CONVOLUTIONAL NEURAL NETWORKS

SUBHAJIT CHATTERJEE

AD20216803

ML LAB

Contents

Introduction to Computer Vision

Computer vision is concerned with the automatic extraction, analysis and understanding of useful information from a single image or a sequence of images.

- The British Machine Vision Association and Society for Pattern Recognition (BMVA)

(or)

Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos.

- Wikipedia

What is CNN(Convolutional Neural Network) ?

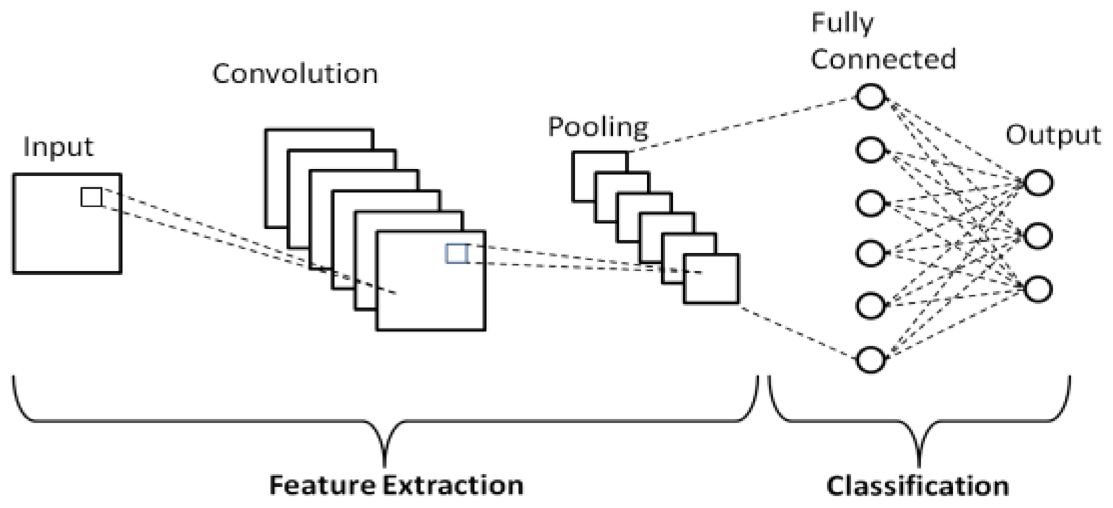
- In deep learning, a convolutional neural network (CNN, or ConvNet) is **a class of artificial neural network, most commonly applied to analyze visual imagery.**

- Convolutional neural network (ConvNet, or CNNs) is one of the main categories to do **images recognition**, **image classification**, **object detections**, **recognition faces** etc.

- It is similar to the basic neural network. CNN also have learnable parameter like neural network i.e., weights, biases etc.

- CNN is heavily used in computer vision.
- There 3 basic components to define CNN
  - The Convolution Layer.
  - The Pooling Layer.
  - The Output Layer (or) Fully Connected Layer.

Basic architecture of CNN



The CNN is a combination of two basic building blocks:

- The Convolution Block — Consists of the Convolution Layer and the Pooling Layer. This layer forms the essential component of Feature-Extraction.
- The Fully Connected Block — A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.

Convolution Layer :

Computers read images as pixels and it is expressed as matrix (N x N x 3) – (height by width by depth).

The Convolutional Layer makes use of a set of learnable filters. A filter is used to detect the presence of specific features or patterns present in the original image (input).

This filters convolved (slided) across the width and height of the input file, and a dot product is computed to give an activation map.

ReLU :

This layer will apply an element wise activation function, this leaves the size of the volume unchanged.

Pooling Layer :

This layer will perform a down sampling operation along the special dimensions (width, height).

Fully-connected Layer :

This layer compute the class scores, resulting in volume of size [1 x 1 x N], where each of the N numbers correspond to a class score, such as among the N categories.

A CNN

How to decide the number of convolution layers and number of filters in CNN ?

- Deeper networks in always better, at the cost of more data and increased complexity of learning.
- You should initially use fewer filters and gradually increase and monitor the error rate to see how it is varying.
- Very small filter sizes will capture very fine details of the image. On the other hand having a bigger filter size will leave out minute details in the image.

Types of CNN

- Based on the problems, we have the different CNN's which are used in computer vision.
- The five major computer vision techniques which can be addressed using CNN.
  - Image Classification
  - Object Detection
  - Object Tracking
  - Semantic Segmentation
  - Instance Segmentation

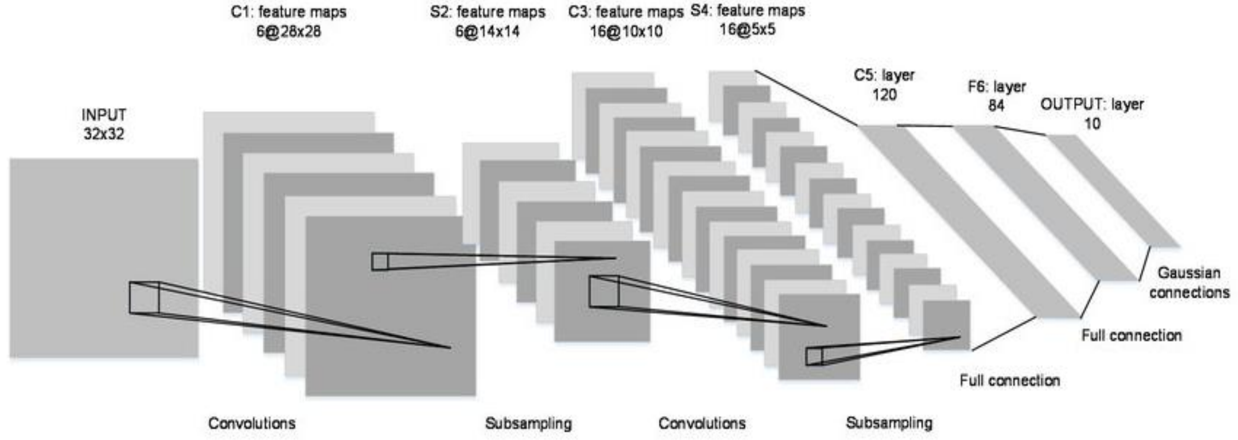
Image Classification

In an image classification we can use the traditional CNN models or there also many architectures designed by developers to decrease the error rate and increasing the trainable parameters.

- LeNet (1998)
- AlexNet (2012)
- ZFNet (2013)
- GoogLeNet (2014)
- VGGNet (2014)
- ResNet (2015)

LENET (1998)

- The LeNet-5 architecture consists of two sets of convolutional and average pooling layers, followed by a flattening convolutional layer, then two fully-connected layers and finally a softmax classifier.
- This network takes a grayscale 32 x 32 x 1 image as input.
- It goes to the convolution layers (C1) and then to the subsampling layer (S2)
- Nowadays the subsampling layer is replaced by a pooling layer.
- Then, there is another sequence of convolution layers (C3) followed by a pooling (that is, subsampling) layer (S4)
- Finally, there are two fully connected layers, including the OUTPUT layer at the end.
- This network was used for zip code recognition in post offices.



Neural network did not spark before 2010, what exactly happened??

We all consider that or thoughts that better algorithm = better results, regardless of data.

This theory is false. The best algorithm would not work well if the data learnt from a didn't reflect the real world.

Professor Fei Fei Li from Stanford University said:

"We decided we wanted to do something completely, historically unprecedented.

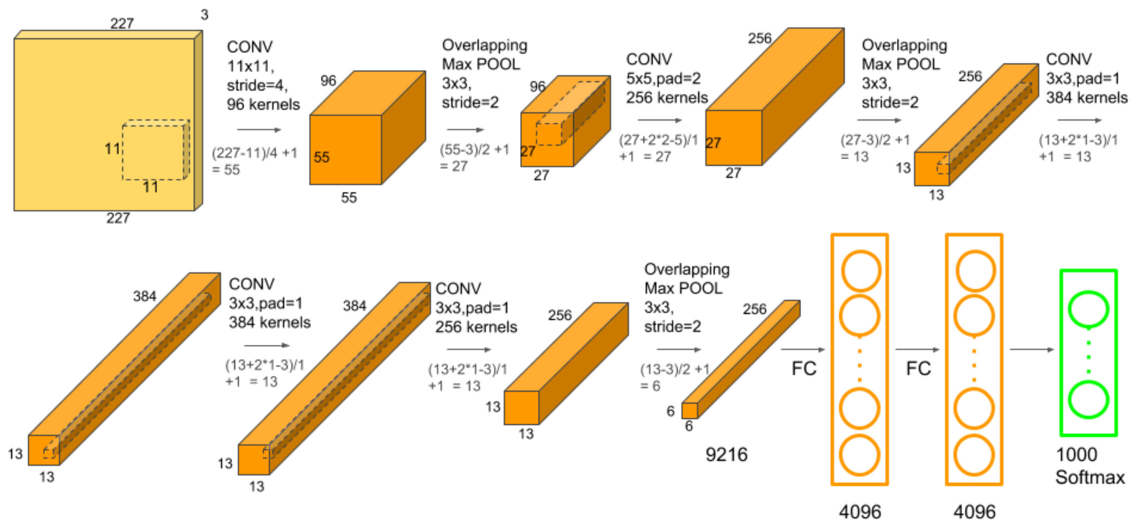
we are going to map the entire world of objects…"

The resulting dataset: ImageNet

After 2010 onwards an annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) held between researchers that which algorithms can yield the best performance.

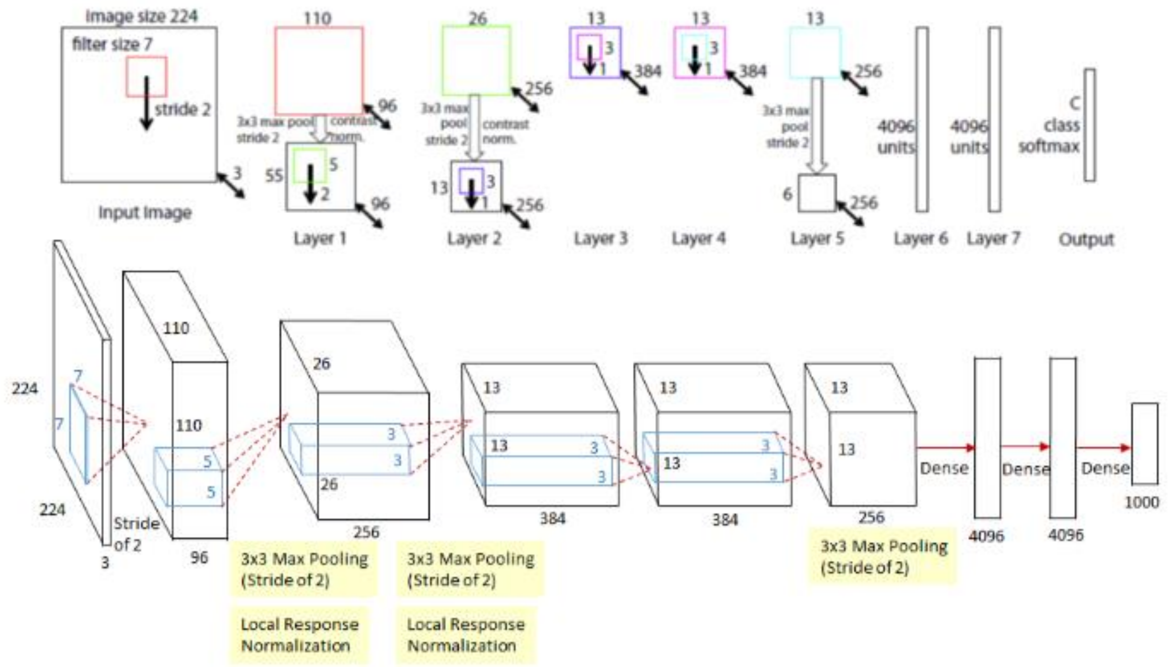
ALEXNET (2012)

- AlexNet consists of 5 Convolutional Layers and followed by 3 Fully Connected Layers.
    1. ReLU (a non saturating non linearity) as activation function.
    2. It can be trained on multiple GPUs.
    3. Batch normalization.
    4. Overlapping pooling.
- In addition, after the first, second, and fifth convolutional layers, the network adds overlapping max pooling layer with a window shape of 3×3 and a stride of 2.
- After the last convolutional layer there are two fully-connected layers with 4096 outputs each.
- Last one fully-connected layer with dense size 1000 channels.


227
3
11
11
227
CONV 11x11, stride=4, 96 kernels
(227-11)/4 +1 = 55
96
55
55
Overlapping Max POOL 3x3, stride=2
(55-3)/2 +1 = 27
96
27
27
CONV 5x5,pad=2 256 kernels
(27+2*2-5)/1 +1 = 27
256
27
27
Overlapping Max POOL 3x3, stride=2
(27-3)/2 +1 = 13
256
13
13
CONV 3x3,pad=1 384 kernels
(13+2*1-3)/1 +1 = 13
384
13
13
CONV 3x3,pad=1 384 kernels
(13+2*1-3)/1 +1 = 13
384
13
13
CONV 3x3,pad=1 256 kernels
(13+2*1-3)/1 +1 = 13
256
13
13
Overlapping Max POOL 3x3, stride=2
(13-3)/2 +1 = 6
256
6
6
9216
FC
4096
FC
4096
1000 Softmax

CLARIFAI or ZFNET(2013)

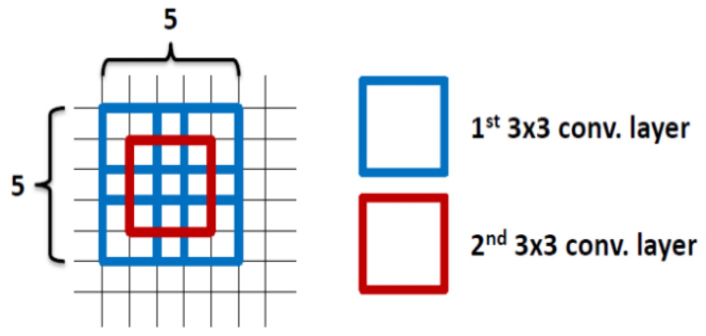
- Architecture of our 8 layer convnet model.
- A 224 x 224 crop of an image (with 3 color planes) is presented as the input.
- This is convolved with 96 different 1st layer filters (red), each of size 7 x 7 filters, using a stride of 2.
- The resulting feature maps are then: (i) passed through a ReLU activation (not shone), (ii) max pooling (max within 3 x 3 regions, using stride 2) and (iii) contrast normalized across feature maps to give 96 different 55 x 55 element feature maps.
- Similar operations are repeated in layers 2,3,4,5.
- The last two layers are fully connected(6,7), taking features from the top convolutional layer as input in vector form.
- The final layer is a C-way softmax function(8), C being the number of classes (1000).



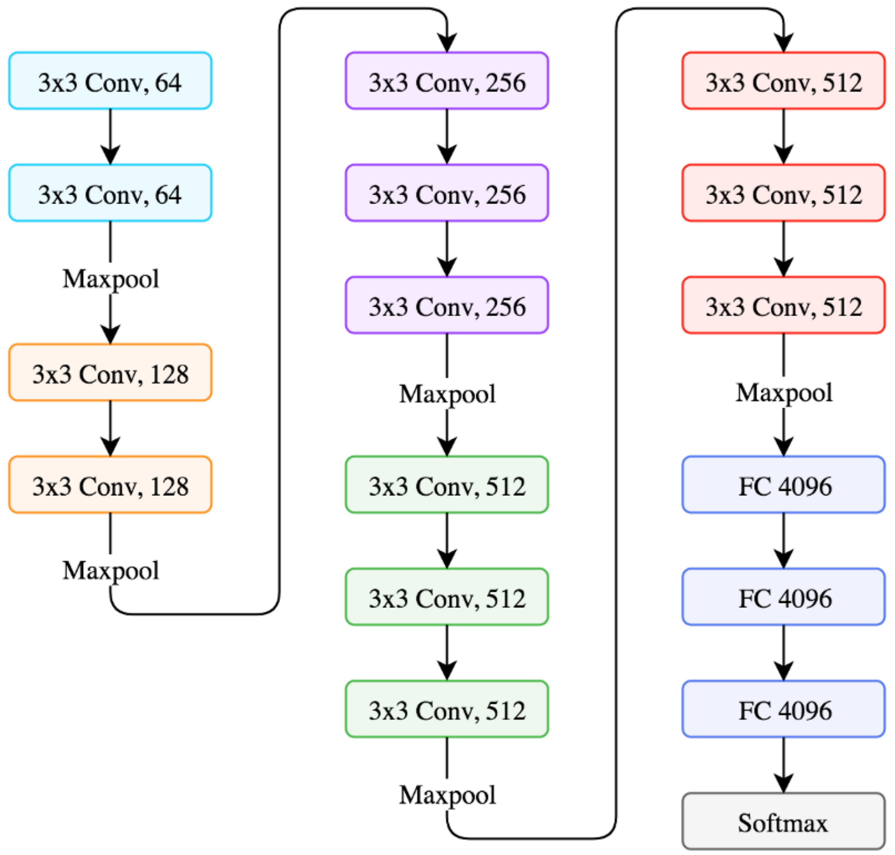
ZFNet

VGGNET (2014)

- There are total of 13 convolutional layers and 3 fully connected layers in VGG16 architecture.
- VGG Net used 3x3 filters compared to 11x11 filters in AlexNet and 7x7 in ZFNet.
- Having 2 consecutive 3x3 filters gives an effective receptive field of 5x5, and 3 – 3x3 filters give a receptive field of 7x7 filters.
- 5 max-pooling layer with stride 2.
- Three Fully Connected layers with 4096 channels each which is followed by another fully connected layer.
- Softmax output of 1000 classes.

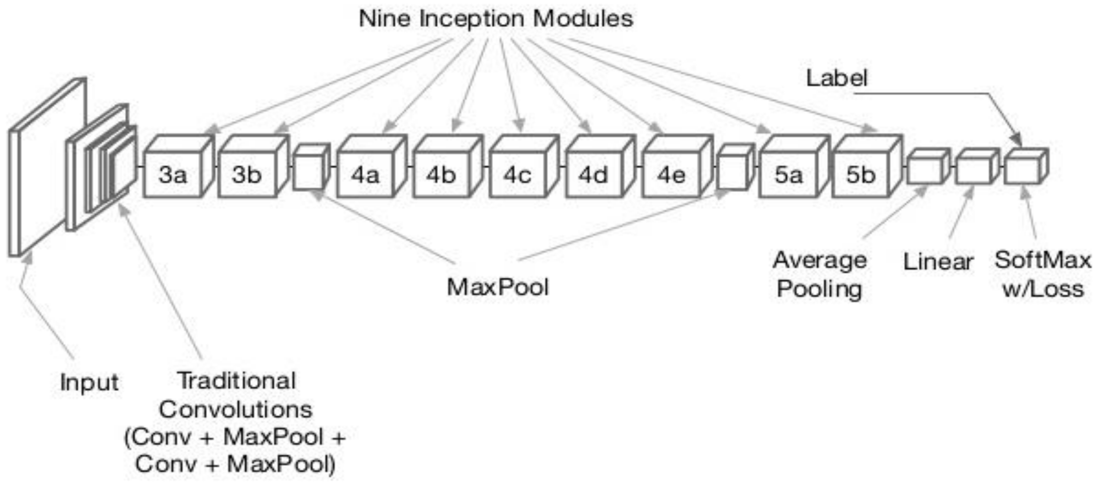


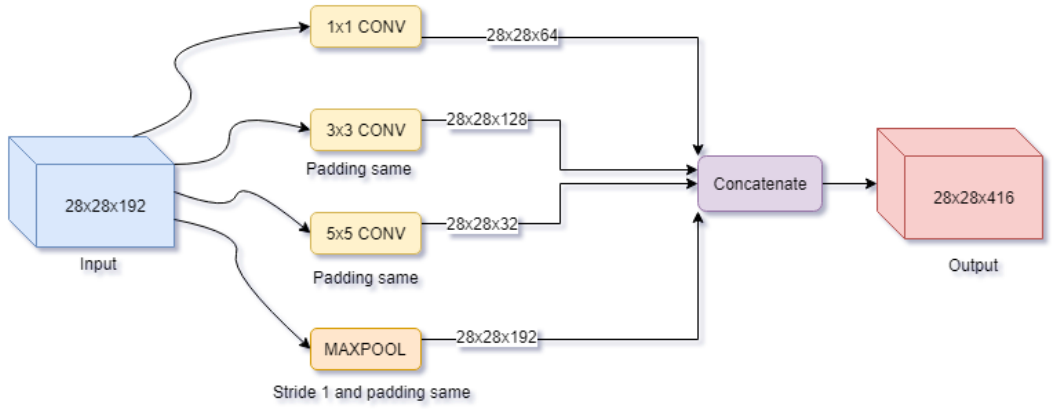
2 layers of 3×3 filters already covered the 5×5 area



GOOGLENET (2014)

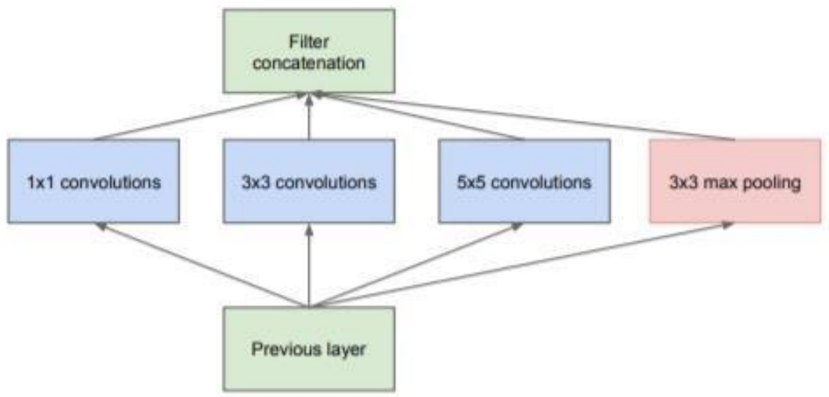
- GoogLeNet uses 9 inception module and it eliminates all fully connected layers using average pooling to go from 7x7x1024 to 1x1x1024.
- The overall architecture is 22 layers deep.
- The idea behind that the architecture can be run on individual devices even with low computational resources.
- The architecture also contains two auxiliary classifier layer connected to the output of Inception (4a) and Inception (4d) layers.
- **The architectural details of auxiliary classifiers as follows**:
- An average pooling layer of filter size 5×5 and stride 3.
- A 1×1 convolution with 128 filters for dimension reduction and ReLU activation.
- A fully connected layer with 1025 outputs and ReLU activation.
- Dropout Regularization with dropout ratio = 0.7.
- A softmax classifier with 1000 classes output similar to the main softmax classifier.





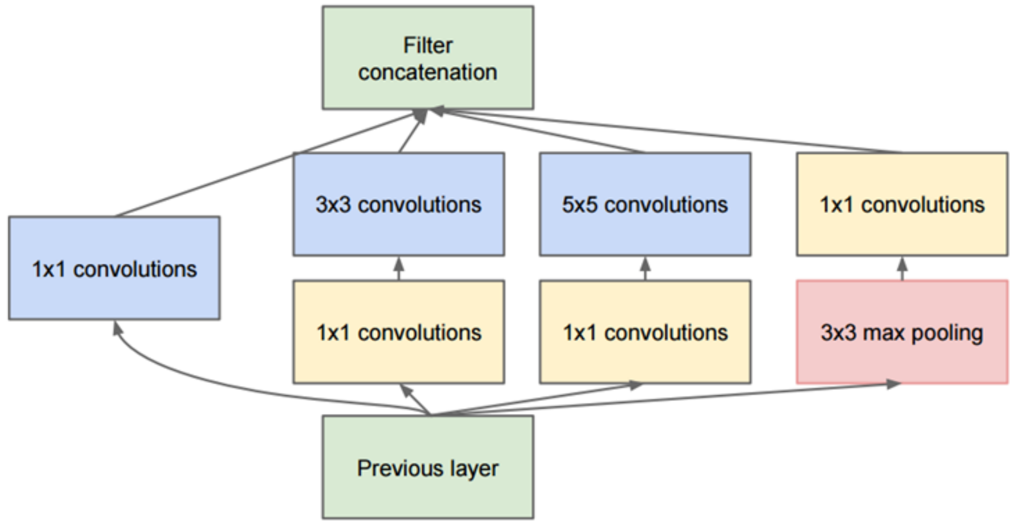
Inception Module

GOOGLENET (2014)



Old Way

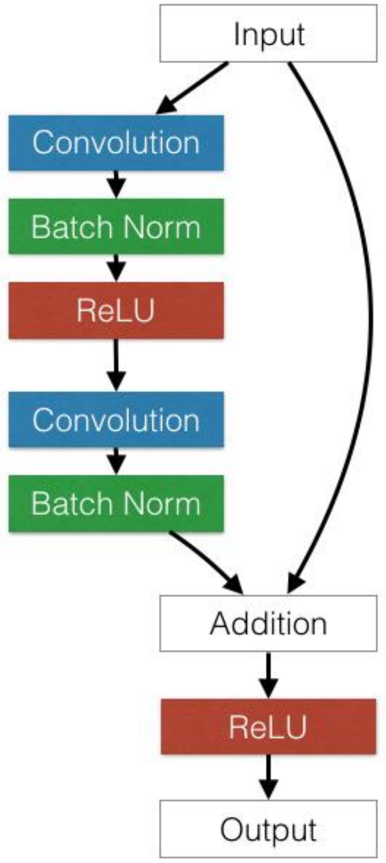
- The deeper we go the CNN extract lower and more complex features.
- Inception do reduction through 1x1 conv, before carrying out each of these expensive 3x3 and 5x5 conv.
- These 1x1 conv allows to models more complexity.

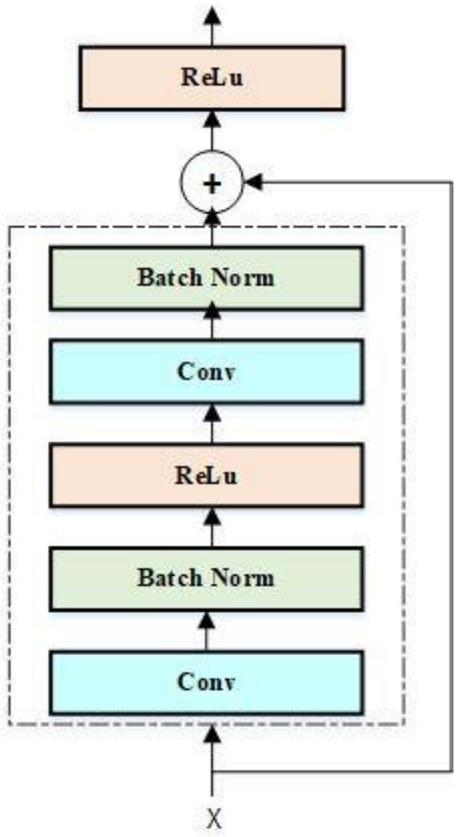


Full Inception module

ResNet (2015)

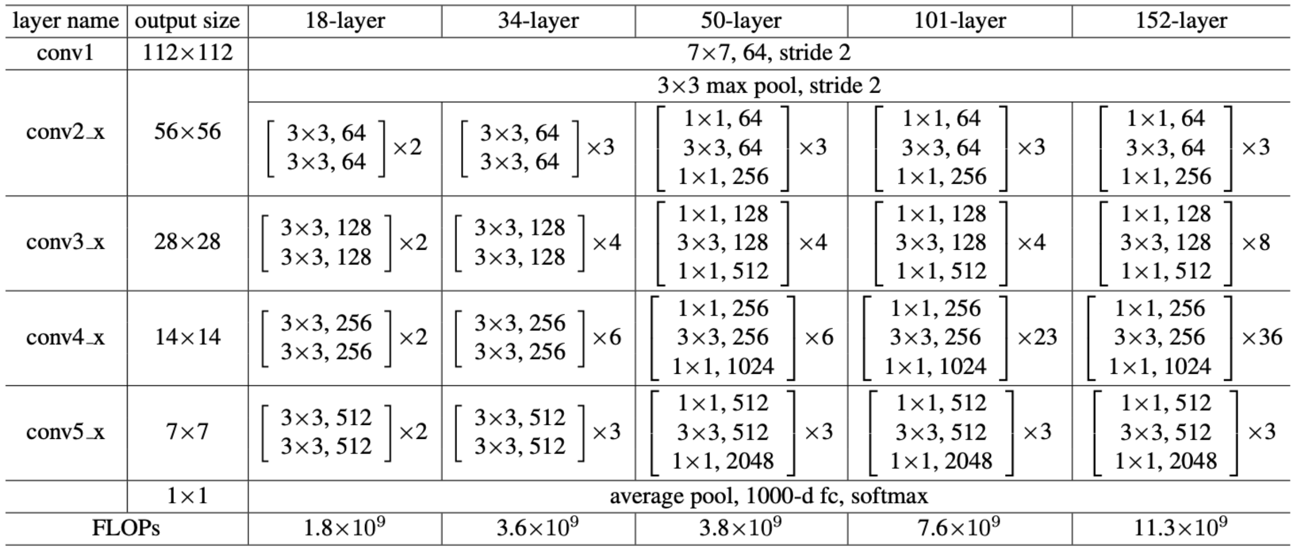
- As deep neural networks are both time-consuming to train and prone to overfitting, a team at Microsoft introduced a residual learning framework to improve the training of networks that are substantially deeper than those used previously.
- ResNet had 152 layers. (8 * VGG19)
- When adding layers, performance increases only up to a certain depth, and then it rapidly decreases. (vanishing gradient problem)
- The ResNet team added connections that can skip layers(skip connection).
- Skip connection : Add the original input to the output of the convolution block.
- Activation function ReLU , a >= 0 .
- The residual block, which adds the output of the previous layer to the output of the next layer.
- ResNet uses a technique called "residual mapping" .





RESNET ARCHITECTURE

- It has different depth variations, such as 18, 34, 50, 101, and 152 layers.
- The ResNet architecture is a stack of residual blocks.
- Every residual block has same 3 x 3 convolution (equal dimension feature vectors, so rather that a fully connected layer these are actually convolution layers, but because they are same convolution, dimension are preserved) layers.
- After the last conv layer, a global average pooling (GAP) layer is added.
- There is only one fully connected layer to classify 1000 classes.
- For a deeper network, say more than 50 layers, it uses the bottleneck features concept to improve efficiency.
- No dropout is used in this network.

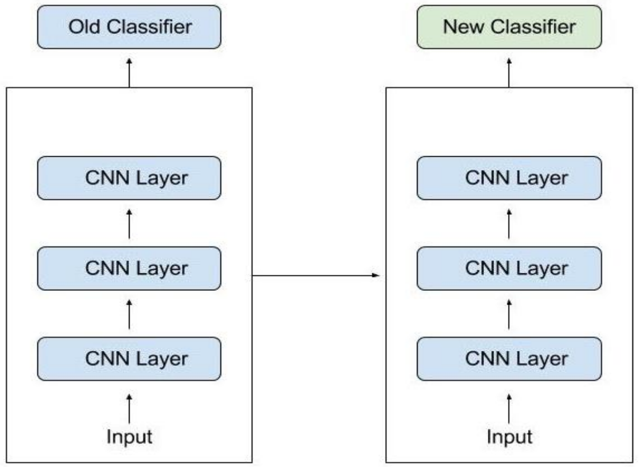


layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Each ResNet block is either 2 layers deep (used in small networks like ResNet 18 or 34), or 3 layers deep (ResNet 50, 101, or 152).

TRANSFER LEARNING

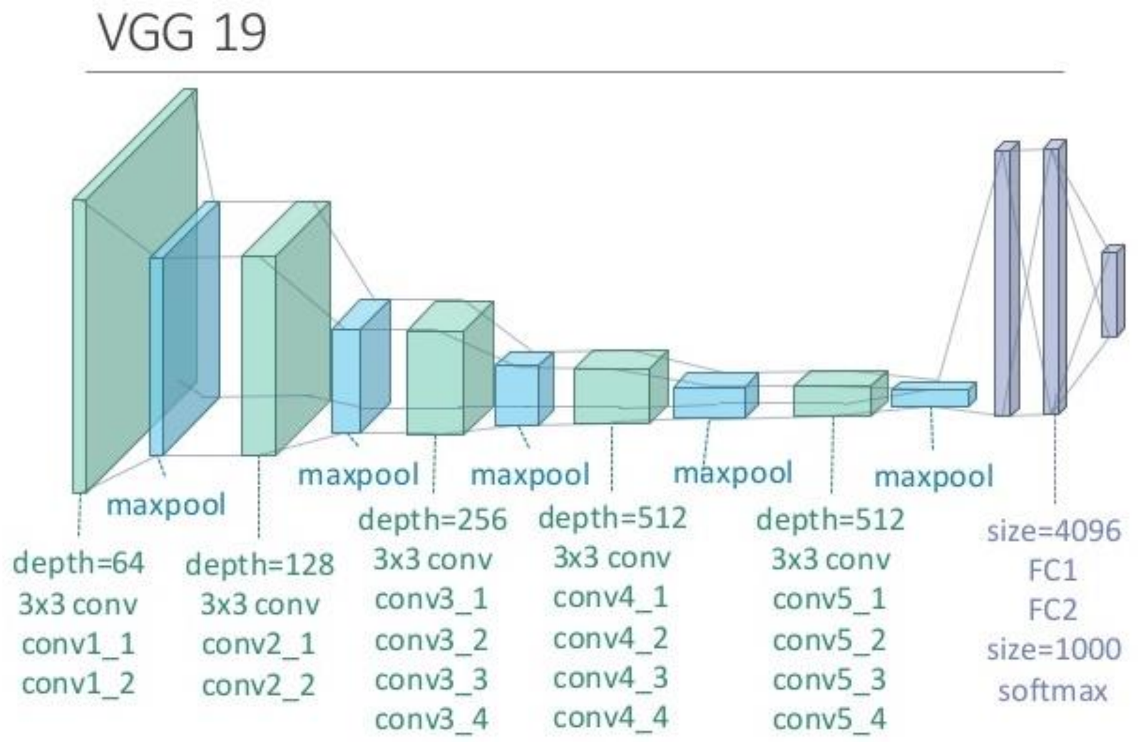
- Transfer learning, used in machine learning, is the reuse of a pre-trained model on a new problem. In transfer learning, a machine exploits the knowledge gained from a previous task to improve generalization about another.
- With transfer learning, we basically try to exploit what has been learned in one task to improve generalization in another. We transfer the weights that a network has learned at "task A" to a new "task B“.
- Transfer learning has several benefits, but the main advantages are saving training time, better performance of neural networks (in most cases), and not needing a lot of data.



How Transfer Learning Works

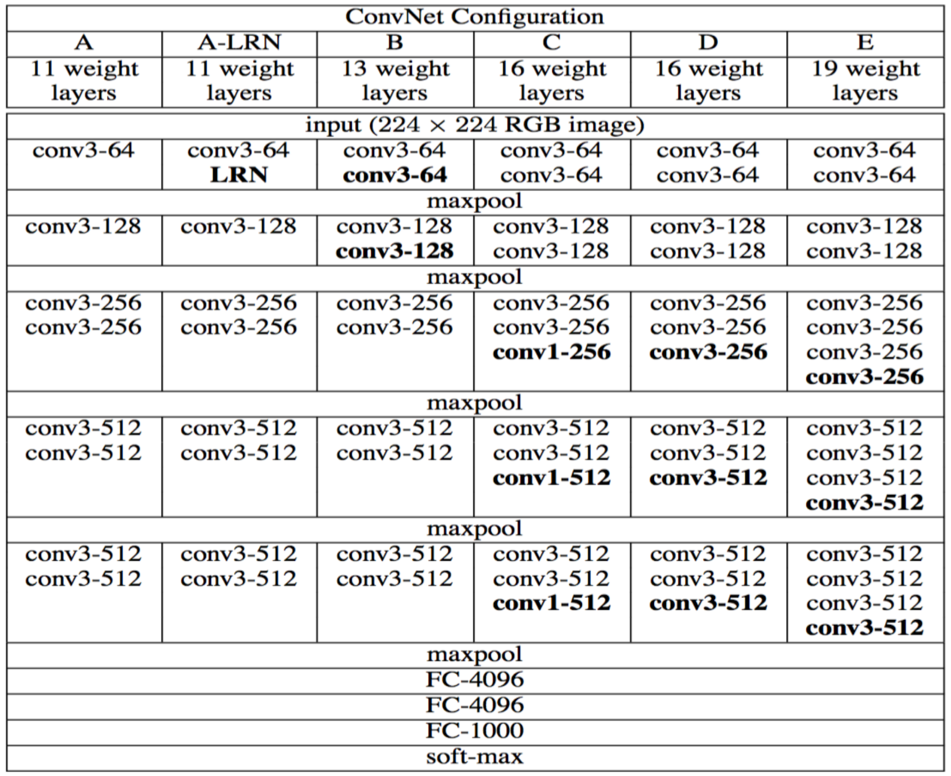
VGG19 (2015)

- VGG-19 is a trained Convolutional Neural Network, from Visual Geometry Group, Department of Engineering Science, University of Oxford.
- VGG19 is a variant of VGG model which in short consists of 19 layers (16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer).
- A fixed size of (224 x 224) RGB image was given as input to this network which means that the matrix was of shape (224,224,3).
- Used kernels of (3 * 3) size with a stride size of 1 pixel.
- max pooling was performed over a 2 * 2 pixel windows with stride 2.
- This was followed by Rectified linear unit(ReLu) as activation.
- Three fully connected layers from which first two were of size 4096 and after that a softmax layer with 1000 channels.



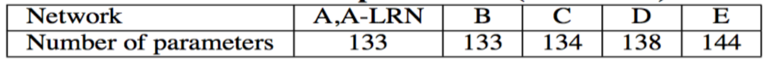
ARCHITECTURE

- A fixed size of (224 * 224) RGB image was given as input to this network which means that the matrix was of shape (224,224,3).
- The only preprocessing that was done is that they subtracted the mean RGB value from each pixel, computed over the whole training set.
- Used kernels of (3 * 3) size with a stride size of 1 pixel, this enabled them to cover the whole notion of the image.
- Spatial padding was used to preserve the spatial resolution of the image.
- Max pooling was performed over a 2 * 2 pixel windows with stride 2.
- This was followed by Rectified linear unit(ReLu) to introduce non-linearity to make the model classify better and to improve computational time as the previous models used tanh or sigmoid functions this proved much better than those.
- Implemented three fully connected layers from which first two were of size 4096 and after that a layer with 1000 channels and the final layer is a softmax function.



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).



Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

The column E in the following table is for VGG19 (other columns are for other variants of VGG models)

INCEPTION V3 (2015)

InceptionV3 mainly focuses on burning less computational power by modifying the previous Inception architectures.

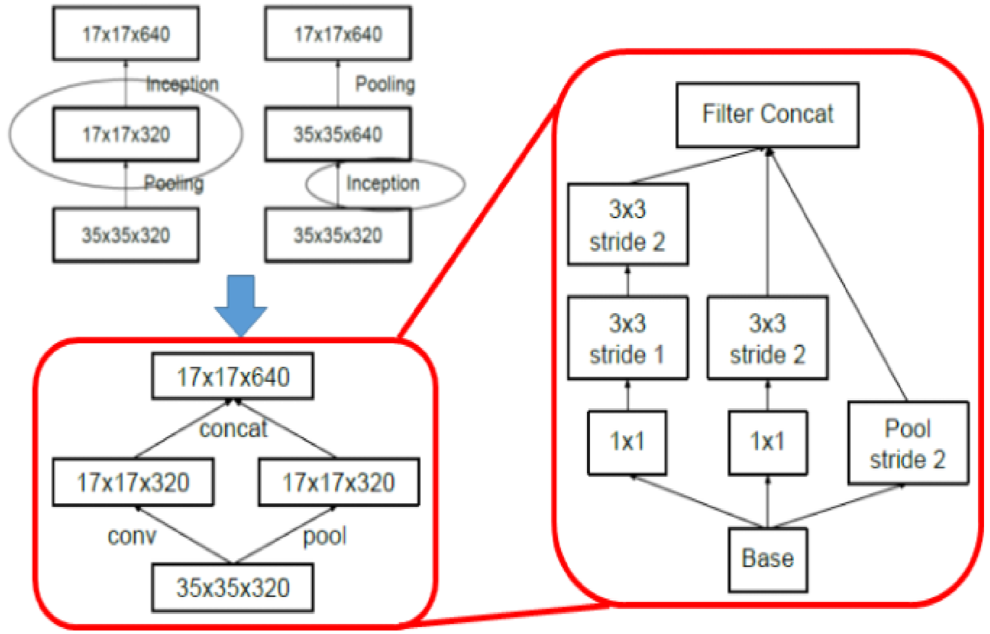
1. **Factorized Convolutions:** this helps to reduce the computational efficiency as it reduces the number of parameters involved in a network. It also keeps a check on the network efficiency.

2. **Smaller convolutions:** replacing bigger convolutions with smaller convolutions definitely leads to faster training. Say a 5 × 5 filter has 25 parameters; two 3 × 3 filters replacing a 5 × 5 convolution has only 18 (3*3 + 3*3) parameters instead.

3. **Asymmetric convolutions:** A 3 × 3 convolution could be replaced by a 1 × 3 convolution followed by a 3 × 1 convolution. If a 3 × 3 convolution is replaced by a 2 × 2 convolution, the number of parameters would be slightly higher than the asymmetric convolution proposed.

4. **Auxiliary classifier:** an auxiliary classifier is a small CNN inserted between layers during training, and the loss incurred is added to the main network loss. In GoogLeNet auxiliary classifiers were used for a deeper network, whereas in Inception v3 an auxiliary classifier acts as a regularizer.

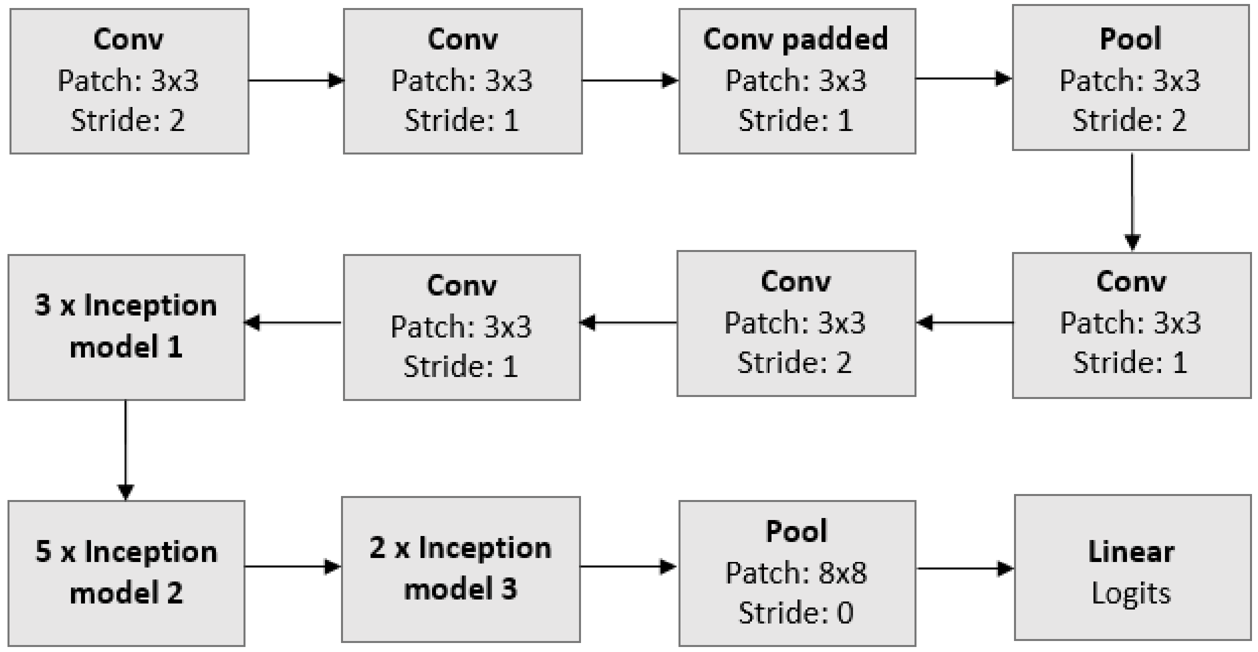
5. **Grid size reduction:** Grid size reduction is usually done by pooling operations.



Conventional downsizing (Top Left), Efficient Grid Size Reduction (Bottom Left), Detailed Architecture of Efficient Grid Size Reduction (Right)

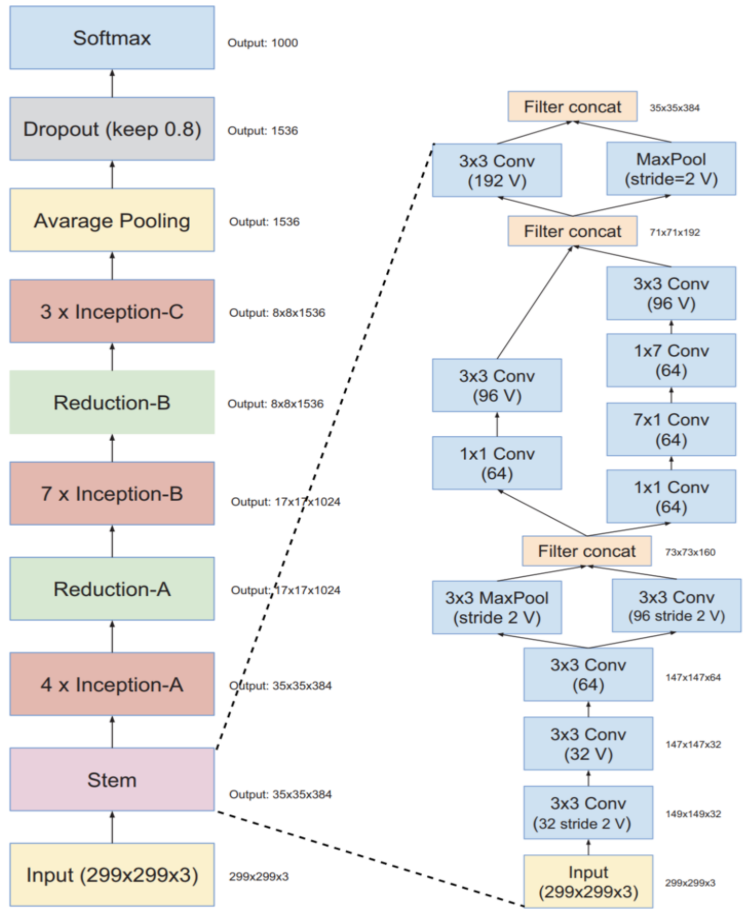
InceptionV3 layer architecture

type	patch size/stride or remarks	input size
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	As in figure 5	35×35×288
5×Inception	As in figure 6	17×17×768
2×Inception	As in figure 7	8×8×1280
pool	8 × 8	8 × 8 × 2048
linear	logits	1 × 1 × 2048
softmax	classifier	1 × 1 × 1000



INCEPTION V4(2016)

- A more uniform simplified architecture and more inception modules than Inception-v3.
- The initial set of layers "**stem of the architecture**" was modified to make it more uniform . These layers are used before **Inception block** in the architecture.
- This model can be trained **without partition of replicas** unlike the previous versions of inceptions which required different replica in order to fit in memory.
- This architecture use memory optimization on **back propagation** to **reduce the memory requirement**.



Xception

Performs convolution in 2 phases:

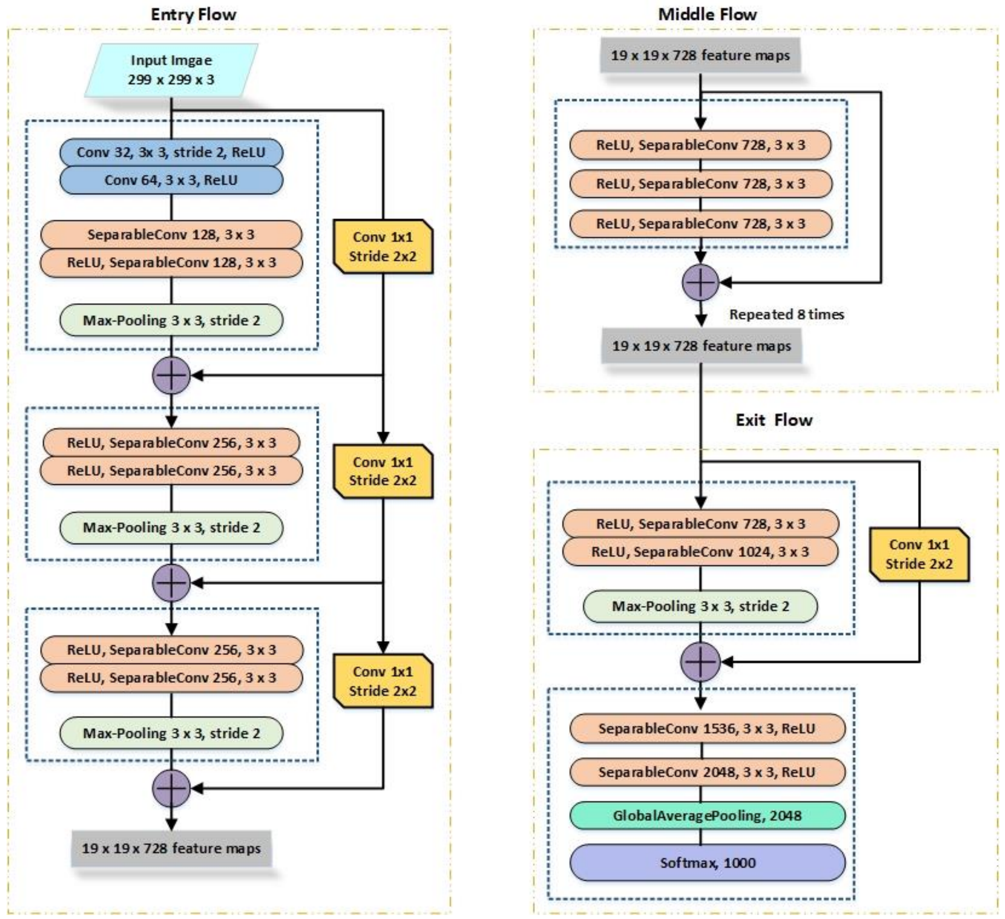
1. Depth wise Convolution (which determines spatial correlation), this is the relationship of pixels with respect its neighbors.

2. Pointwise Convolution (which determines cross-channel correlation), the relationship between points in the feature volume along the different channels.

Xception :

A simplified Inception module with out the pooling layers with only having 3 x 3 convolutions.

An “extreme” version of our Inception module, we first use 1x1 convolution to mapped cross channel correlations and then separately mapped spatial correlations of every output channels.

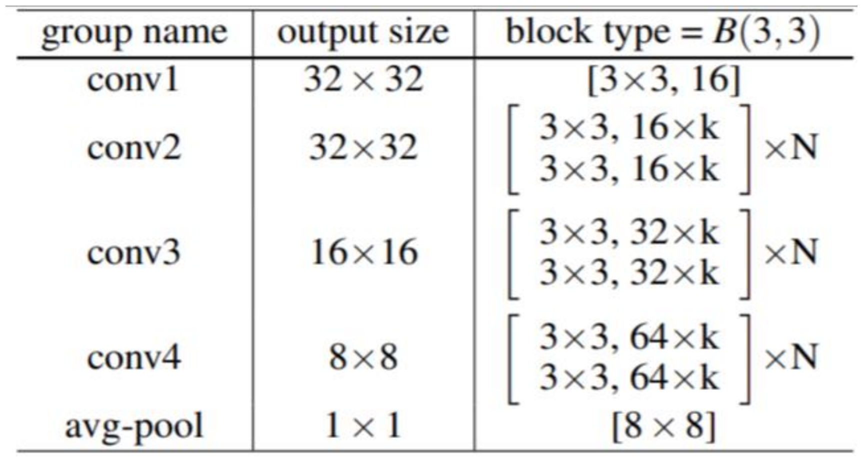

Entry Flow
Input Imgae 299 x 299 x 3
Conv 32, 3x 3, stride 2, ReLU
Conv 64, 3 x 3, ReLU
SeparableConv 128, 3 x 3
ReLU, SeparableConv 128, 3 x 3
Max-Pooling 3 x 3, stride 2
Conv 1x1 Stride 2x2
ReLU, SeparableConv 256, 3 x 3
ReLU, SeparableConv 256, 3 x 3
Max-Pooling 3 x 3, stride 2
Conv 1x1 Stride 2x2
ReLU, SeparableConv 256, 3 x 3
ReLU, SeparableConv 256, 3 x 3
Max-Pooling 3 x 3, stride 2
Conv 1x1 Stride 2x2
19 x 19 x 728 feature maps
Middle Flow
19 x 19 x 728 feature maps
ReLU, SeparableConv 728, 3 x 3
ReLU, SeparableConv 728, 3 x 3
ReLU, SeparableConv 728, 3 x 3
Repeated 8 times
19 x 19 x 728 feature maps
Exit Flow
ReLU, SeparableConv 728, 3 x 3
ReLU, SeparableConv 1024, 3 x 3
Max-Pooling 3 x 3, stride 2
Conv 1x1 Stride 2x2
SeparableConv 1536, 3 x 3, ReLU
SeparableConv 2048, 3 x 3, ReLU
GlobalAveragePooling, 2048
Softmax, 1000

WIDE RESNET (2016)

- Deep Residual Networks have shown remarkable accuracies that have aided in performing tasks like image recognition, among many others, with relative ease.
- **Convolution type:** B(3, 3) shown above has 3 × 3 convolutional layers in the residual block. Other possibilities, such as integrating 3 × 3 convolutional layers with 1 × 1 convolutions, could be explored as well.
- **Convolutions per block:** The depth of the block has to be determined by estimating the dependency of this metric on the performance of the model.
- **Width of residual blocks:** The width and depth have to be checked consistently to keep an eye on both computational complexity and performance.
- **Dropout:** A dropout layer should be added between convolutions in every residual block. This prevents overfitting.

Wide ResNet Architecture

A Wide ResNet has a group of ResNet blocks stacked together, where each ResNet block follows the BatchNormalization-ReLU-Conv structure. This structure is depicted as follows:



group name	output size	block type = $B(3,3)$
conv1	32×32	$[3{\times}3, 16]$
conv2	$32{\times}32$	$\begin{bmatrix} 3{\times}3, 16{\times}k \\ 3{\times}3, 16{\times}k \end{bmatrix} \times N$
conv3	$16{\times}16$	$\begin{bmatrix} 3{\times}3, 32{\times}k \\ 3{\times}3, 32{\times}k \end{bmatrix} \times N$
conv4	$8{\times}8$	$\begin{bmatrix} 3{\times}3, 64{\times}k \\ 3{\times}3, 64{\times}k \end{bmatrix} \times N$
avg-pool	1×1	$[8 \times 8]$

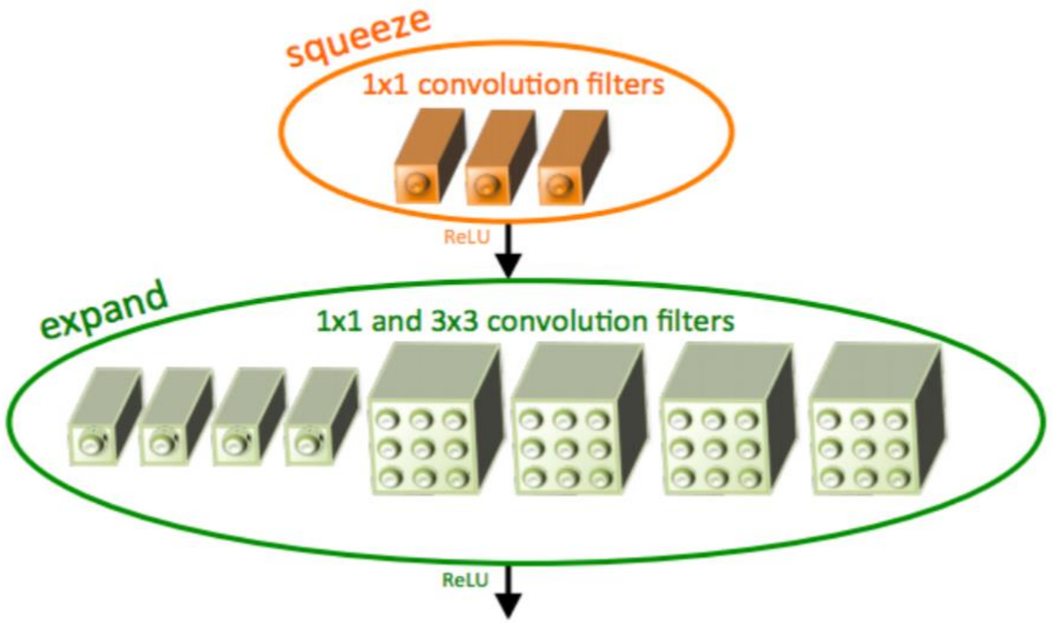
There are **five groups** that comprise a **wide ResNet**. The block here refers to the **residual block B(3, 3). Conv1** remains intact in any network, whereas **conv2**, **conv3**, and **conv4** vary according to ***k***, a value that defines the width. The convolutional layers are succeeded by an **average-pool layer** and **a classification layer**.

SQUEEZENET (2016)

- SqueezeNet is a smaller network that was designed as a more compact replacement for AlexNet.
- It has almost 50x fewer parameters than AlexNet, yet it performs 3x faster.
- This architecture was proposed by researchers at DeepScale, The University of California, Berkeley, and Stanford University in the year 2016.

- key ideas behind SqueezeNet:

Strategy One : use 1 × 1 filters instead of 3 × 3
Strategy Two : decrease the number of input channels to 3 × 3 filters
Strategy Three: Down sample late in the network so that convolution layers have large activation maps



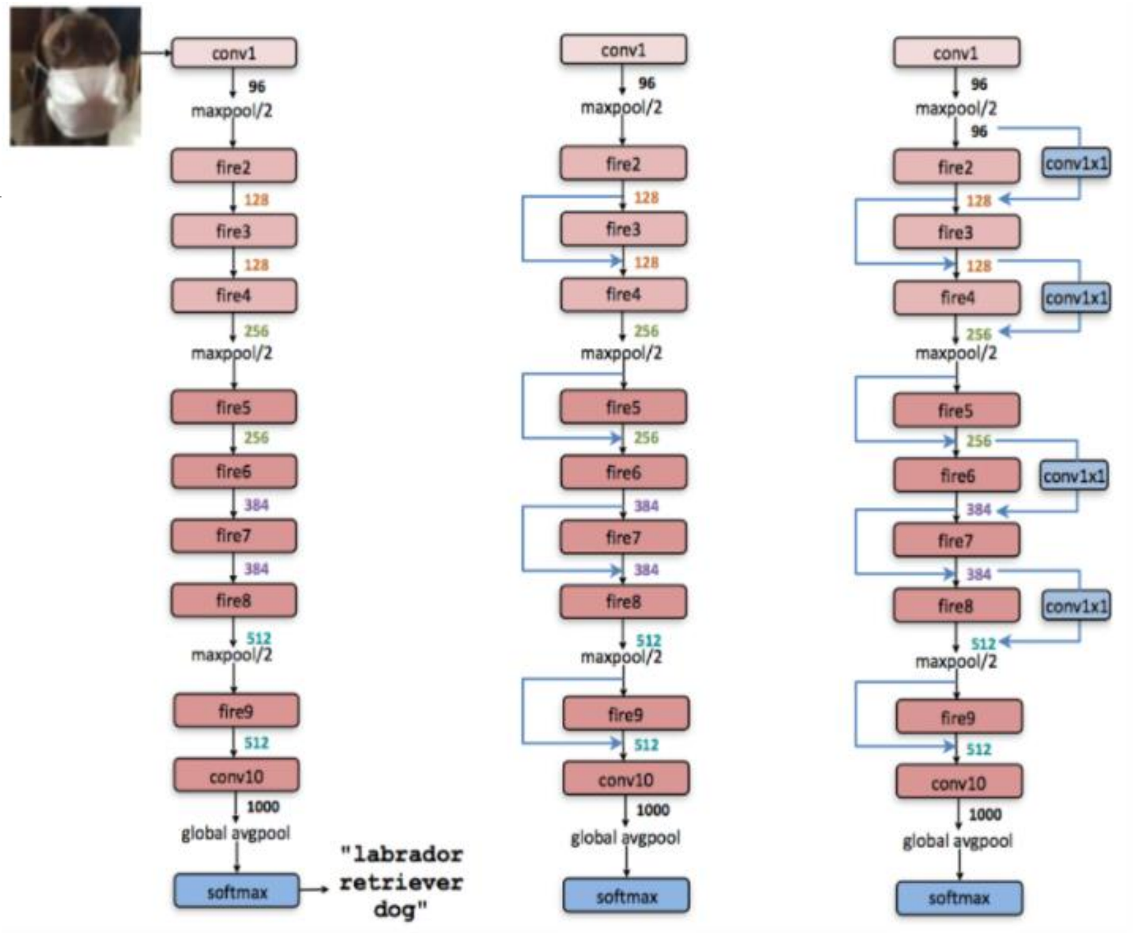
"Fire Module"

The SqueezeNet architecture is comprised of "squeeze" and "expand" layers. A squeeze convolutional layer has only 1 × 1 filters. These are fed into an expand layer that has a mix of 1 × 1 and 3 × 3 convolution filters. This is shown above.

An input image is first sent into a standalone convolutional layer. This layer is followed by 8 "fire modules" which are named “fire2-9”, according to Strategy One.

Following Strategy Two, the filters per fire module are increased with "simple bypass." Lastly, SqueezeNet performs max-pooling with a stride of 2 after layers conv1, fire4, fire8, and conv10.

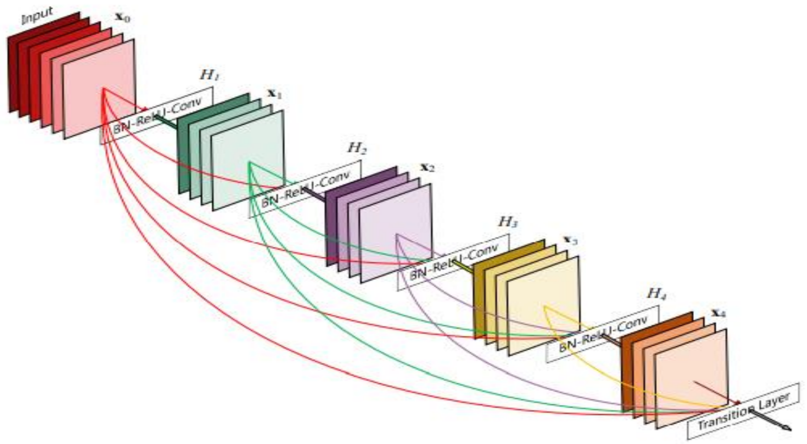
According to Strategy Three, pooling is given a relatively late placement, resulting in SqueezeNet with a "complex bypass" (the rightmost architecture in the image)



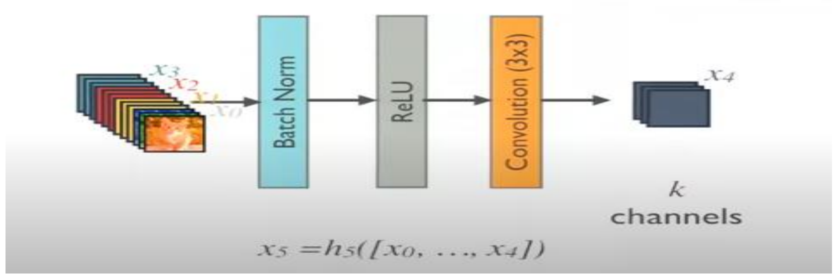
From left to right: SqueezeNet, SqueezeNet with simple bypass, and SqueezeNet with complex bypass

DENSENET (2016)

- Densely Connected Convolutional Networks , DenseNets, are the next step on the way to keep increasing the depth of deep convolutional networks.
- DenseNet, which connects each layer to every other layer in a feed-forward fashion. (Forward propagation)
- Instead of drawing representational power from extremely deep or wide architectures, DenseNets exploit the potential of the network through feature reuse.
- Connectivity Pattern : Feature maps from previous layers are concatenated onto the inputs of future layers(also known as dense connectivity).
- Each of the linear transformation, followed by Batch Norm, ReLU, Convolution with filter size (3 x 3).

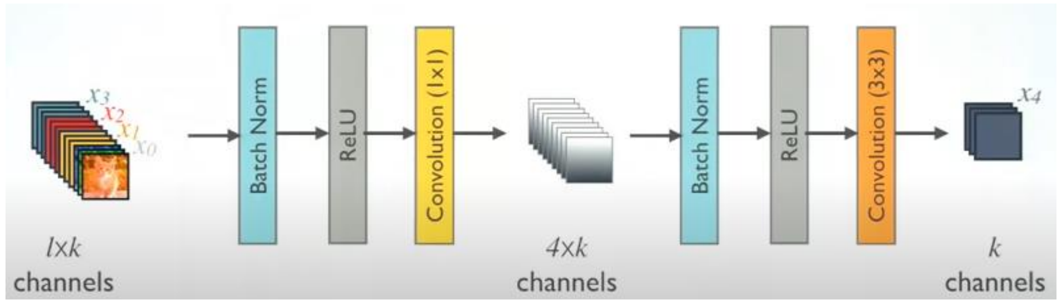


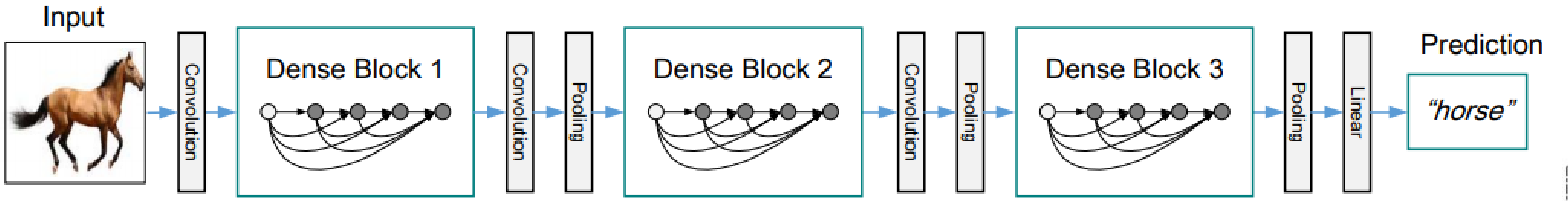
A 5-layer dense block with a growth rate of k = 4. Each layer takes all preceding feature-maps as input.



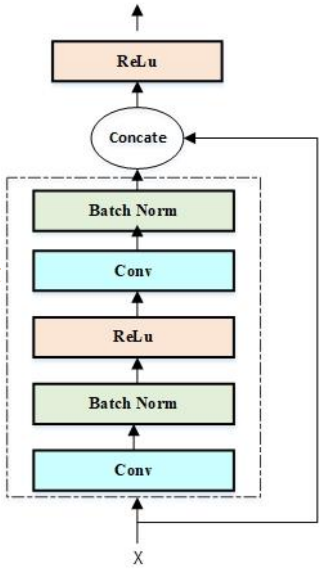
Composite layer in DenseNet

- As we are keep concatenating features into the network, the input to deeper layers will become very wide and these may introduce to much computational for deeper layers.
- To address this problem, using relatively cheaper convolution with filter size 1 x 1 (reduce the dimension of the channels) to 4 x k channels.



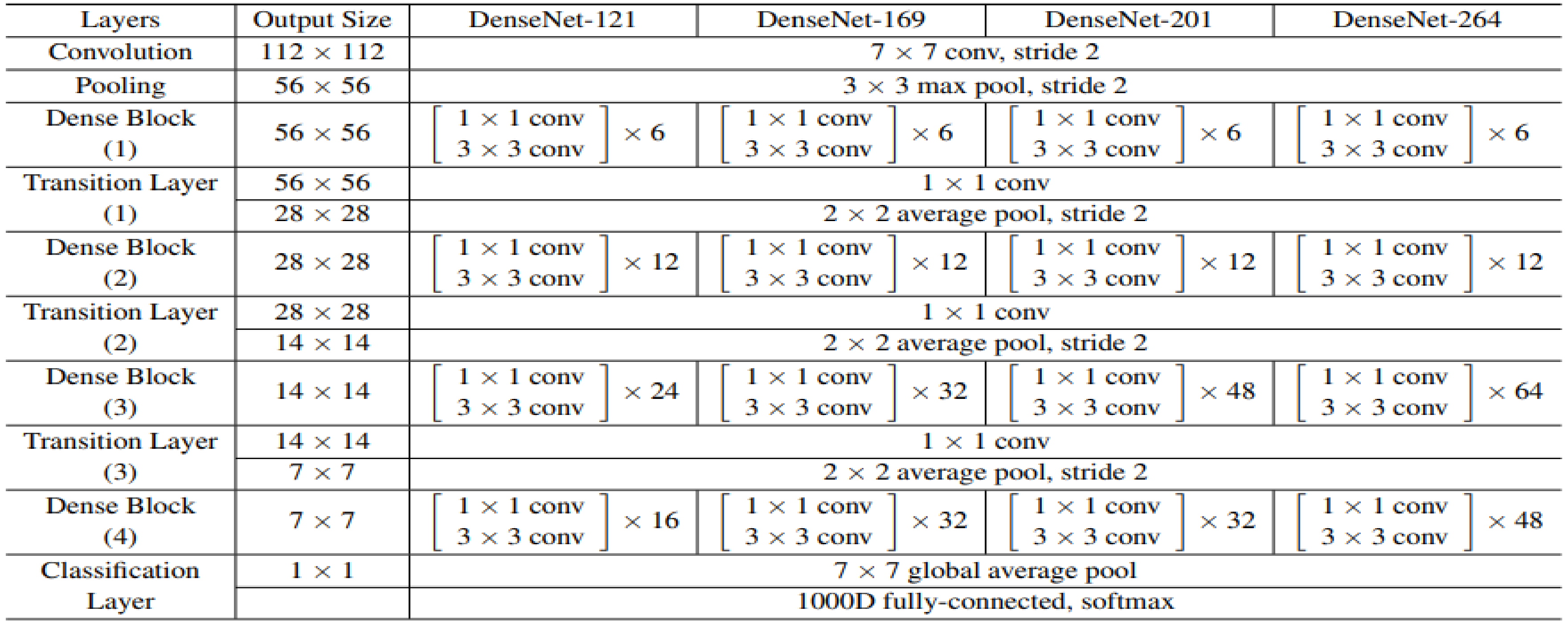


A deep DenseNet with **three dense blocks**. The layers between two adjacent blocks are referred to as **transition layers** and change feature-map sizes via **convolution** and **pooling**



- Perform down-sampling on feature maps : Controlling depth with 1 x 1 convolutions and pooling (reduces feature map sizes).
- Continuously concatenating feature maps will result in very deep inputs.
- Channel wise concatenation, each layer dense and slim.
- High computational efficiency and parameter efficiency.

DENSENET ARCHITECTURES FOR IMAGENET

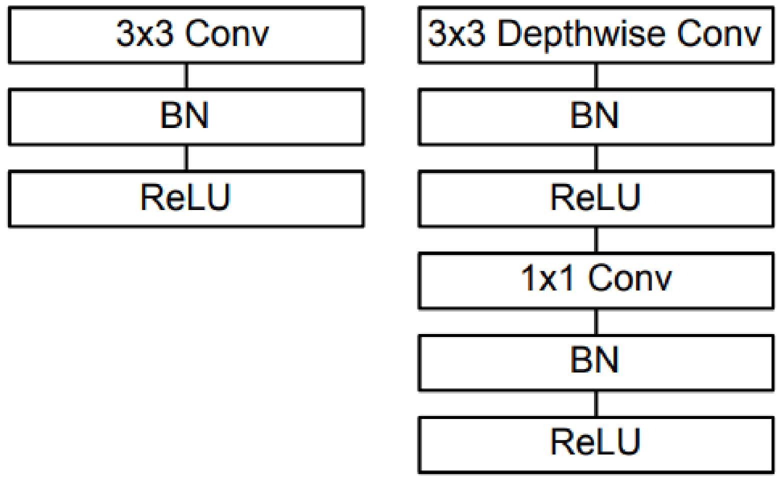


Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

The growth rate for all the networks is **k = 32**. Note that each "**conv**" layer shown in the table corresponds the sequence **BN-ReLU-Conv**.

MOBILENETS (2017)

- As the name applied, the MobileNet model is designed to be used in Mobile Vision Applications, and it is TensorFlow's first mobile computer vision model.
- MobileNet uses depthwise separable convolutions. It significantly reduces the number of parameters when compared to the network with regular convolutions with the same depth in the nets. This results in lightweight deep neural networks.
- A depthwise separable convolution is made from two operations.
  1. Depthwise convolution (Filtering Stage).
  2. Pointwise convolution (Combination Stage).
- Instead of the usual (CONV, BATCH_NORM,RELU), it splits 3x3 convolutions up into a 3x3 depthwise convolution, followed by a 1x1 Pointwise CONV.
- They call this block a depthwise separable convolution.



Left : Standard convolutional layer with batchnorm and ReLU. Right : Depthwise Separable convolutions with Depthwise and Pointwise layers followed by batchnorm and ReLU.

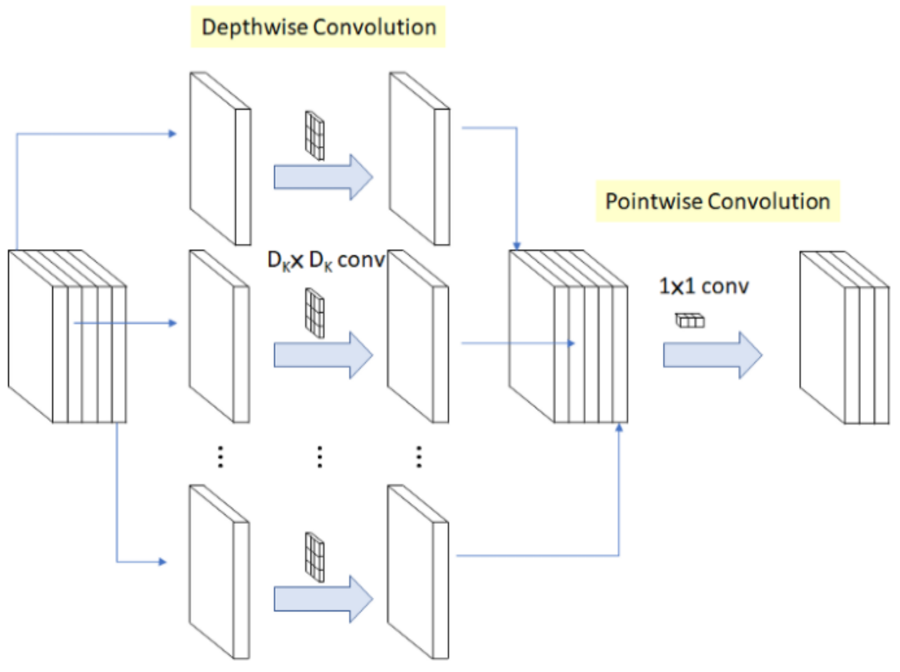
MOBILENETS

- Depthwise convolution is the channel-wise DK×DK spatial convolution. Suppose in the figure, and we have five channels; then, we will have 5 DK×DK spatial convolutions.

- Pointwise convolution is the 1×1 convolution to change the dimension.

- Control parameters :

- MobileNets uses two hyperparameters to help control the trade-off between accuracy and speed.
- Width Multiplier: Controls the Depthwise CONVs accuracy by uniformly reducing the number filters used throughout the network.
- Resolution Multiplier: Simply scales down the input image to different sizes.



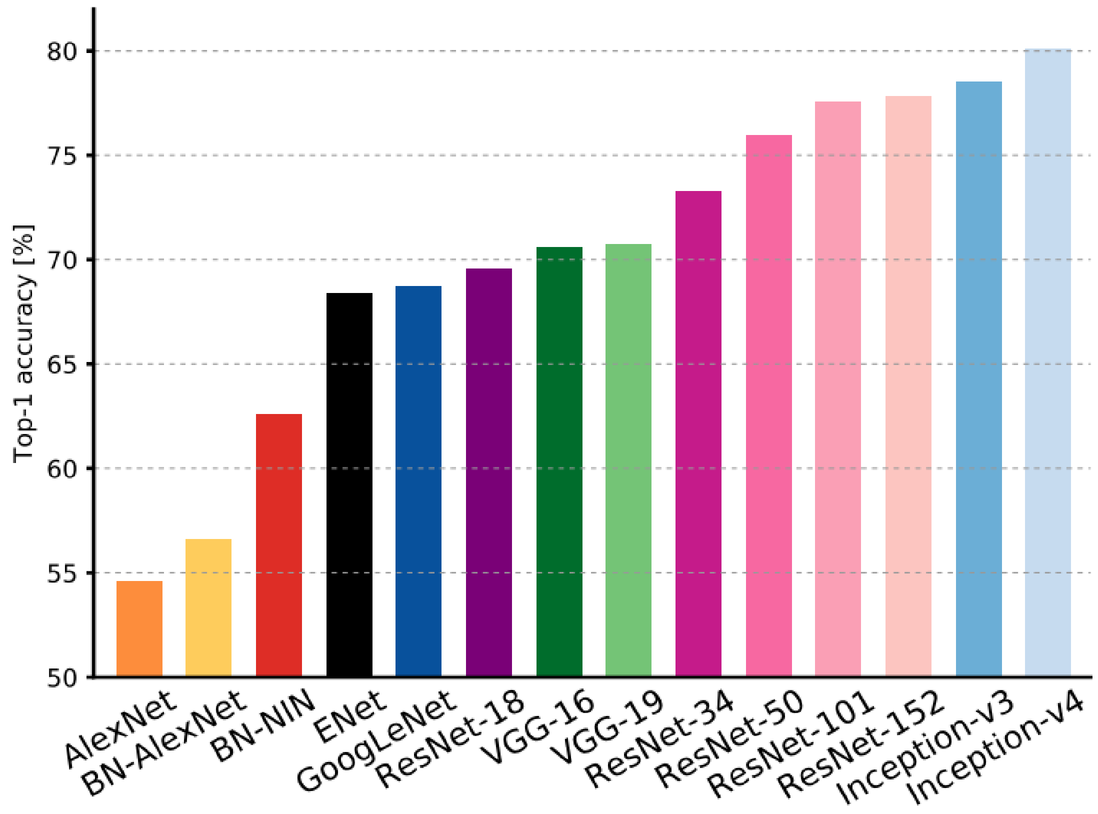
Depthwise Separable Convolution

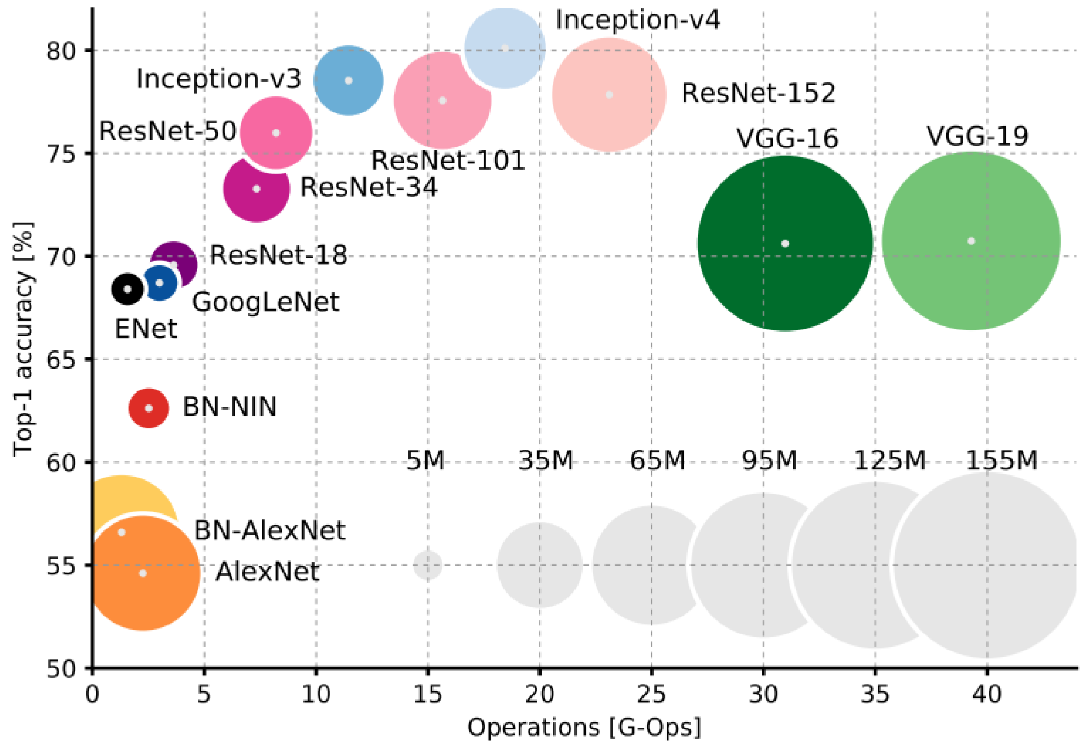
CONCLUSION

- We tried to understand what is Computer Vision.
- What is CNN and its different layers.
- Different types of CNN and its uses in Computer Vision techniques.

Comparison Top1 validation accuracy vs network

Comparison Top1 accuracy vs operations, size ∝ parameters





REFERENCES :

https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/

https://iq.opengenus.org/evolution-of-cnn-architectures/

https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5

https://arxiv.org/abs/1605.07678

https://en.wikipedia.org/wiki/Universal_approximation_theorem

Thank You