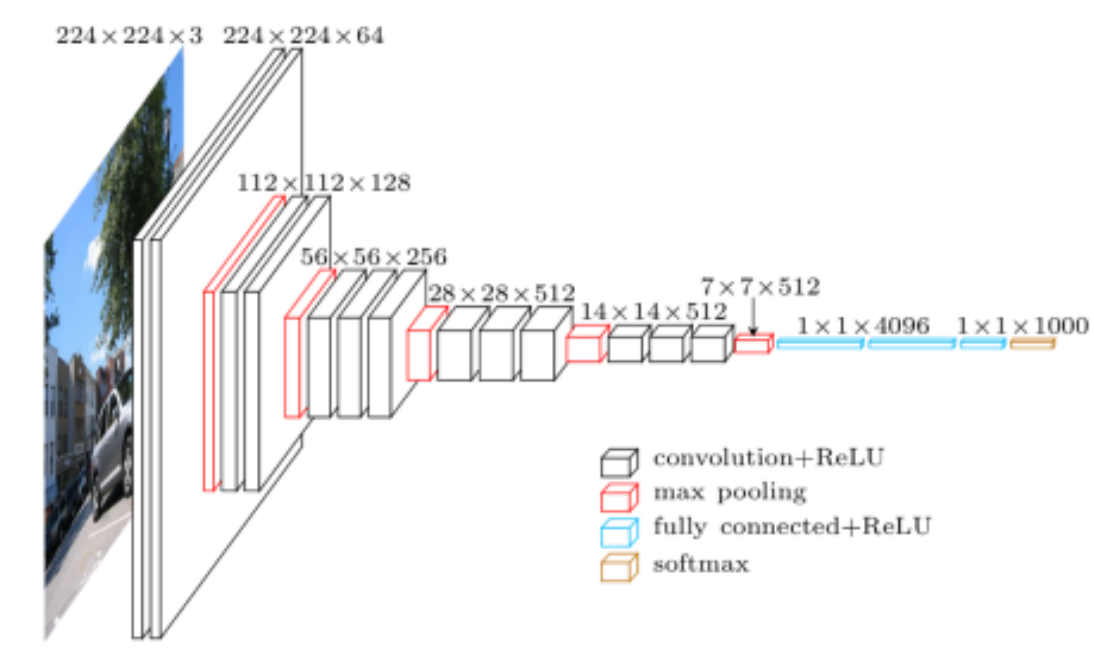
VERY DEEP CONVOLUTIONAL NETWORKS VGG

AISHWARYA NAGVEKAR,SANJANA K,SILPA BALAGOPAL
@Computational Engineering and Networking, Amrita School of Engineering

The architecture we are using is VGG-19. Visual Geometry Group developed and trained this algorithm at Oxford University which gave good performance for ImageNet Dataset. It is a very deep CNN network which consist of 19 convolution layers.
Initially a VGG network with less layers was formulated among which VGG 16 and VGG 19 is found to be most effective for ImageNet dataset . They are considered to be deep neural network as it consist of many layers and eventually takes a lot of time for training. Therefore method of using pre trained network is found to be most effective. An architecture consisting minimum layer is trained first and later addition of layers is done. This process is repeated till the required amount of layers are added to get a good accuracy. This makes the training process more efficient.

ARCHITECTURE
VGG-16



Architecture of VGG-16

The main difference between VGG 16 and VGG 19 is the number of convolution layers. In VGG 16 it is 16 convolution layer where as in VGG 19 it is 19 convolution layers. The number of stacking in third, fourth and fifth layer has increased from 3 in VGG-16 to 4 in VGG -19 making it to total 19 layers. Thus increasing the number of learnable parameters which in turn increases the accuracy.

VGG-19

The default size of 224 X 224 X 3 is kept as the size of input image. The convolution filter of size 3X3 .

In the first stage, there are 2 convolution filters each of size 3 X 3 as such 64 filters ,stride is 1,and zero padding is 1. The two convolution layer are stacked together as to increase the receptive field.The output of the convolution layer will be 224 X 224 X 64. The output of these filters is max pooled with stride 2 and size is reduced to 112 X 112 with depth 64. So the size decreased to half the size of the output image.

Second layer has 128, 3 X 3 convolution filters with zero padding as 1. Like these 2 convolution layers are stacked. The output of the convolution layer will be 112 X112X128. Again max pooling layer is done with stride 2 to reduce the size to 56 X 56 with depth as 128.

Third layer has a size of 3 X 3 convolution filters as such 256 filters with stride 1 and zero padding as 1. Like these 4 convolution layer are stacked together. The output of the convolution layer will be 56X56X256. Next is the max pooling layer with stride 2 which decreases the size to 28 X 28 with depth as 256.

Forth layer consist of convolution filter of size 3 X 3 as such 512 filter with stride as 1 and zero padding as 1. Similar to previous layer 4 such layers are stacked together. The output of the convolution layer will be 28 X28X512. Next is a max pooling layer which decreases the size to 14 X 14 with depth 512.

Fifth stage has convolution filter of size 3 X 3 as such 512 filter with stride 1 and zero padding as 1. Here also 4 such layers are stacked together.The output of the convolution layer will be 14 X14X512.Next max pooling is done and size of 7 X 7 is obtained with depth 512.

Every stage has 3 X 3 filter with 3 filters for RGB but the only difference is in each layer has different filter depth. An activation function used in all the layers is Relu. Finally there are 2 fully connected layers which of size 4096 which allows node to node connections. So to apply this we need to flatten the output of the previous 7 X 7 X 512 matrix. The output of the 2 fully connected layer is given to another fully connected layer of size 1000 and in the end to the softmax function which predicts the probabilities. Here the output is of 1000 classes because its trained to classify 1000 classes of imagenet dataset.

To sum it up, there are 16 weighted layers and 3 fully connected layers, total of 19 layers that's why the name VGG-19. There are intermediate max pooling layers which reduces the size of the immediate output In the end there is a prediction layer which uses softmax function which has size equal to the number of subjects.

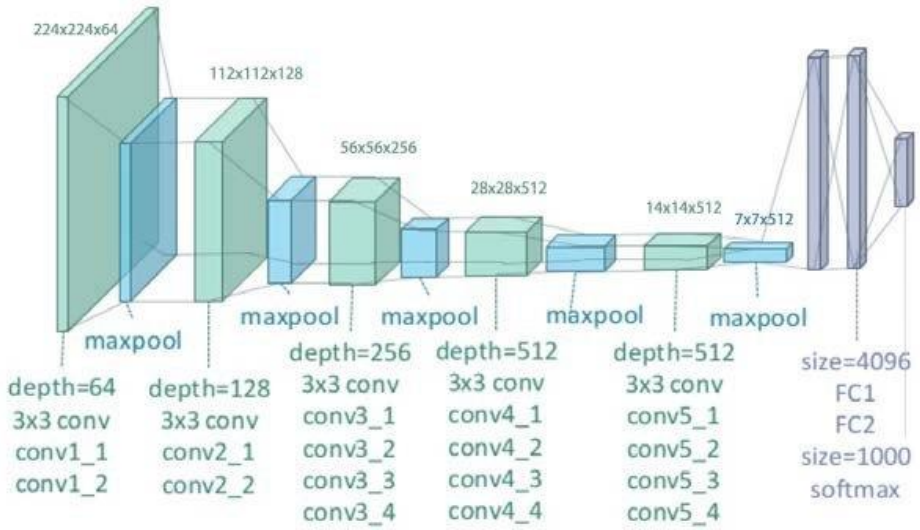


Fig: VGG-19 architecture

DETAILS OF LAYERS OF VGG 19

19 weight layers	Stride	Output Shape	Parameters
Input(224 X 224 X 3 RGB image)	0	224,224,3	0
Conv3-64+ReLU	1	224,224,64	1792
Conv3-64+ReLU	1	224,224,64	36928
Max pooling	2	112,112,64	0
Conv3-128+ReLU	1	112,112,128	73856
Conv3-128+ReLU	1	112,112,128	147584
Max pooling	2	56,56,128	0
Conv3-256+ReLU	1	56,56,256	295168
Conv3-256+ReLU	1	56.56,256	590080
Conv3-256+ReLU	1	56,56,256	590080
Conv3-256+ReLU	1	56,56,256	590080
Max pooling	2	28,28,256	0
Conv3-512+ReLU	1	28,28,512	1180160
Conv3-512+ReLU	1	28,28,512	2359808
Conv3-512+ReLU	1	28,28,512	2359808
Conv3-512+ReLU	1	28,28,512	2359808
Max pooling	2	14,14,512	0
Conv3-512+ReLU	1	14,14,512	2359808
Conv3-512+ReLU	1	14,14,512	2359808
Conv3-512+ReLU	1	14,14,512	2359808
Conv3-512+ReLU	1	14,14,512	2359808
Max pooling	2	7,7,512	0
Flatten	0	25088	0
Fully connected layer-4096+ReLU	0	4096	102764544
Fully connected layer-4096+ReLU	0	4096	16781312
Fully Connected layer-1000+ReLU	0	1000	4097000
Softmax(Prediction)		Total parameters	14,36,67,240

Total of 144 million parameters.

CALCULATION

Input image size: 224 X 224 X 3

16 Convolution filter size: 3 X 3 X 3(RGB) with stride 1 and zero padding as 1

Max Pooling: 2 X 2 window size with stride 2

2 Fully Connected Layer of size 4096

1 Fully connected layer of size 1000

Softmax layer

$$Output\ image\ size = \frac{input\ image\ size - filter\ size + 2 * zero\ padding}{stride} + 1$$

Rather than using large size convolution filter of 11 X 11 with stride 4 or 7 X 7 with stride 2, the convolution filter of size 3 X 3 with stride 1 is used. The receptive field is increased by stacking two convolution filters which has equivalent receptive field as of 5 X 5 convolution filter. To further increase the receptive field the filters are stacked one after the other.

$$Total\ learnable\ Parameters = inputs * outputs + bias$$

Where Inputs = number of filters when it is a convolution layer

Inputs = number of nodes if it's a dense layer

Outputs= number of filters * size of filters

Bias = number of filters

For Example: To calculate the learnable parameters in the first layer; it is equal to

3 X 3 X 3 X 64 + 64 = 1792

Max Pooling layer with stride 2 and 2 X 2 pixel window.

For example given matrix

4	8	7	8
2	1	3	9
3	7	2	1
6	3	6	4

Max Pooling output

8	9
7	6

The receptive field is defined as the region of input space that is effected a particular unit of the filter. The input can be given as input image or output from the previous layer. The receptive field can be increased by:

i) Stacking convolution layers (as applied above)
ii) Subsampling(pooling, striding) %how does it effective receptive field
iii) Filter dilation(dilated convolution)% find dilated convolution

Receptive field is dynamic and changes during training.

The major Drawbacks of VGG:

i) The architecture is very large with many layers
ii) It takes a lot of time for training

REFERENCES:

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014)