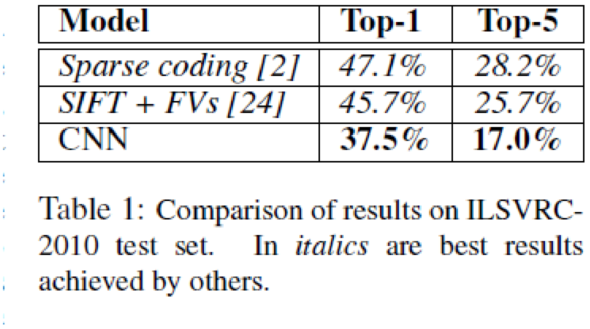
ImageNet Classification with Deep Convolutional Neural Networks

s(Paper review)

The intended goal of the experiments was to create a deep, convolutional network that uses supervised learning to achieve better (lower) error rates than the rates previously observed, to identify images, on a highly challenging dataset.
The parameters used for judging if the CNN is able to recognise the object is given by “Top-1” and “Top-5” predictions made – that is the top prediction made, and the the top 5 predictions made, and matching which ones are correct. The training set used was ILSVRC-2010 images, and ILSVRC-2012 images as training sets, and ILSVRC-2010 for the test set .which containd 15 million high resolution images in about 22,000 categories. Given a rectangular image, they first rescaled the image such that the shorter side was of length 256, and then cropped out the central 256x256 patch from the resulting image.
The main feature of the architecture of the CNN is the different layers – there are 5 convolutional layers, and 3 fully connected layers, and this feature is the main feature which helps to reduce the error rates, as the removal of even a single layer degrades the performance by about 2%.



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

Table 1: Comparison of results on ILSVRC-2010 test set. In *italics* are best results achieved by others.

This figure shows the test results for the previously used models, and the new CNN model. The network’s input is 150,528-dimensional, and the number of neurons in the network’s remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.
The main features of their model included the usage of

- ReLU nonlinearity (Rectified Linear Units) instead of the usual saturating nonlinearities (with hyprebolic functions)which resulted in accelerated ability to fit the training set and increasing the learning rate.
- Training on multiple GPUs, one of which dealt with images in a color-agnostic manner, and the other in a color-specific manner. This increased the memory that they had to increase the size of the training set. Half the kernels are on one training set, and the other half on the other, with the GPUs communicating only in certain layers. This allows them to precisely tune the amount of communication until it is an acceptable fraction of the amount of computation. For example, the kernels of layer 3 take input from all kernel maps in layer 2. However, kernels in layer 4 take input only from those kernel maps in layer 3 which reside on the same GPU. Two-GPU net also takes less time to train than one-GPU net.
- Other than this, the local response normalization aids generalization even though ReLUs don't require it. This sort of response normalization implements a form of lateral inhibition inspired by the type found in real neurons, creating competition for big activities amongst neuron outputs computed using different kernels.
- Also, overlapping pooling by pooling layers in CNNs summarize the outputs of neighboring groups of neurons in the same kernel map. (a pooling layer can be thought of as consisting of a grid of pooling units spaced s pixels apart, each summarizing a neighborhood of size z z centered at the location of the pooling unit) During training that models with overlapping pooling find it slightly more difficult to overfit.
- The architecture is such that the kernels of the second, fourth, and fifth convolutional layers are connected only to those kernel maps in the previous layer which reside on the same GPU. The kernels of the third convolutional layer are connected to all kernel maps in the second layer. The neurons in the fully-connected layers are connected to all neurons in the previous layer. Response-

normalization layers follow the first and second convolutional layers. Max-pooling layers, follow both response-normalization layers as well as the fifth convolutional layer. The ReLU non-linearity is applied to the output of every convolutional and fully-connected layer.
Overfitting is reduced by augmenting data to artificially enlarge the dataset using label-preserving transformations, either by generating image translations, and horizontal reflections, or by performing PCA on the set of RGB pixel values on all images.. Another method that reduces overfitting is dropout which 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 backpropagation. So every time an input is presented, the neural network samples a different architecture, but all these architectures share weights. As for the learning rate, they used an equal learning rate for all layers, and divided the learning rate by 10 when the validation error rate stopped improving with the current learning rate. These ideas have all contributed to the significant reduction in error rates for top-1 and top-5 predictions.

The network is completely dependent on all the layers, so the depth is important for the performance, which may be improved upon in the coming versions. No unsupervised pre-training was used, which could be implemented to further decrease the value of error rates. Also, it is dependent on computational power to compute the labels for the large datasets, and it has been achieved by making network larger and training it longer.
Ultimately, very large and deep convolutional nets on video sequences can be used, where the temporal structure provides very helpful information that is missing or far less obvious in static images.