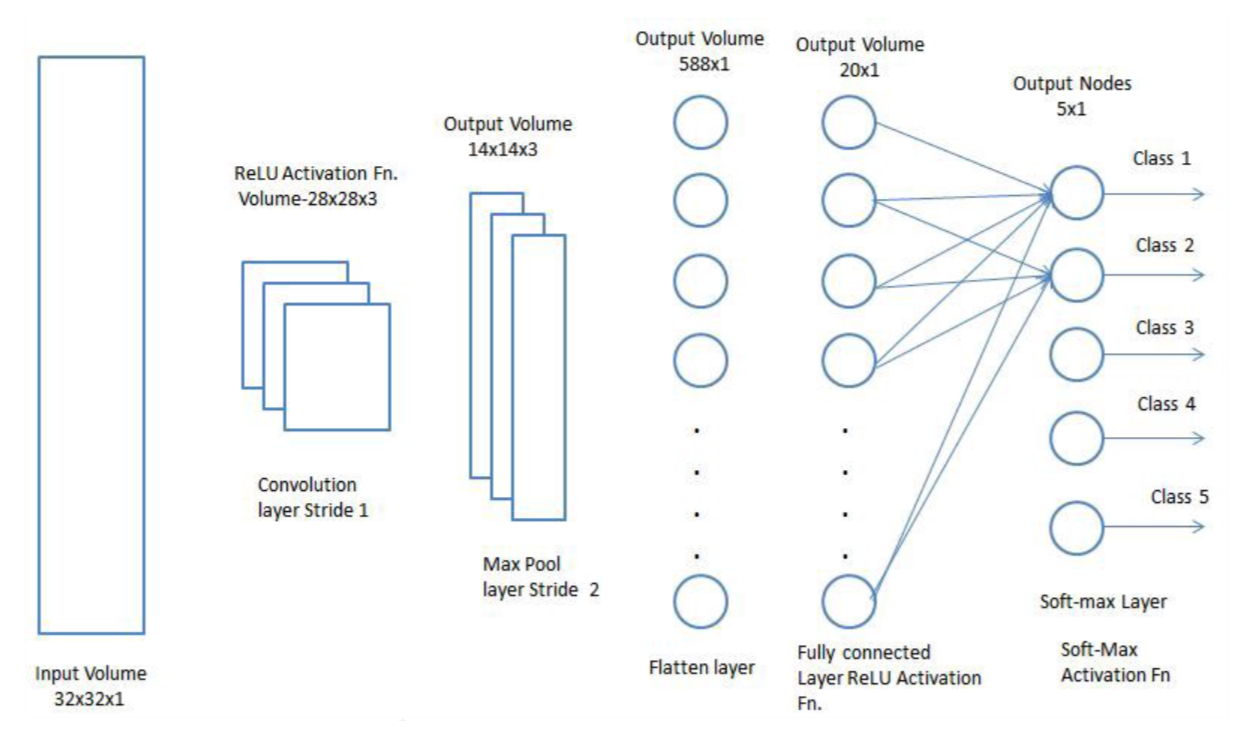
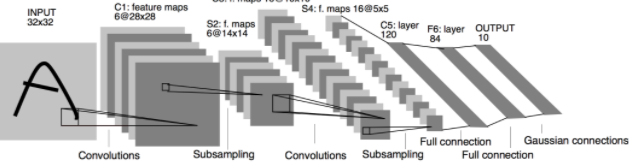
Study of Different CNN Architecture

Convolutional Neural Network : A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. A ConvNet is able to **successfully capture the Spatial and Temporal dependencies** in an image through the application of relevant filters. The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.Adding a Fully-Connected layer is a (usually) cheap way of learning non-linear combinations of the high-level features as represented by the output of the convolutional layer. The Fully-Connected layer is learning a possibly non-linear function in that space.

There are various **architectures of CNNs** available which have been key in building algorithms which power and shall power AI as a whole in the foreseeable future. Some of them have been listed below:

1. LeNet
2. AlexNet
3. VGG16/VGG19
4. ResNet
5. Inception Network
6. EfficientNet

LeNet: The LeNet architecture was first introduced by LeCun et al. in their 1998 paper, [**Gradient-Based Learning Applied to Document *Recognition***](http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf)*.* As the name of the paper suggests, the authors’ implementation of LeNet was used primarily for OCR and character recognition in documents.

The LeNet architecture is *straightforward* and small, making it *perfect for teaching the basics of CNNs.*

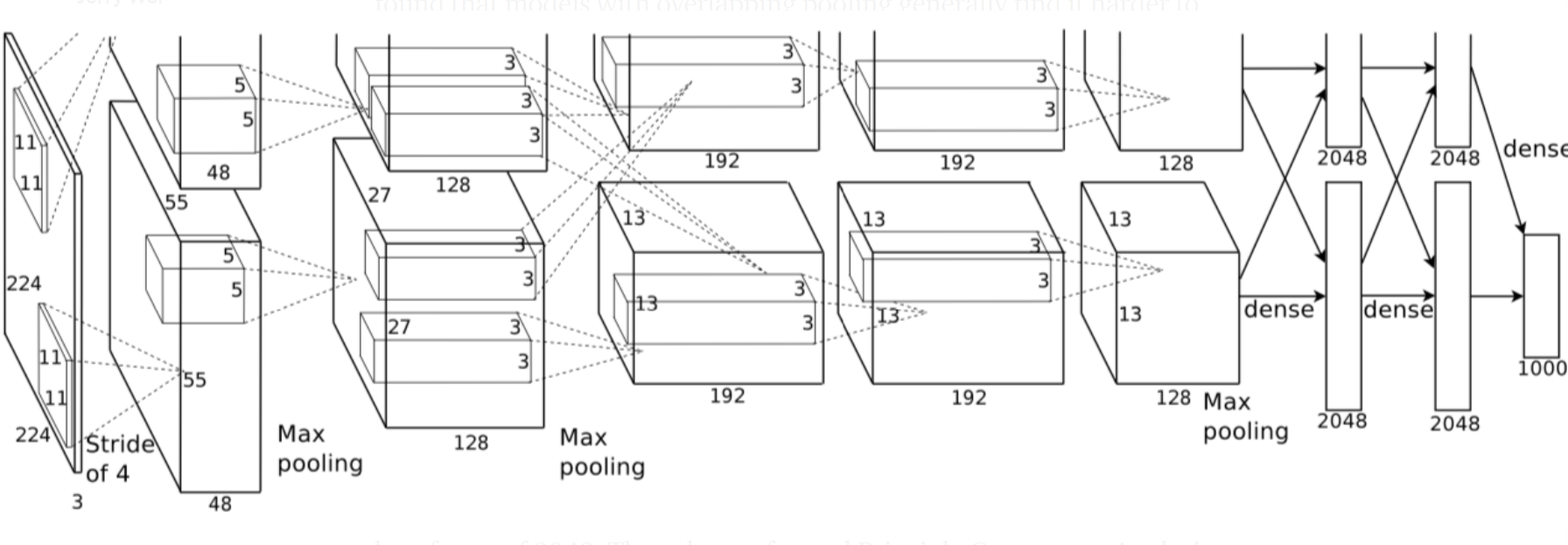
**The LeNet-5 architecture utilizes two significant types of layer construct: convolutional layers and subsampling layers.** Within the research paper and the image above, convolutional layers are identified with the ‘*Cx’*, and subsampling layers are identified with ‘*Sx’*, where ‘*x’* is the sequential position of the layer within the architecture. ‘*Fx’* is used to identify fully connected layers.

AlexNet: A few years back, we still used small datasets like CIFAR and NORB consisting of tens of thousands of images. These datasets were sufficient for machine learning models to learn basic recognition tasks. However, real life is never simple and has many more variables than are captured in these small datasets. The recent availability of large datasets like ImageNet, which consist of hundreds of thousands to millions of labeled images, have pushed the need for an extremely capable deep learning model. Then came[AlexNet](https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf).

**AlexNet -** The architecture consists of eight layers: **five convolutional layers and three fully-connected layers**. But this isn’t what makes AlexNet special; these are some of the features used that are new approaches to convolutional neural networks:

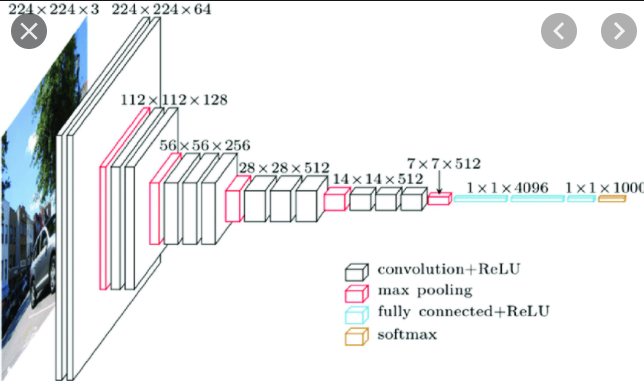
1. **ReLU Nonlinearity**
2. **Multiple GPUs**
3. **Overlapping Pooling**
4. **The Overfitting Problem.** AlexNet had 60 million parameters, a major issue in terms of overfitting.

* **Data Augmentation**
* **DropOut**



**AlexNet** is an incredibly powerful model capable of achieving high accuracies on very challenging datasets. However, removing any of the convolutional layers will drastically degrade **AlexNet’s** performance.

VGG 16: AlexNet came out in 2012 and was a revolutionary advancement; it improved on traditional Convolutional Neural Networks (CNNs) and became one of the best models for image classification… until [VGG](https://arxiv.org/pdf/1409.1556.pdf) came out.While previous derivatives of AlexNet focused on smaller window sizes and strides in the first convolutional layer, **VGG** addresses another very **important aspect of CNNs: depth**.



**Input -**  VGG takes in a 224x224 pixel RGB image.

**Convolutional Layers.** The convolutional layers in VGG use a very small receptive field (3x3, the smallest possible size that still captures left/right and up/down). There are also 1x1 convolution filters which act as a linear transformation of the input, which is followed by a ReLU unit.

**Fully-Connected Layers.** VGG has three fully-connected layers: the first two have 4096 channels each and the third has 1000 channels, 1 for each class.

**Hidden Layers.** All of VGG’s hidden layers use ReLU (a huge innovation from AlexNet that cut training time). VGG does not generally use Local Response Normalization (LRN), as LRN increases memory consumption and training time with no particular increase in accuracy.

**Differences:**

* Instead of using large receptive fields like AlexNet (11x11 with a stride of 4), VGG uses very small receptive fields (3x3 with a stride of 1).
* VGG incorporates 1x1 convolutional layers to make the decision function more non-linear without changing the receptive fields.
* The small-size convolution filters allows VGG to have a large number of weight layers; of course, more layers leads to improved performance.

InceptionV3 : The Inception network was an important milestone in the development of CNN classifiers. The Inception network on the other hand, was complex (heavily engineered). It used a lot of tricks to push performance; both in terms of speed and accuracy.

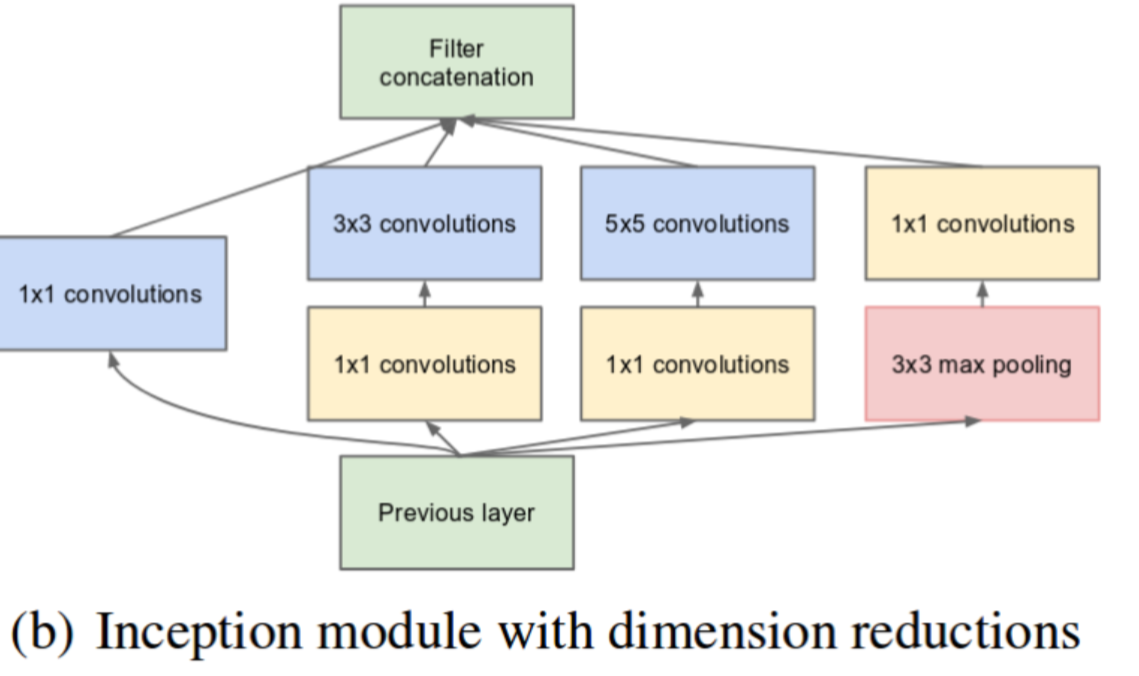
**Inceptionv3** is a CNN for assisting in image analysis and object detection, and got its start as a module for GoogleNet. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge. Just as imagenet can be thought of as a database of classified visual objects, Inception helps classification of object in the world of [computer vision](https://en.wikipedia.org/wiki/Computer_vision).

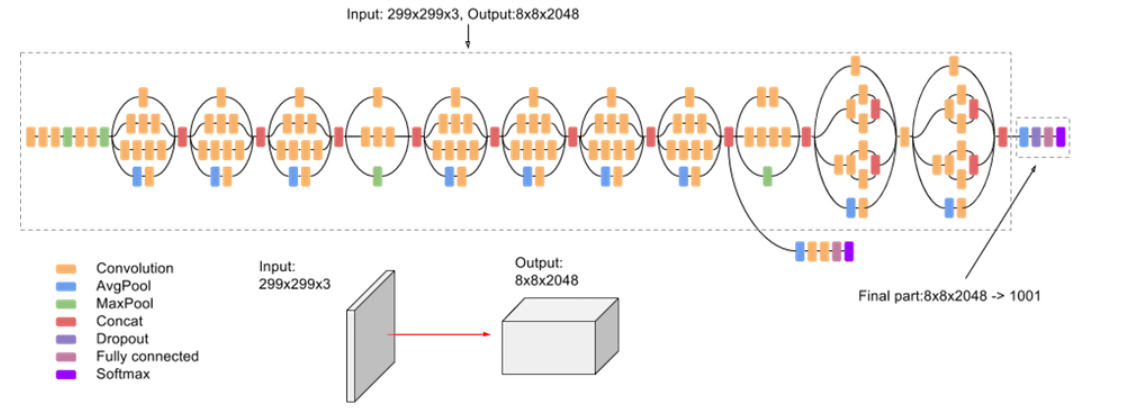
**What problem it was purported to solve?**

Because of the huge variation in the location of the information in an image choosing the **right kernel size** for the convolution operation become tough.

**Solution :**

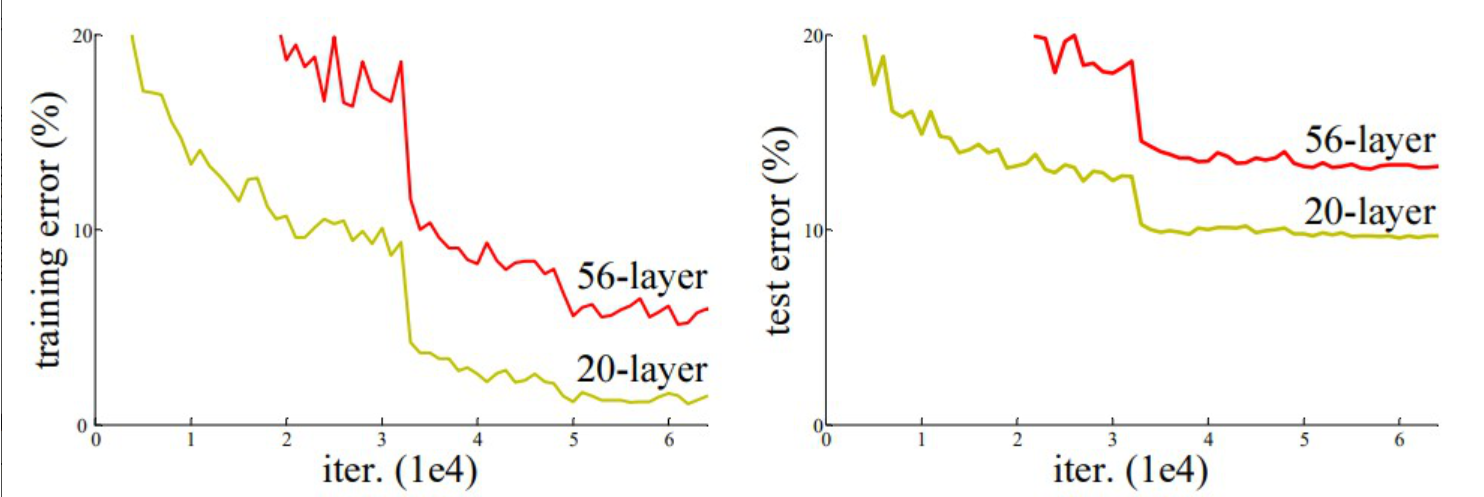
The network essentially would get a bit “**wider**” rather than “deeper”. The authors designed the inception module to reflect the same.

The outputs are **concatenated** and sent to the next inception module.Using the dimension reduced inception module, a neural network architecture was built. This was popularly known as GoogLeNet.



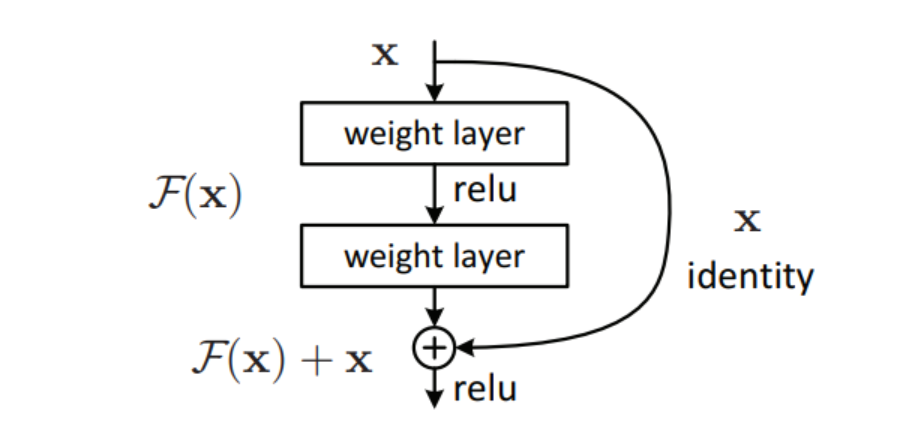
**Inception Net v3** incorporated all of the above upgrades stated for , and in addition used the following:

* RMSProp Optimizer.
* Factorized 7x7 convolutions.
* BatchNorm in the Auxillary Classifiers.
* Label Smoothing (A type of regularizing component added to the loss formula that prevents the network from becoming too confident about a class. Prevents over fitting).

ResNet : After the first CNN-based architecture (AlexNet) that win the ImageNet 2012 competition, Every subsequent winning architecture uses more layers in a deep neural network to reduce the error rate.

56-layer CNN gives more error rate on both training and testing dataset than a 20-layer CNN architecture. After analyzing more on error rate the authors were able to reach conclusion that it is caused by vanishing/exploding gradient.

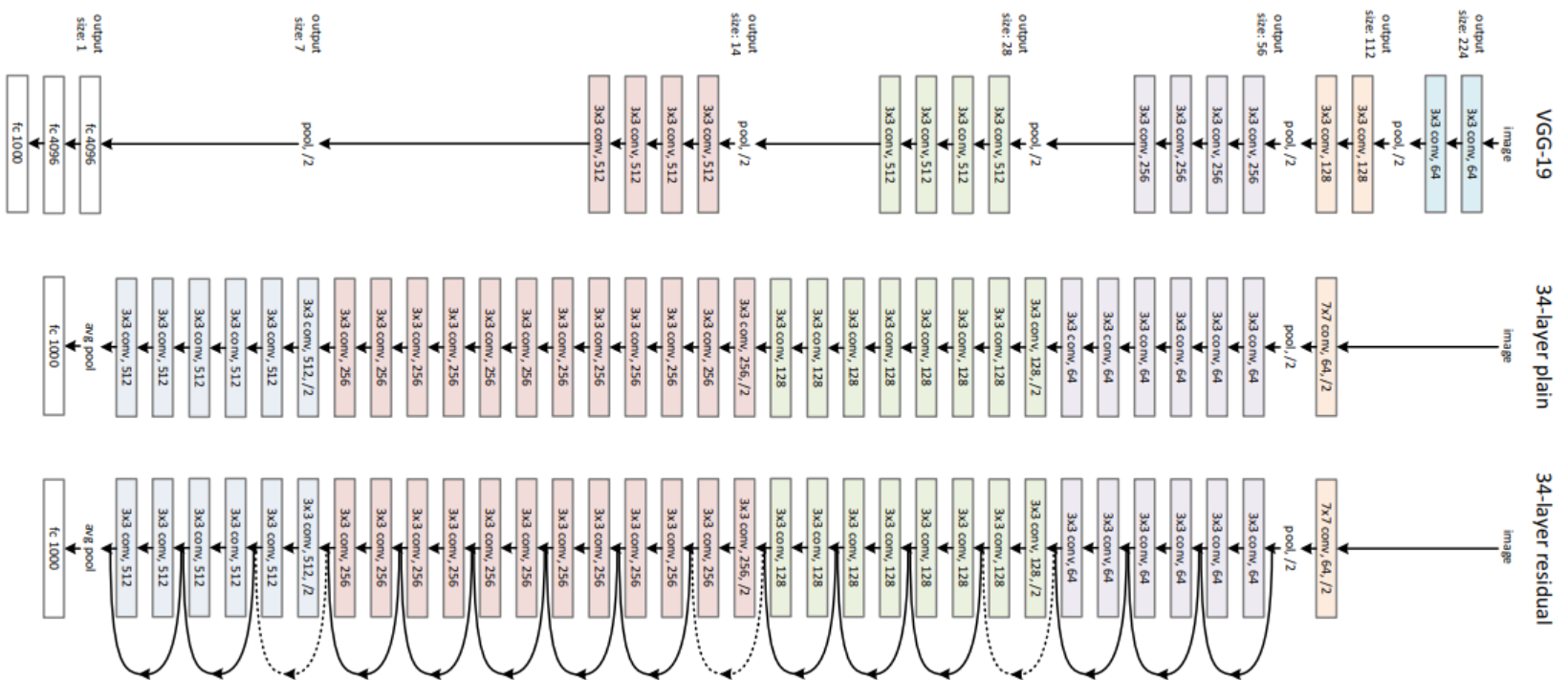
ResNet, which was proposed in 2015 by researchers at Microsoft Research introduced a new architecture called Residual Network.

**Residual Block:**  
**In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Network. In this network we use a technique called *skip connections*** . The skip connection skips training from a few layers and connects directly to the output.

The advantage of adding this type of skip connection is because if any layer hurt the performance of architecture then it will be skipped by regularization. So, this results in training very deep neural network without the problems caused by vanishing/exploding gradient.

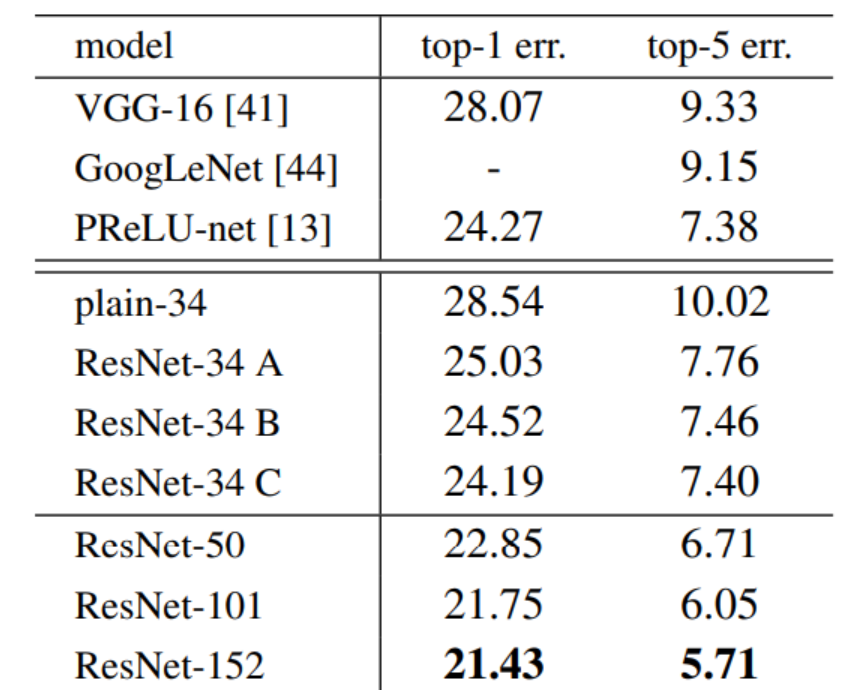
**Network Architecture:**

This network uses a 34-layer plain network architecture inspired by VGG-19 in which then the shortcut connection is added. These shortcut connections then convert the architecture into residual network.



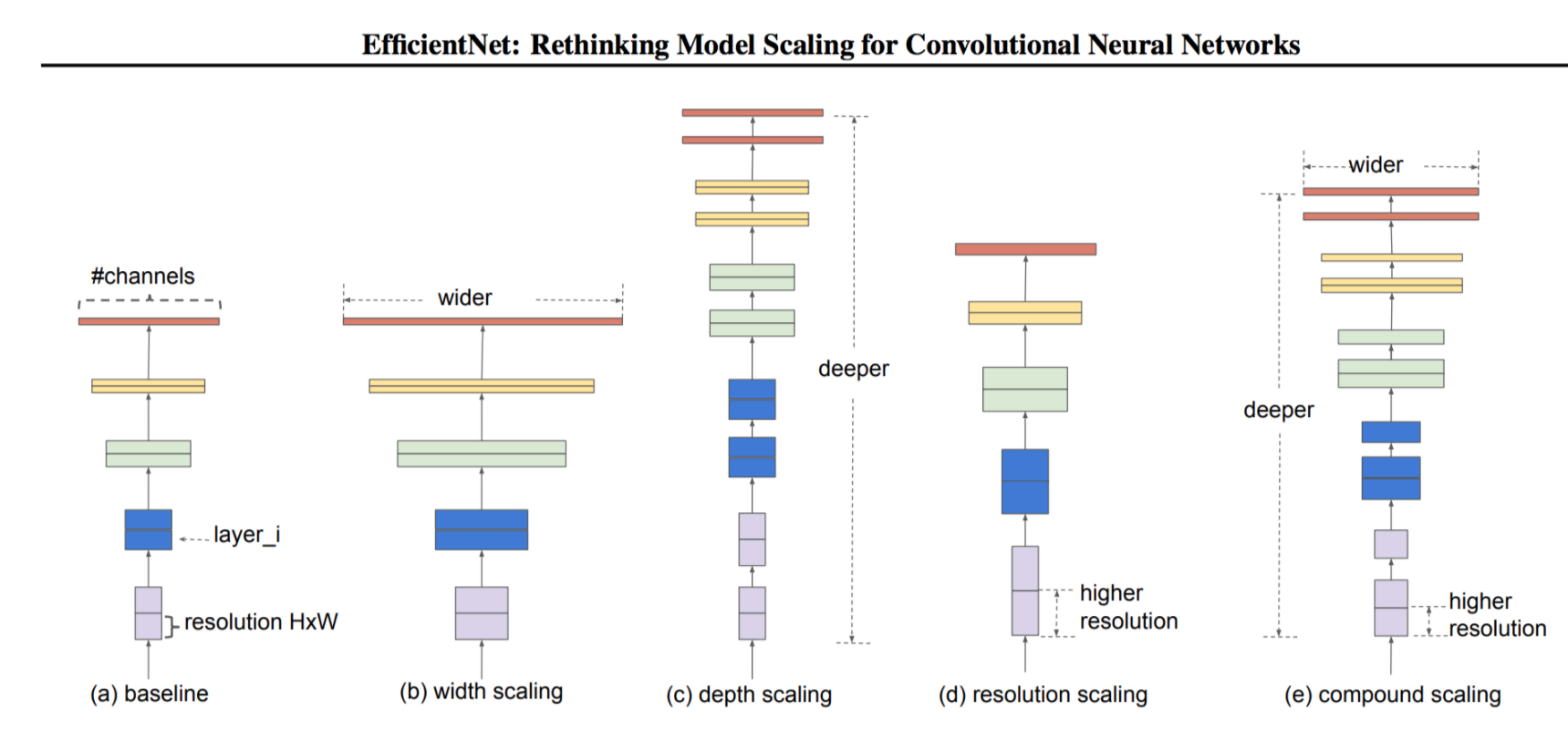
Comparision:

On the ImageNet dataset, the authors uses a 152-layers ResNet, which is 8 times more deep than VGG19 but still have less parameters. An ensemble of these ResNets generated an error of only 3.7% on ImageNet test set, the result which won ILSVRC 2015 competition. On COCO object detection dataset, it also generates a 28% relative improvement due to its very deep representation.



EfficientNet: ***EfficientNet***  not only focuses on improving the accuracy, but also the efficiency of models.

Scaling in context of CNN: There are three scaling dimensions of a CNN: *depth, width,* and *resolution.* **Depth** simply means how deep the networks is which is equivalent to the number of layers in it. **Width** simply means how wide the network is. One measure of width, for example, is the number of channels in a Conv layer whereas **Resolution** is simply the image resolution that is being passed to a CNN.

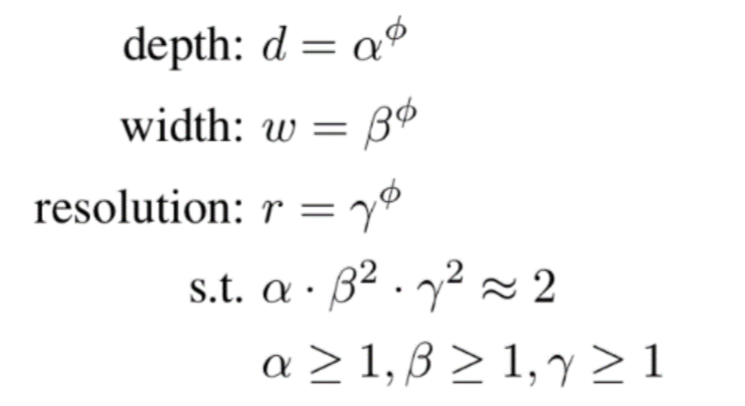


***Scaling up any dimension of network (width, depth or resolution) improves accuracy, but the accuracy gain diminishes for bigger models.***

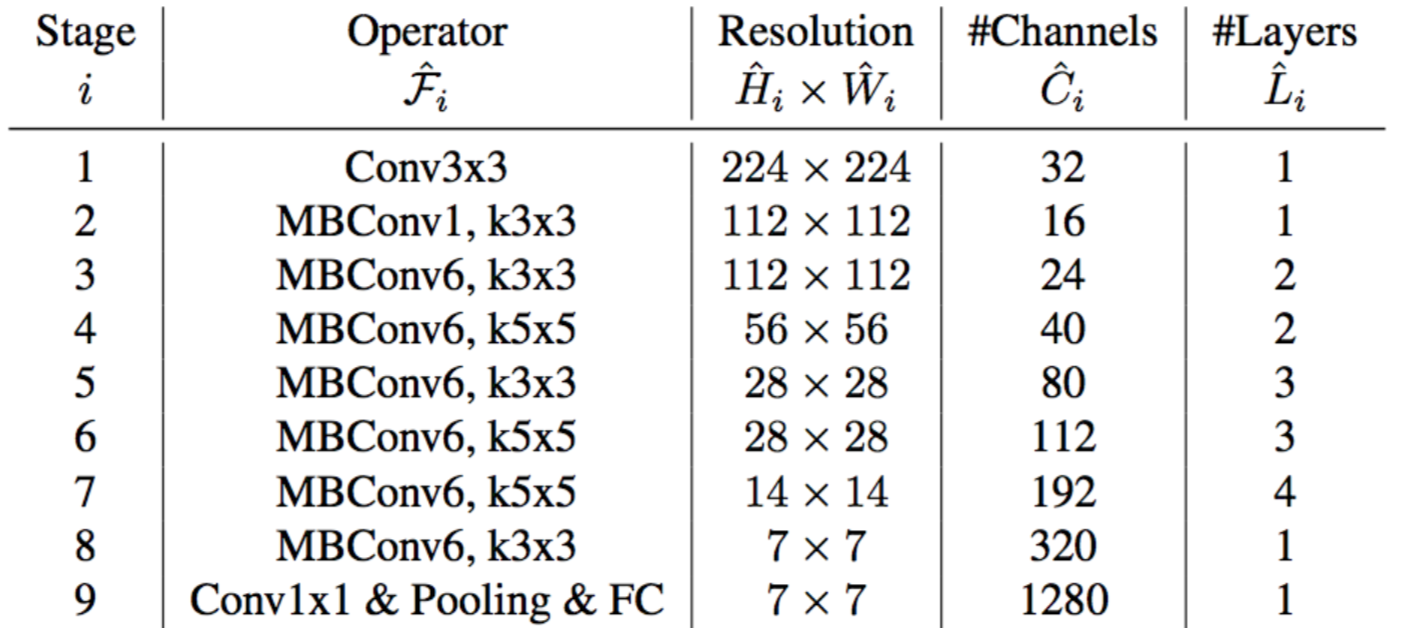
***It is critical to balance all dimensions of a network (width, depth, and resolution) during CNNs scaling for getting improved accuracy and efficiency.***

## Proposed Compound Scaling:

The authors proposed a simple yet very effective scaling technique which uses a **compound coefficient ɸ** to uniformly scale network width, depth, and resolution in a principled way:



EfficientNet-B0 Architecture: Scaling doesn’t change the layer operations, hence it is better to first have a good baseline network and then scale it along different dimensions using the proposed compound scaling.



The **MBConv** block is nothing fancy but an **Inverted Residual Block (used in MobileNetV2)** with a Squeeze and Excite block injected sometimes.