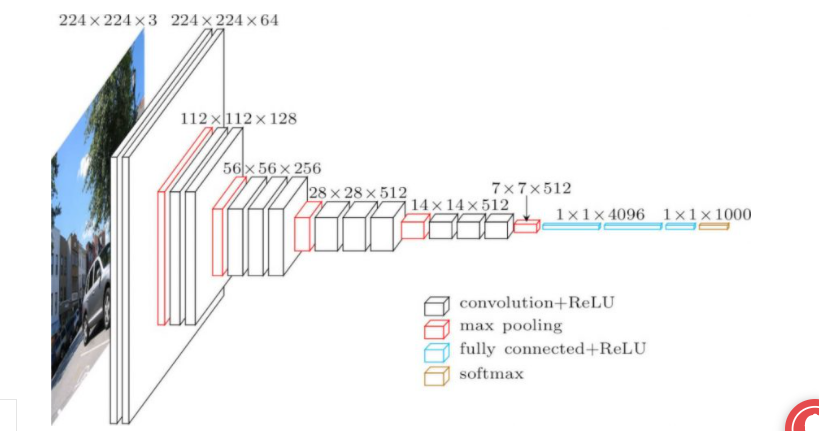
**VGG16**

**VGG16** (also called OxfordNet) is a convolutional neural network architecture named after the Visual Geometry Group. It is considered to be one of the excellent vision model architecture. It is a convolutional neural network that is 16 layers deep. You can load a pretrained version of the network **trained** on more than a million images from the ImageNet database. VGG-16 has roughly 138 million parameters and ResNet has 25.5 million parameters.



Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights.

Need to make a sequential model

Import imageDataGenerator from keras.preprocessing – as it has functions to rescale, rotate, zoom, flip…

Create object of imageDataGenerator for both training and testing

Automatically ImageDataGenerator will label all the data inside the folder

Add all the convolution layers, maxpooling layers of keras and padding

Also add relu activation to all the layers so that negative values are not passed to nesxt layers

Then pass the data to dense layers to flatten the vector that comes out of the convolutions

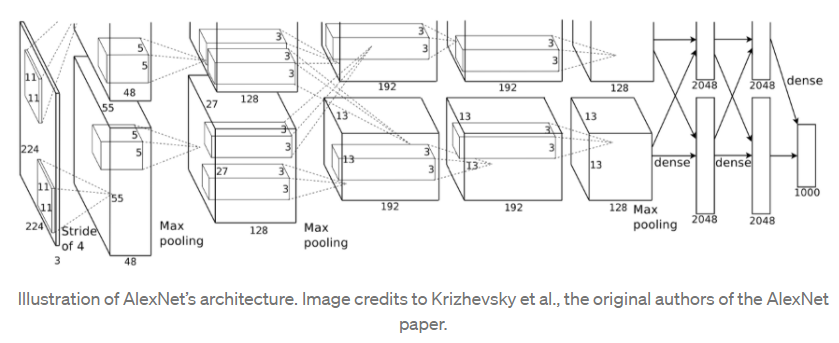
At the end use the softmax layer, then the model is finally prepared

Now compile the model

**AlexNet**

**AlexNet** – ImageNet Classification with Deep Convolutional Neural Networks. **AlexNet has** about 660K units, 61M **parameters**, and over 600M connections. Notice: the convolutional layers comprise most of the units and connections, but the fully connected layers **are** responsible for most of the weights. **VGG**-**16** which has 13 convolutional and 3 fully-connected layers, carrying with them the ReLU tradition from **AlexNet**. This network stacks more layers onto **AlexNet**, and use smaller size filters (2×2 and 3×3).

The architecture consists of eight layers: five convolutional layers and three fully-connected layers. AlexNet uses Rectified Linear Units (ReLU) instead of the tanh function, which was standard at the time, It accelerates the speed by 6 times at the same accuracy.. AlexNet allows for multi-GPU training by putting half of the model’s neurons on one GPU and the other half on another GPU.



When there’s a problem of overfitting in AlexNet, then there are two ways to solve.

They are; Data Augmentation and drop out