AlexNet: A Deep CNN for ImageNet Classification

Mahshid Alinoori

01 **Motivation**
Why AlexNet was introduced and what it contributed

02 **The Dataset**
How the data looks like

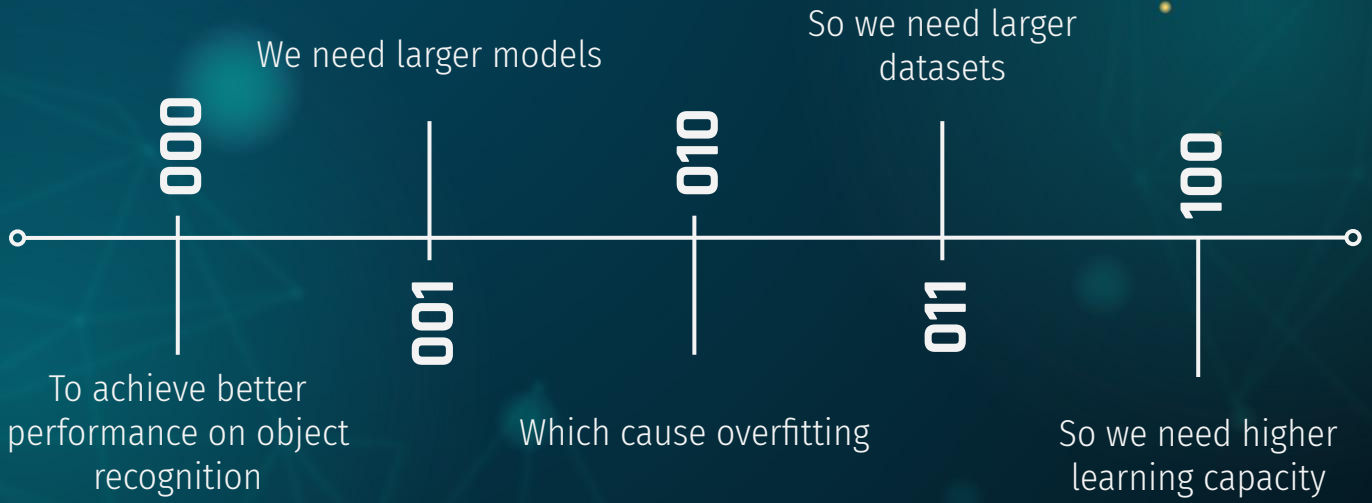
03 **The Architecture**
How the network is designed

04 **Challenges**
How to overcome overfitting

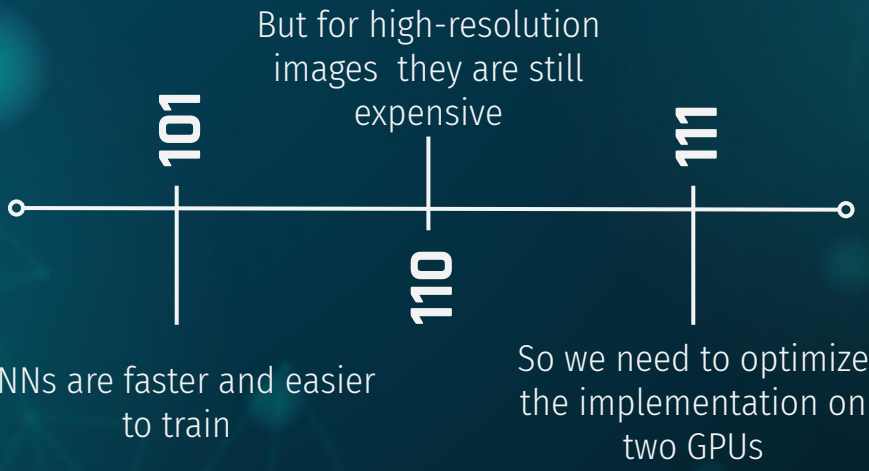
05 **Results**
How good the network is

06 **Conclusion**
Just the conclusion!

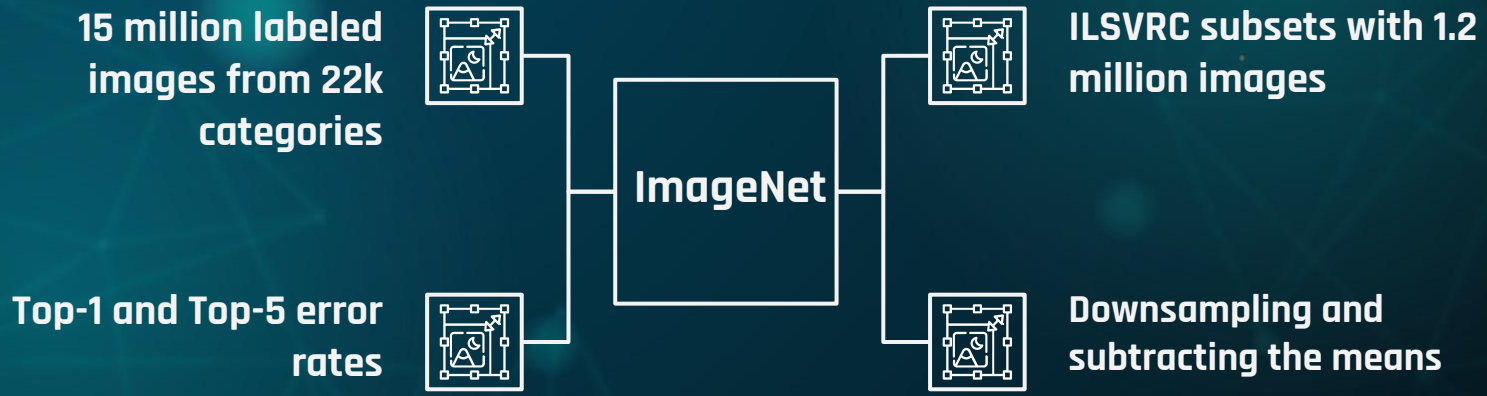
Motivation


000
To achieve better performance on object recognition
001
We need larger models
010
Which cause overfitting
011
So we need larger datasets
100
So we need higher learning capacity

Motivation


But for high-resolution images they are still expensive
101
110
111
CNNs are faster and easier to train
So we need to optimize the implementation on two GPUs

The Dataset

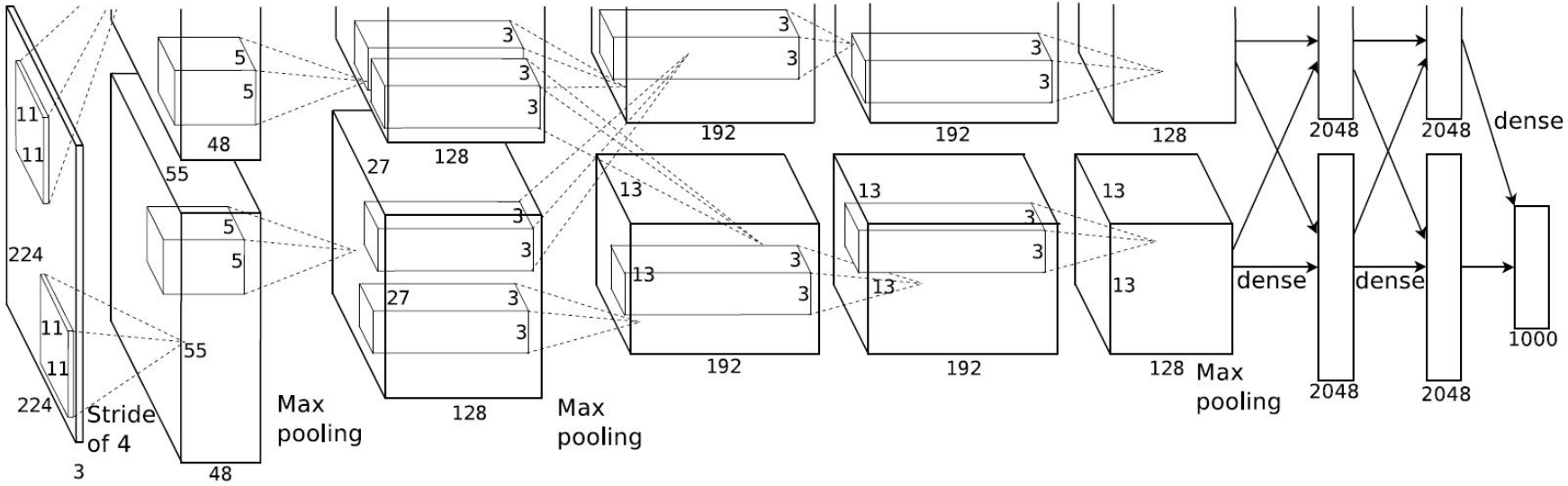

15 million labeled images from 22k categories
Top-1 and Top-5 error rates
ImageNet
ILSVRC subsets with 1.2 million images
Downsampling and subtracting the means

8 layers:
5 convolutional
3 fully connected

Input: 150,528 dimensional input (224*224*3)
Output: 1000 class labels (1000-way softmax layer)

Activation Function: ReLU as a non-saturating nonlinearity
Effect: Faster training

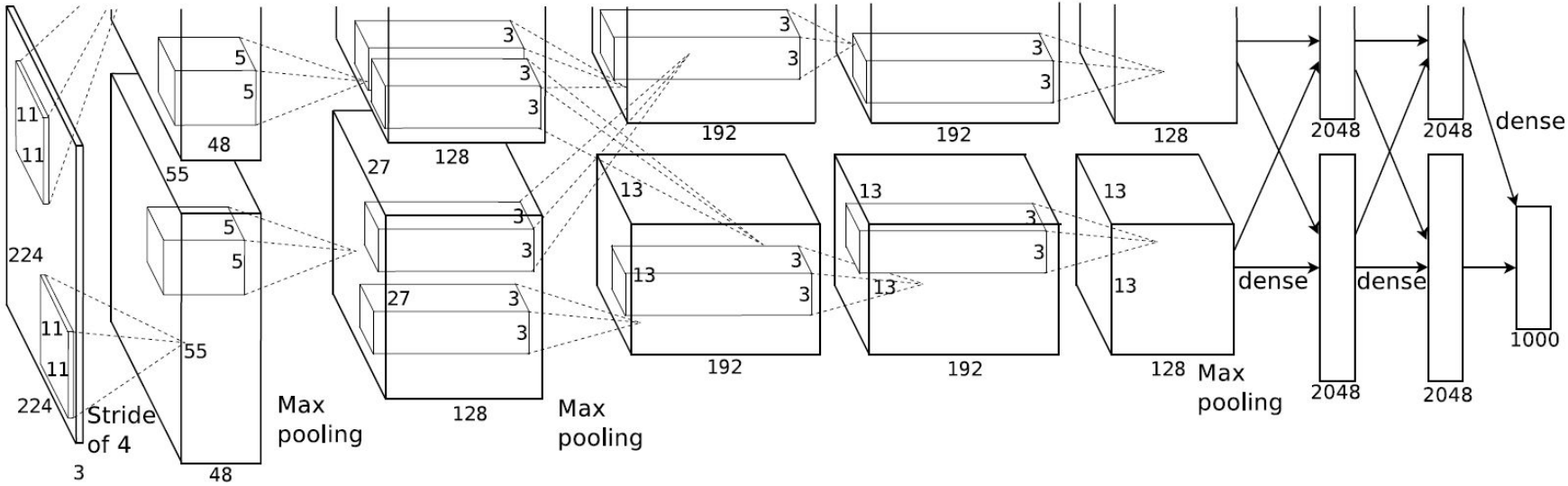
The Architecture



GPUs: 2 GTX 580 GPUs with access to each other's memory
Memory: 6 GB (2*3GB)
Communication: Only in certain layers
Effect: 1.7% and 1.2% drop in top-1 and top-5 error rates

Normalization: Local response normalization in first and second layers
Effect: Increased generalization and 1.4% and 1.2% drop in error rates

Pooling: Overlapping max-pooling following normalization and in fifth layer
Effect: Making overfitting less probable



Overfitting is still a challenge...

Data Augmentation:
1. Image translation and horizontal reflection of four corner patches and the center patch
2. Alteration in RGB intensities using PCA

Effect:
1. Increase the size of training set
2. 1% Drop in top-1 error

Dropout: Picking hidden neurons with probability of 0.5 and setting their weight to 0

Note: Only in training

Effect:
1. Making neurons to learn more robust features and prevents overfitting
2. More iteration required until convergence

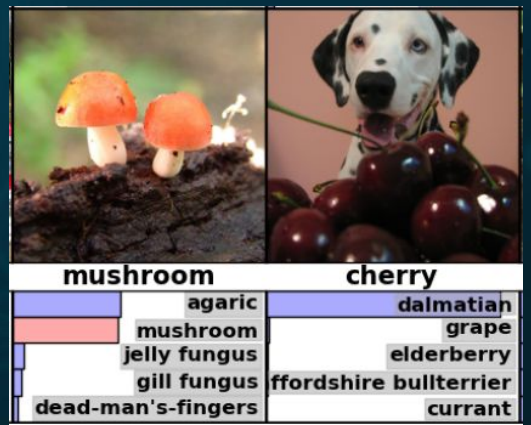
Results

Color-specific vs color-agnostic features
Frequency and orientation

Kernels

Reasonable mistakes

Top-5 predictions



Small Euclidean distance results in similar values in higher layers

Similarity

Results

37.5% top-1 and 17% top-5 with about 8% improvement

ILSVRC-2010

Further steps required to achieve a better performance: extra sixth convolutional layer, fine tuning on ILSVRC-2012

ILSVRC-2012

Conclusion

Large deep CNN on a large dataset

Techniques for faster training and avoiding overfitting

Detail-oriented design

Promising results

Thanks!