**VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION**

*Karen Simonyan & Andrew Zisserman*

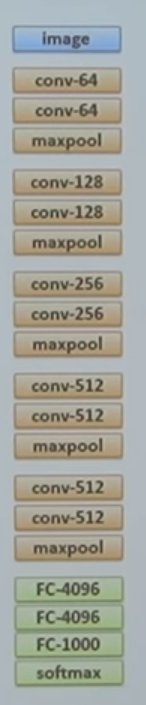
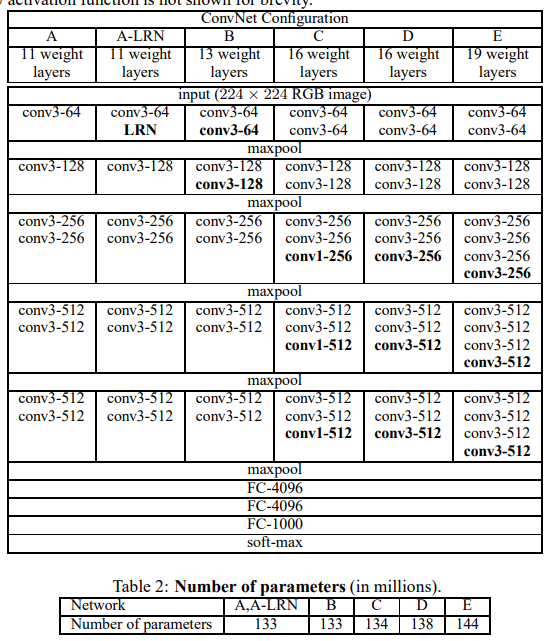
* The basic idea of VGG is to use small filters of same size (3\*3) in all conv. layers, but to incorporate more such layers in the model.
* Input image size: 224\*224\*3
* Preprocessing: subtracting mean RGB value, computed over training set, from each pixel
* Filter size in all conv. layers: 3\*3 with a stride of 1
* Padding : all conv. layers use padding such that the output shape is same as the input shape; i.e. for filters of 3\*3 shape, padding becomes 1
* Maxpooling: 2\*2 with a stride of 2
* FC layers: 3 layers at the end; two containing 4096 neurons each and the last one containing 1000 neurons, whose output is then fed to softmax activation
* All layers use ReLU activation
* This paper shows Local Response Normalization from AlexNet is not useful.
* No. of filters in the first conv. layer is 64. It is then doubled in each layer until 512.
* A stack of multiple small receptive field conv. layers (without pooling in between) is same as a single conv. layer with a large receptive field.

One advantage of having multiple smaller receptive field conv. layers is that the input goes through multiple non-linearity functions (i.e. activation functions).

Another advantage is that the no. of parameters in the model decreases.

* VGG model outperforms AlexNet
* Various models that authors experimented with are given below:

(The image on the right side is taken from a youtube video)



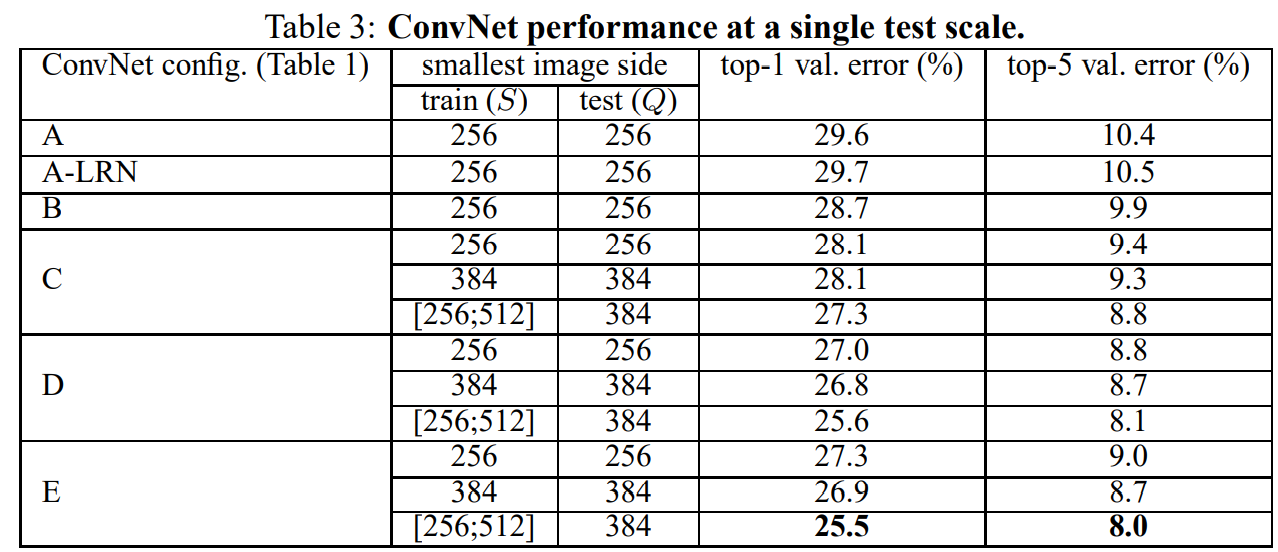
* The training procedure is similar to that of AlexNet, except that the AlexNet uses random crops from input images for training whereas VGG uses one random crop per input image for training. Random crop is taken per iteration, so crops in different epochs will be different for same image.
* Multinomial logistic regression cost function
* Mini-batch gradient descent with momentum.
  + Batch size: 256
  + momentum: 0.9
  + L2 regularization with regularization rate 5\*10-4
  + Dropout for the first two FC layers with rate 0.5
  + Initial learning rate: 0.01, which was reduced to learning rate/10 whenever validation accuracy stopped improving.
  + 74 epochs
* Initialization:
  + Random initialization here means normal distribution with mean 0 and std 0.01
  + Started with 11-layer CNN with random initialization. Trained it, and used the learned weights to initialize larger CNN network.
  + Since in each larger model, more conv. layers were added, only initial few conv. layers in the new model could utilize the weights learned from the previous smaller model. For remaining conv. layers, random initialization was used.
  + Since FC layers remained same for all models, the weights learnt in 11-layer model were assigned to the FC layers in the larger model.
  + Biases were set to 0
  + For training larger models, smaller learning rate was not used even though weights in few layers were already tuned.
  + **Paper mentions that Glorot and Bengio initialization can also be used directly rather than using pre-trained weights from smaller models.**
* Data Augmentation:
  + Only one random crop of size 224\*224 per input image was used for training. In each epoch, new random crop was used.
  + The crops went through random horizontal flipping and random RGB colour shift, as explained in AlexNet.
* Tried various approaches for input image size
  + In one trial, input image size of 256\*256
  + In another trial, input image size of 384\*384 was used. During this, the weights learned for 256\*256 images were used for initialization and small learning rate 0.001 was used.
  + In third approach, each input image was rescaled to random size in the range [256, 512]

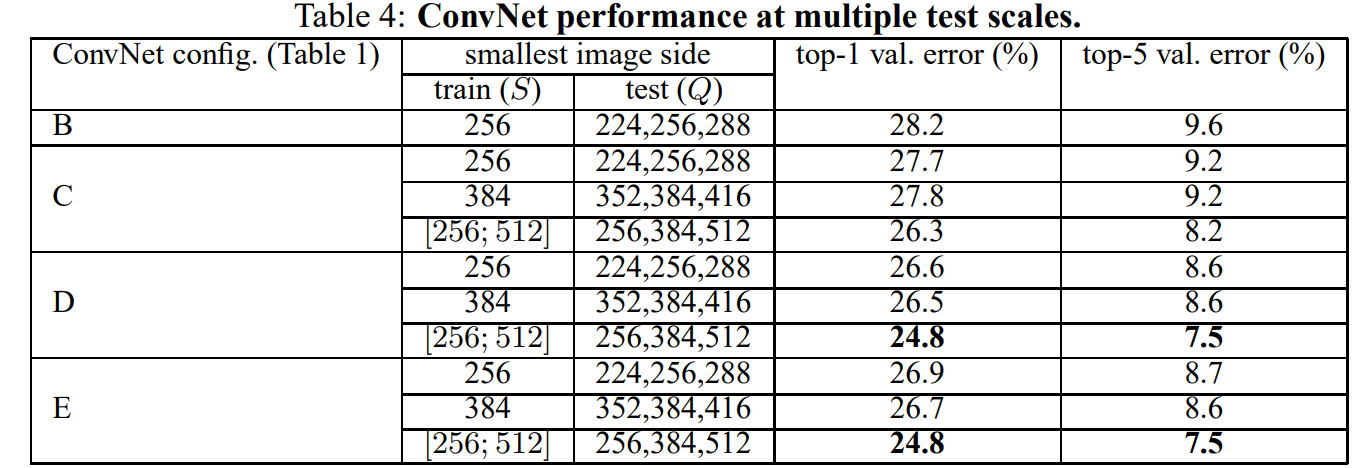
So, the model could receive an image of any size in this range.

This was helpful because, in real world, objects in images could be of any size.

This third model was initialized by weights which were learnt over the images of the fixed shape 384

* Testing:
  + Rescale the image to its lowest side dimension
  + Convert the FC layers in the model to Conv. layers (same as the convolutional implementation of window approach shown in DL specialization course 4)
  + So, the output would be a grid, where each output entry will correspond to each crop
  + Average the output entries to get the final prediction
* Results:
  + ILSVRC 2012
  + 1000 classes, training size: 1.3 million, validation: 50k, and testing: 100k
  + A deep net with small filters outperforms a shallow net with larger filters
  + Training on images of various resolution improves accuracy of the model





* Using ensemble of various VGG models improves accuracy further. The best error rate (7%) was achieved by an ensemble having models D and E.
* SKIPPED localization part of the paper.