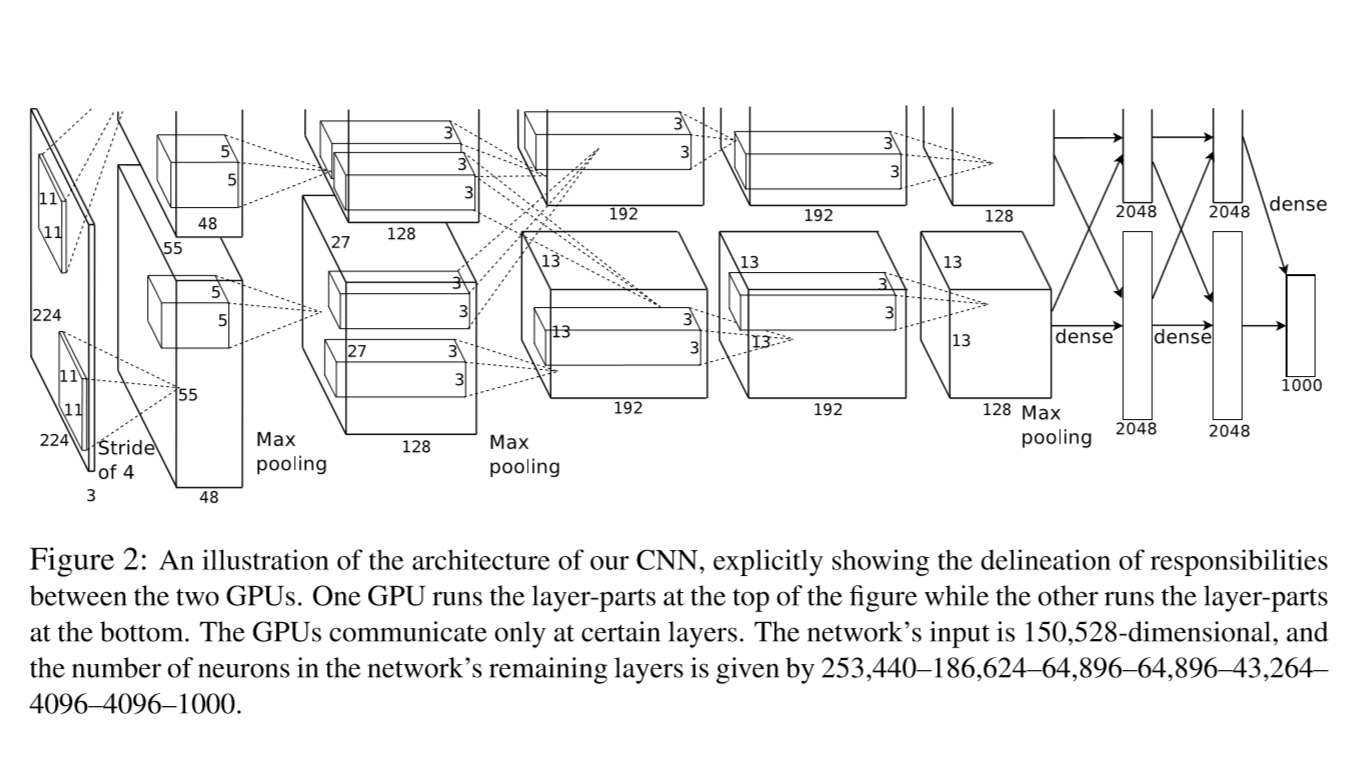
**ALEXNET PAPER POINTS**

* The neural network, which has 60 million parameters and 650,000 neurons, consists of ﬁve convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a ﬁnal 1000-way softmax  
   We found that removing any convolutional layer(each of which contains no more than 1% of the model's parameters)resulted in inferior performance.
* ImageNet [6] consists of over 15 million labeled high-resolution images in over 22,000 categories.
* CNN`s capacity can be controlled by varying their depth and breadth,and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies)
* On ImageNet,it is customary to report two error rates: top-1 and top-5, where the top-5 error rate is the fraction of test images for which the correct label is not among the ﬁve labels considered most probable by the model.
* Networks with ReLUs consistently learn several times faster than equivalents with saturating neurons. f(x) = tanh(x) or f(x) = (1 + e−x)−1 in terms of training time with gradient descent, these saturating nonlinearities are much slower than the non-saturating nonlinearity f(x) = max(0,x).
* we spread the net across two GPUs. Current GPUs are particularly well-suited to cross-GPU parallelization, as they are able to read from and write to one another’s memory directly, without going through host machine memory.
* . The parallelization scheme that we employ essentially puts half of the kernels (or neurons) on each GPU, with one additional trick: the GPUs communicate only in certain layers.
* LOCAL RESPONSE NORMALIZATION:The ordering of the kernel maps is of course arbitrary and determined before training begins. 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. The constants k,n,α, and β are hyper-parameters whose values are determined using a validation set; we used k = 2, n = 5, α = 10−4, and β = 0.75. We applied this normalization after applying the ReLU nonlinearity in certain layers
* in CNNs If we set s < z, we obtain overlapping pooling. 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.
* 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.



* It turns out to be insufﬁcient to learn 60 million parameters without considerable overﬁtting
* DATA AUGMENTATION: 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.
* Without this scheme,our network suffers from substantial overﬁtting,which would have forced us to use much smaller networks
* The ﬁrst form of data augmentation consists of generating image translations and horizontal reﬂections
* The second form of data augmentation consists of altering the intensities of the RGB channels in training images
* This scheme approximately captures an important property of natural images, namely,that object identity is invariant to changes in the intensity and color of the illumination
* We use dropout in the ﬁrst two fully-connected layers.Without dropout,our network exhibits substantial overﬁtting. Dropout roughly doubles the Number Of Iterations Required to converge
* We trained our models using stochastic gradient descent with a batch size of 128 examples, momentum of 0.9, and weight decay of 0.0005. We found that this small amount of weight decay was important for the model to learn. In other words,weight decay here is not merely regularizer: it reduces the model’s training error
* Computing similarity by using Euclidean distance between two 4096-dimensional, real-valued vectors is inefﬁcient,but it could be made efﬁcient by training an auto-encoder to compress these vectors to short binary codes. This should produce a much better image retrieval method than applying autoencoders to the raw pixels [14], which does not make use of image labels and hence has a tendency to retrieve images with similar patterns of edges, whether or not they are semantically similar
* We would like to use very large and deep convolutional nets on video sequences where the temporal structure provides very helpful information that is missing or far less obvious in static images.