Slide 2-3

2010, the "NEC-UIUC" team (a collaboration between NEC Laboratories America and the University of Illinois Urbana-Champaign) won. [before the deep learning revolution]

1. Hand-crafted Features: Extracted SIFT, HOG, and LBP features from images.

2. Smart Encoding: Used Linear Coordinate Coding (LCC) and Spatial Pyramids to organize these features into a single vector.

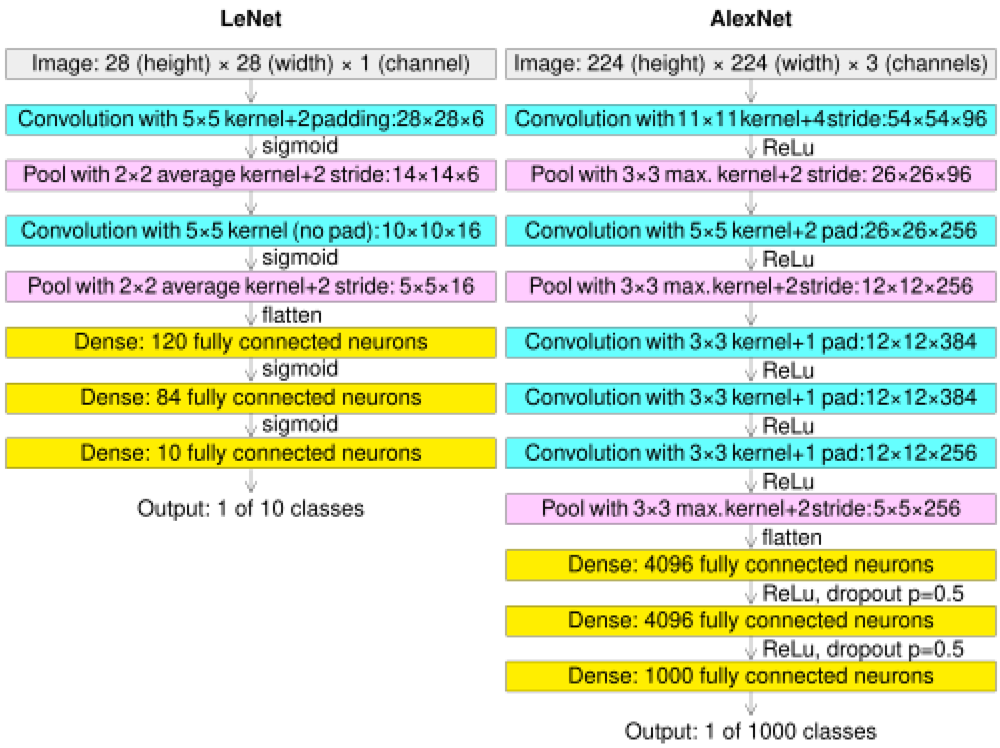
3. Classifier: Fed this vector into a Support Vector Machine (SVM) to make the final classification.

Με 2 λέξεις: feature engineering

AlexNet - CNN

CNNs attracted attention after they won the ImageNet Challenge from 2012–2017, which is a largescale image recognition contest for classifying 50,000 high-resolution color images into 1,000 categories after training 1.2 million images, held every year since 2010.

<u>Bonus</u>



In 1980, Kunihiko Fukushima proposed an early CNN named neocognitron. It was trained by an unsupervised learning algorithm. The LeNet-5 (Yann LeCun et al., 1989) was trained by supervised learning with backpropagation algorithm, with an architecture that is essentially the same as AlexNet on a small scale.

AlexNet's success in 2012 was enabled by the convergence of three developments that had matured over the previous decade: large-scale labeled datasets, general-purpose GPU computing, and improved training methods for deep neural networks.

Slide 4

The ReLU non-linearity is applied to the output of every convolutional and fully-connected layer.

Bonus:

Training on Multiple GPUs…
A single GTX 580 GPU has only 3GB of memory, which limits the maximum size of the networks that can be trained on it. It turns out that 1.2 million training examples are enough to train networks which are too big to fit on one GPU. Therefore we spread the net across two GPUs.

Slide 5

Ό,τι λέει η διαφάνεια.

Bonus:

ReLUs have the desirable property that they do not require input normalization to prevent them from saturating. If at least some training examples produce a positive input to a ReLU, learning will happen in that neuron. However, we still find that the following local normalization scheme aids generalization.

Slide 6

This is what we use throughout our network, with s = 2 and z = 3. This scheme reduces the top-1 and top-5 error rates by 0.4% and 0.3%, respectively, as compared with the non-overlapping scheme s = 2, z = 2, which produces output of equivalent dimensions. We generally observe during training that models with overlapping pooling find it slightly more difficult to overfit.

Slide 7 - Τίποτα

Slide 8

The easiest and most common method to reduce overfitting on image data is to artificially enlarge the dataset using label-preserving transformations (e.g., [25, 4, 5]). We employ two distinct forms of data augmentation, both of which allow transformed images to be produced from the original images with very little computation, so the transformed images do not need to be stored on disk. In our implementation, the transformed images are generated in Python code on the CPU while the GPU is training on the previous batch of images. So these data augmentation schemes are, in effect, computationally free - We do this by extracting random 224 × 224 patches (and their horizontal reflections).

Slide 9

This technique reduces complex co-adaptations of neurons, since a neuron cannot rely on the presence of particular other neurons. It is, therefore, forced to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.

Slide 10 - Visualizing and Understanding Convolutional Networks (2014 paper)

Visualization: Visualizing features to gain intuition about the network is common practice, but mostly limited to the 1st layer where projections to pixel space are possible. In higher layers alternate methods must be used.

Slide 11

Visualization with a Deconvnet Understanding the operation of a convnet requires interpreting the feature activity in intermediate layers. We present a novel way to map these activities back to the input pixel space, showing what input pattern originally caused a given activation in the feature maps. We perform this mapping with a Deconvolutional Network (deconvnet) Zeiler et al. [29]. A deconvnet can be thought of as a convnet model that uses the same components (filtering, pooling) but in reverse, so instead of mapping pixels to features does the opposite.