinton. Geoffrey Hinton, the winner of this year's $1M Turing award, was originally resistant to the idea of his student.

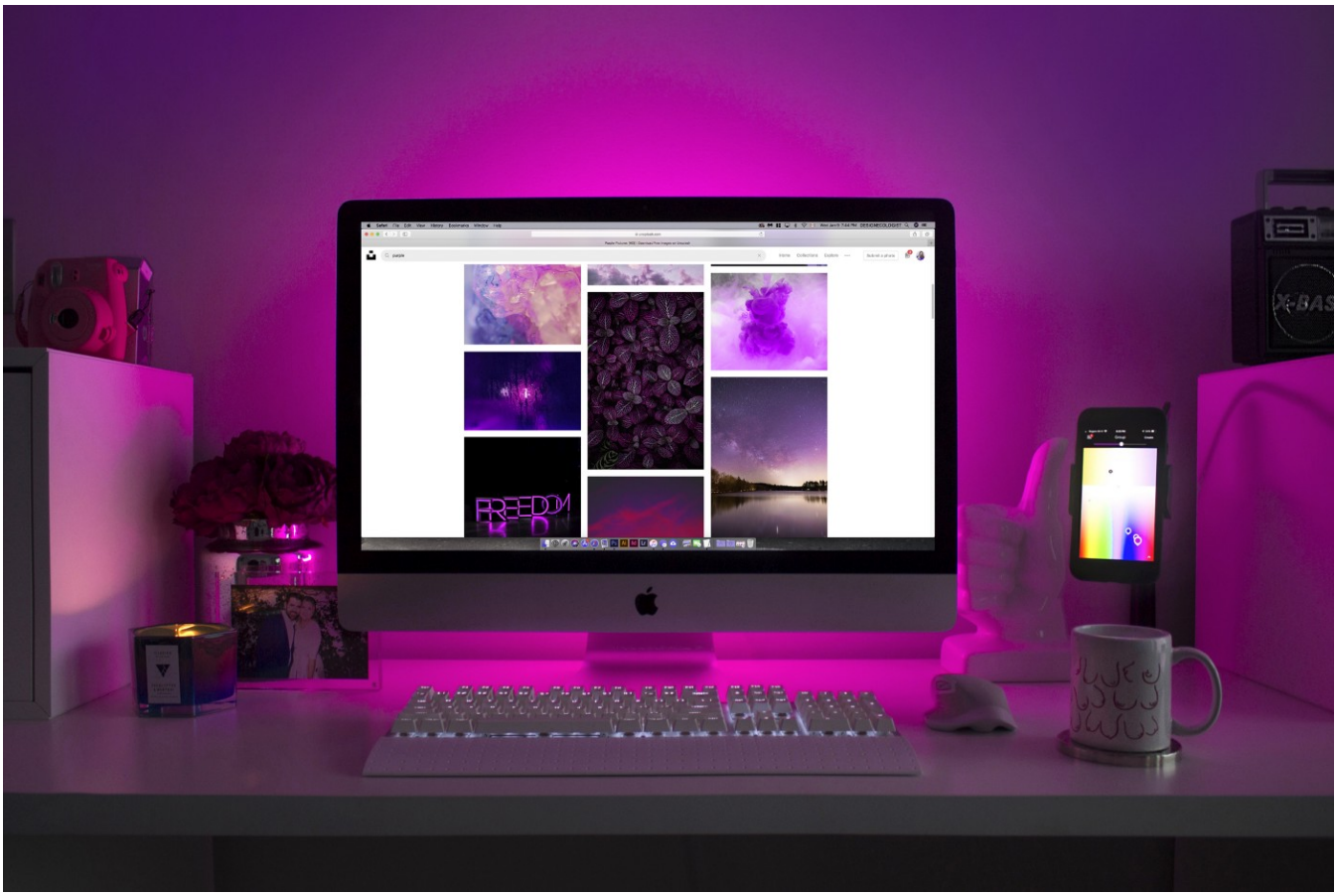
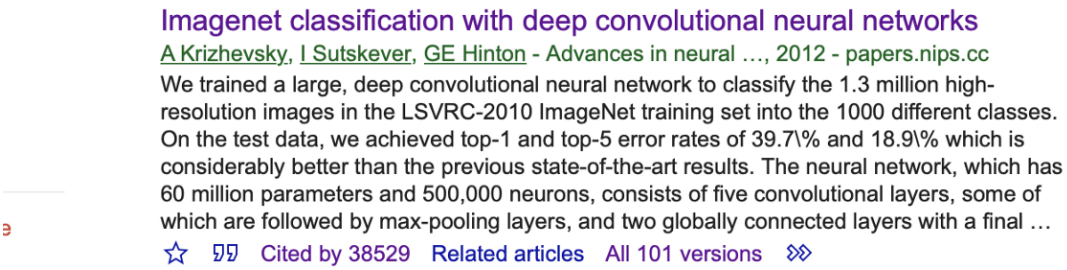


Photo by DESIGNECOLOGIST on Unsplash

LSVRC a.k.a. Large Scale Visual Recognition Challenge is a competition where research teams evaluate their algorithms on a huge dataset of labelled images (ImageNet), and

compete to achieve higher accuracy on several visual recognition tasks. The popularity of this paper can be seen through the number of its citations alone



38K Citations

. . .

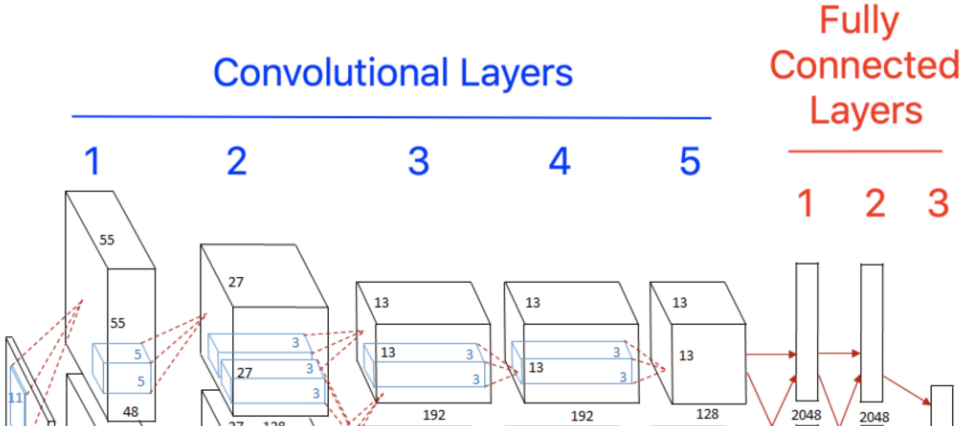
The devil is in the details

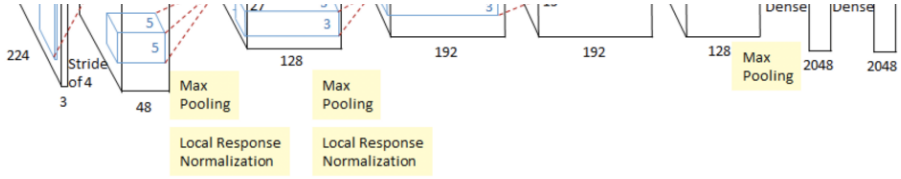
What sets AlexNet apart from all the other CNNs was that 1) it was one of the early ones 2) the paper employed a number of techniques that were unusual when it came to the state of the art, at that time. Let's take a look at some of the differentiating features of this paper.

. . .

Layers

AlexNet folks decided to not only try a very deep network they also tried some cross connections between layers. AlexNet contains **8 layers**. The first five are convolutional and the remaining three are fully-connected.





Source Sik-Ho Tsang's article

Connections between layers — Notice the 2 parallel paths in the image above. Each path processes a part of the data parallelly. Yet there are some cross connections between the paths e.g. between layer 2 and 3.

This cross-connection scheme was a neat trick which reduced their top-1 and top-5 error rates by 1.7% and 1.2%, respectively. This was huge given that they were already ahead of the state of the art.

7 Discussion

Our results show that a large, deep convolutional neural network is capable of achieving record-breaking results on a highly challenging dataset using purely supervised learning. It is notable that our network's performance degrades if a single convolutional layer is removed. For example, removing any of the middle layers results in a loss of about 2% for the top-1 performance of the network. So the depth really is important for achieving our results.

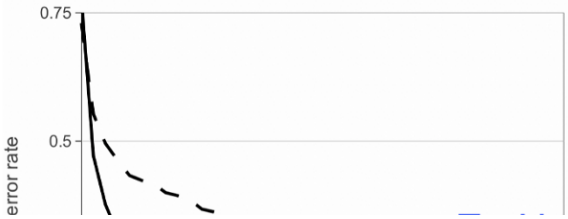
Discussion Section of the paper.

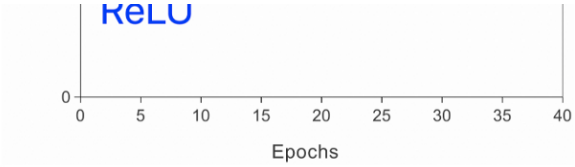
. . .

Non-Linearity

Experimenting with layers helped them a lot. The next thing they targetted was the sort of non-linear transformation that data would have to go through when entering a neuron.

AlexNet team chose a non linear activation function with the non-linearity being a **Rectified Linear Unit (ReLU)**. They claimed that it ran much faster than **TanH** the more popular choice for linearity at the time.





AlexNet proved that ReLU was much faster

The questions is why would we transform our data and that too in a non-linear fashion!

AlexNet is a Convolutional Neural Network. Hence it is bound to be made up of neurons. These b*iologically inspired neural networks* possess an activation function which decides whether the input stimulus is enough for a neuron to fire — i.e. get activated.

Without the non-linearity introduced by the activation function, multiple layers of a neural network are equivalent to a single layer neural network — the 8 layer depth would be useless without this.

The keyword being faster. Above all AlexNet needed a faster training time and ReLU helped them. But they needed something more. Something that could transform the speed with which CNNs were computed. This is where the GPUs figured.

. . .

GPUs and Training Time

GPUs are devices that can perform parallel computations. Remember how an average laptop is either a *Quadcore(4 cores)* or an *Octacore(8 cores)*. This refers to the number of parallel computations that can happen in a processor. A GPU can have 1000s of cores leading to a lot of parallelization. AlexNet made use of a GPU that NVIDIA launched a year before AlexNet came out.

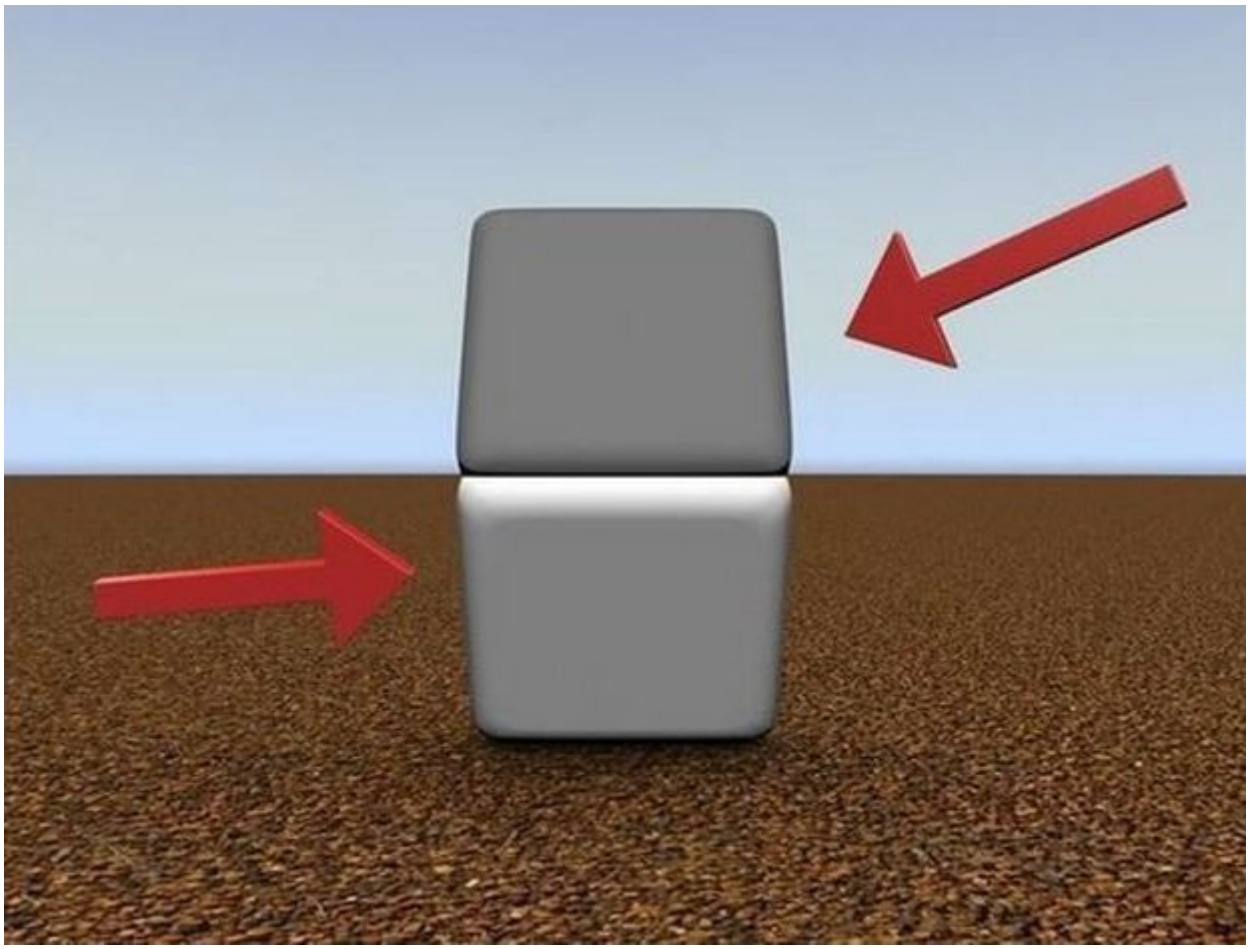
The noticeable thing was that AlexNet made use of 2 GPUs in parallel which made their design extremely fast.

. . .

Local Response Normalization

Not only did they need to speed up the processing they also needed the data to be *balanced.* So they used **local response normalization (LRN)**. It basically helps you normalize your data. AlexNet employed LRN to aid generalization. Response normalization reduced their top-1 and top-5 error rates by 1.4% and 1.2%, respectively.

The biological equivalent of LRN is called “lateral inhibition”. This refers to the capacity of an excited neuron to subdue its neighbors. The neuron does that to increase the contrast in its surroundings, thereby increasing the sensory perception for that particular are.



Lateral Inhibtion in action: The two blocks are of same color. Put a finger across the separating line and see for yourself.

. . .

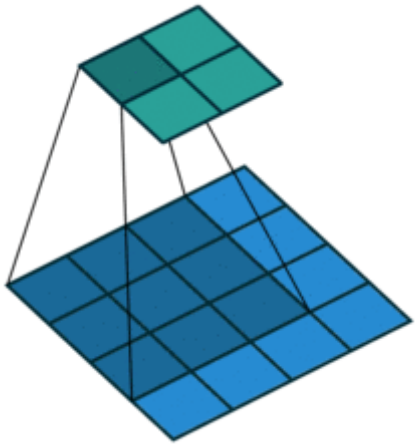
Overlapping Pooling

Every CNN has pooling as an essential step. Up until 2012 most pooling schemes

part of the process.

> Pooling is the process of picking a patch of s x s pixels and finding its max or mean.

Traditionally, these patches were non-overlapping i.e. once an ***s x s*** patch is used you don't touch these pixels again and move on to the next ***s x s*** patch. They realized that overlapping pooling reduced the top-1 and top-5 error rates by 0.4% and 0.3%, respectively, as compared with the non-overlapping scheme.



Overlapped Pooling — Source: https://goo.gl/nrMk2P

. . .

Overfitting Prevention

Having tackled normalization and pooling AlexNet was faced with a huge overfitting challenge. Their 60-million parameter model was bound to overfit. They needed to come up with an overfitting prevention strategy that could work at this scale.

Whenever a system has huge number of parameters, it becomes prone to overfitting.

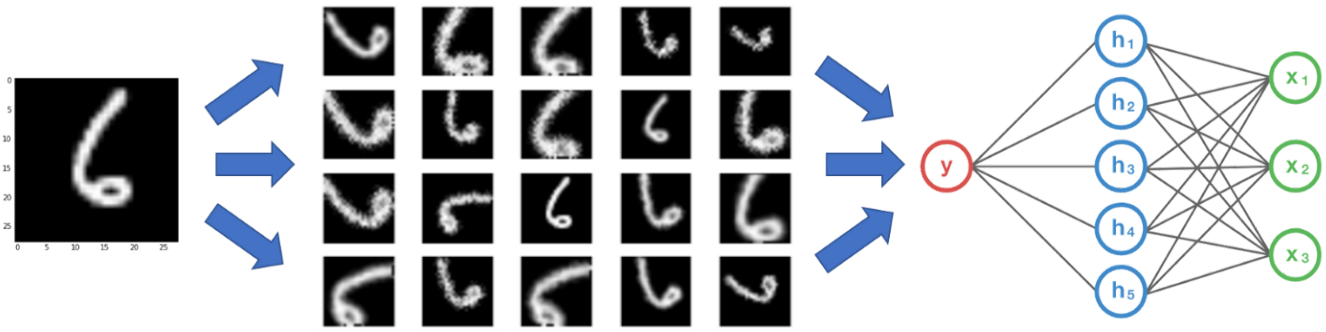
> Overfitting — Given a question that you've already seen you can answer perfectly but you'll perform

They employed two methods to battle overfitting

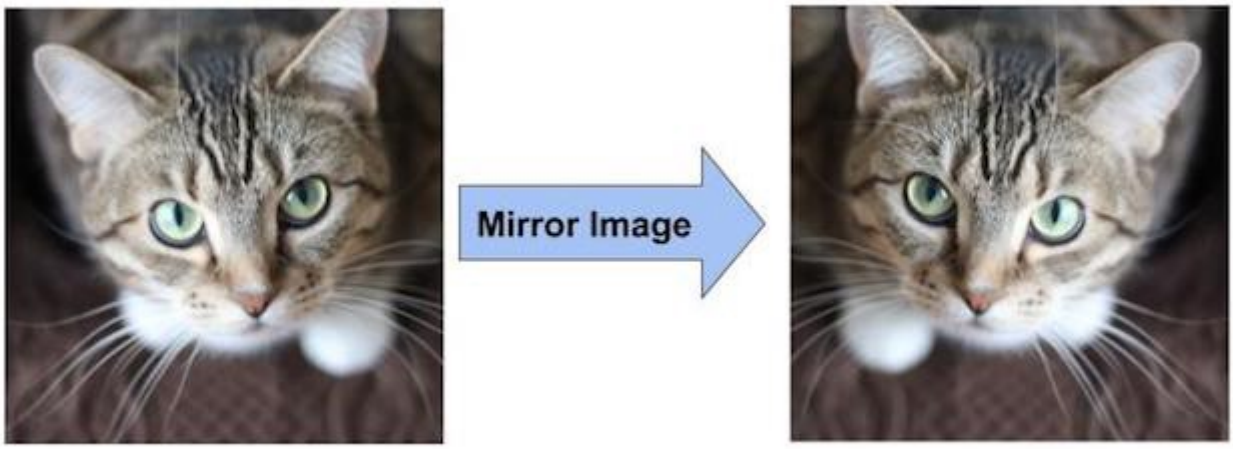
- Data Augmentation
- Dropout

Data Augmentation

Data augmentation is increasing the size of your dataset by creating transforms of each image in your dataset. These transforms can be simple scaling of size or reflection or rotation.



See how no. 6 is rotated in various directions Source



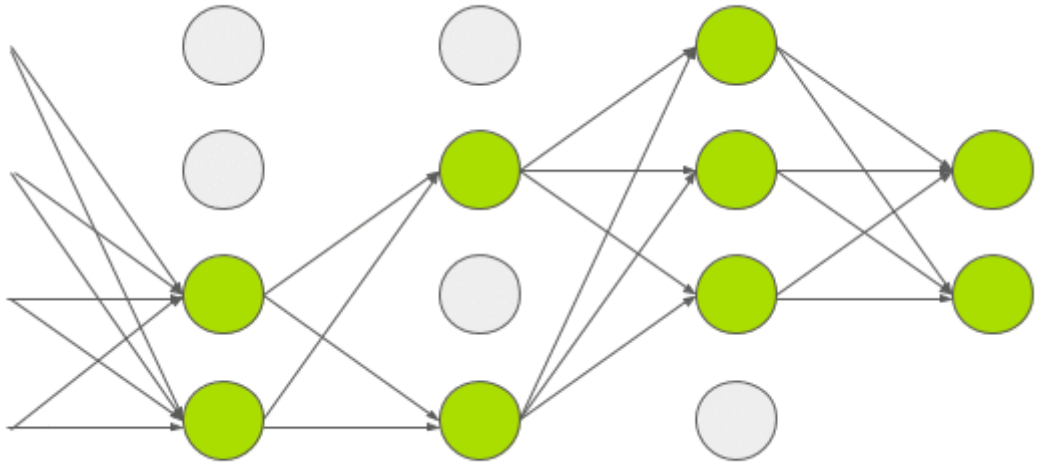
Horizontal reflection

These schemes led to an error reduction of 1% in their top-1 error metric. By augmenting the data you not only increase the dataset but the model tries to become rotation invariant, color invariant etc. and prevents overfitting

Dropout

The second technique that AlexNet used to avoid overfitting was dropout. It consists of setting to zero the output of each hidden neuron with probability 0.5. The neurons which are “dropped out” in this way do not contribute to the forward pass and do not participate in back- propagation. So every time an input is presented, the neural network samples a different architecture.

This *new-architecture-everytime* is akin to using multiple architectures without expending additional resources. The model, therefore, forced to learn more robust features.



Dropout in action. Source

. . .

The 2012 Challenge

AlexNet won the ILSVRC. This was a major breakthrough because the dataset was a formidable one and they produced highly accurate results

Predictions by AlexNet

AlexNet also released how feature extraction looked after each layer. This data is available in their supplementary material.

. . .

Prof.Hinton who won the Turing's award this year was apparently not convinced by Alex's proposed solution at first. The success of AlexNet goes on to show that with enough grit and determination, innovation does find its way to success.

. . .

Deep Dive

- Architecture source — Sik-Ho Tsang's article
- ReLU — A very nice blog by Danqing Liu

A Practical Guide to ReLU

Start using and understanding ReLU without BS or fancy equations

medium.com

- Data Augmentation

Data Augmentation | How to use Deep Learning when you have Limited Data — Part 2

This article is a comprehensive review of Data Augmentation techniques for Deep Learning, specific to images. This is...

medium.com

. . .

X8 aims to organize and build a community for AI that not only is open source but also looks at the ethical and political aspects of it. We publish an article on such simplified AI concepts every Friday. If you liked this or have some feedback or follow-up questions please comment below.

Thanks for Reading!

Machine Learning Artificial Intelligence Data Science Technology Neural Networks

About Help Legal