Transfer Learning

Utilizing Pre-trained Model for Better Results

Dr. Muhammad Sajjad

R.A: Kaleem Ullah

R.A: Imran Nawar

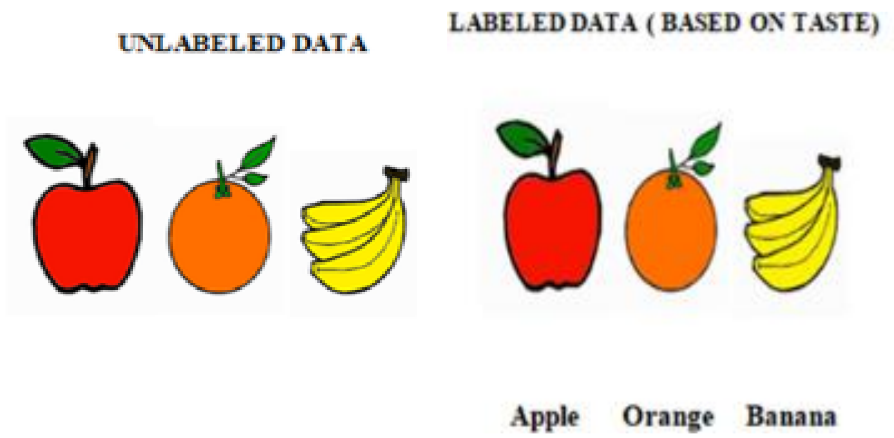
Overview

- **Challenges in training custom Deep Learning models**
- **Transfer Learning**
- **How Transfer Learning Works**
- **Kinds of Transfer Learning**
- **Pre-trained Models**
- **ImageNet**
- **ImageNet Competition**
- **AlexNet Architecture**
- **LeNet5**
- **VGG16/19**
- **Problems with Very Deep Networks**
- **RESNET**
- **INCEPTON**
- **1x1 Convolution**
- **EFFICIENTNET**
- **MobileNet**
- **Applications of Transfer Learning**

Challenges in Training Custom Deep Learning Models

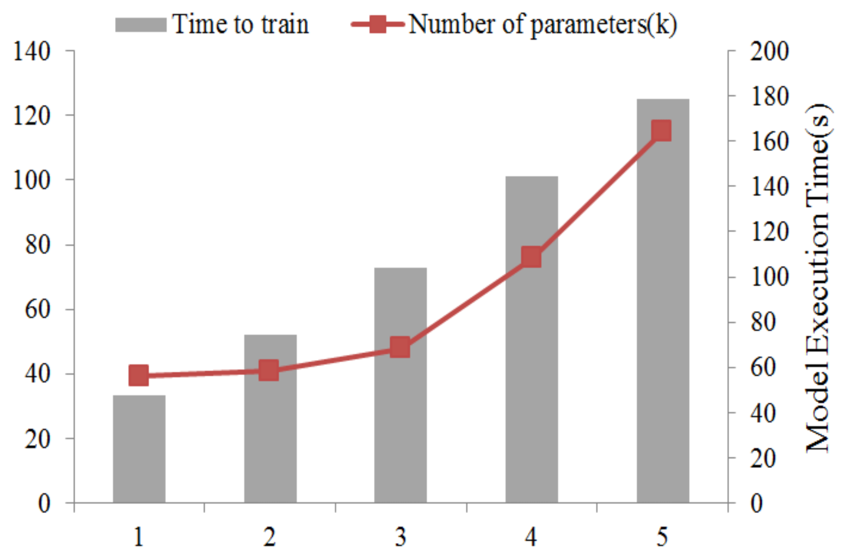
Data:

- Deep learning models typically require a large amount of labeled data to learn effectively.
- Gathering and labeling this data can be time-consuming and expensive.
- Without sufficient data, the model may not generalize well to new, unseen examples, leading to poor performance.



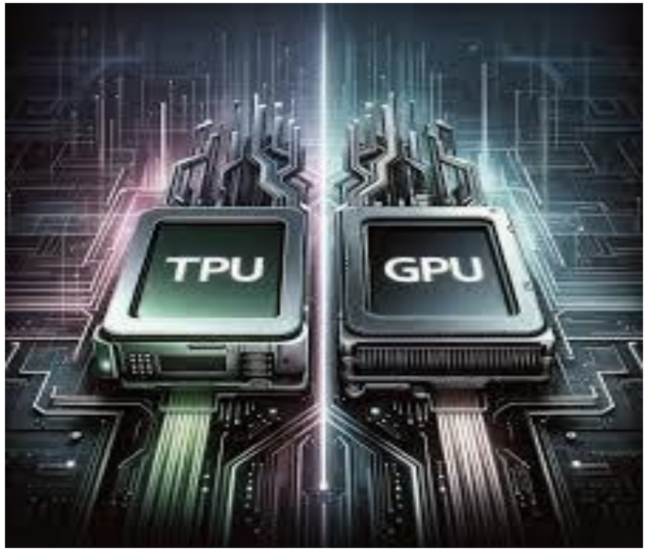
Training Time:

- Training deep learning models can be computationally intensive and time-consuming.
- Depending on the complexity of the model architecture, size of the dataset, and available computational resources, training can take days, weeks, or even longer.
- Longer training times also increase the cost associated with experimentation and model development.



Challenges in Training Custom Deep Learning Models

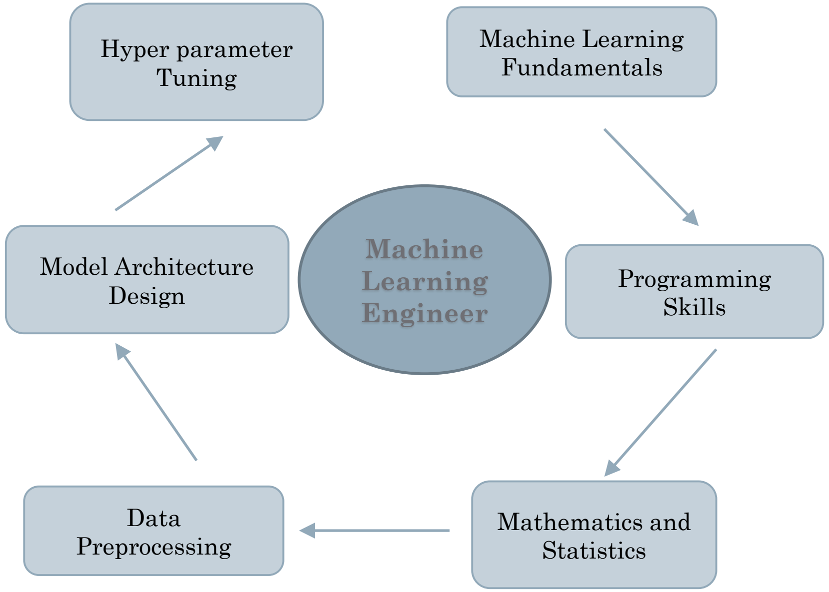
Computational Resources:



- Training deep learning models often requires significant computational resources, including powerful GPUs or even specialized hardware like TPUs.
- Not everyone has access to these resources, limiting the ability to train complex models effectively.

Expertise Requirement:

- Building and training custom deep learning models requires expertise in machine learning, deep learning, and software engineering.
- This expertise may not be readily available to everyone, especially those new to the field.



Solution

Transfer Learning

- A powerful technique in deep learning that allows us to reuse knowledge gained from solving one problem to tackle a different but related problem.
- Instead of training a model from scratch, we start with a pre-trained model and fine-tune it for the new task.
- It will not only speed up training considerably, but also requires significantly less training data.

Need of Transfer Learning?

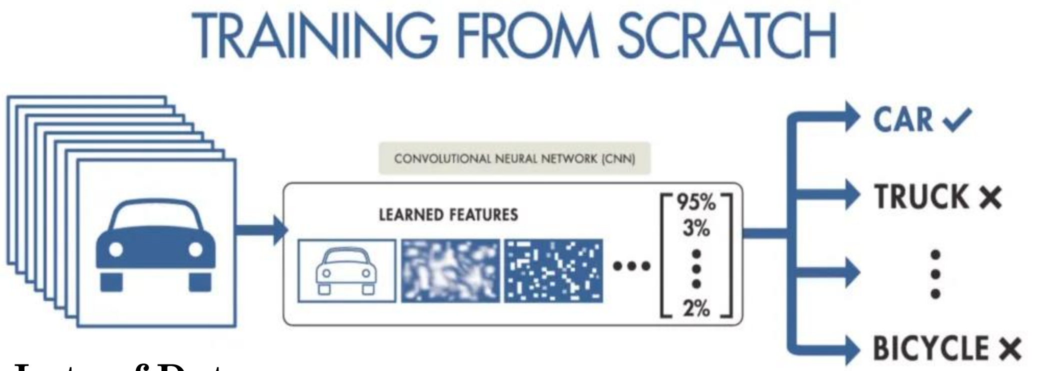
- If our dataset is really small
- Low Computation Power
- If our dataset is similar to pre-trained data then we have to only fine tuning our model it would save lot of time.

Advantages:

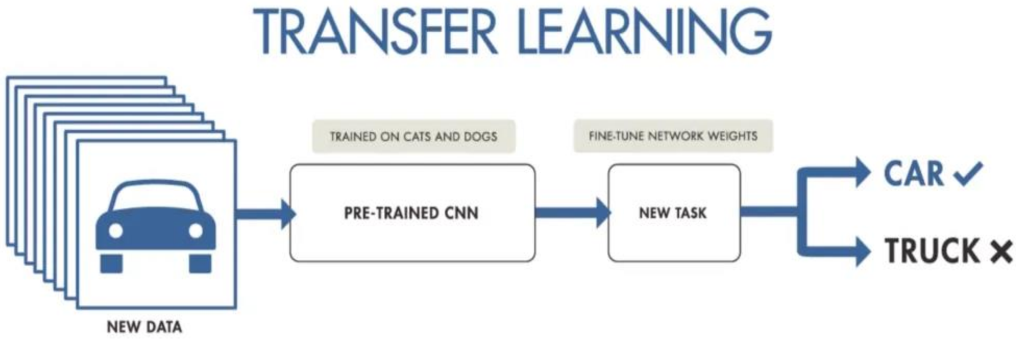
- Work with Limited data
- Reduced training time
- Improved neural network performance(in most cases)

Limitations

- Dataset is completely different from pre-trained data.



Lots of Data



Medium amount of Data

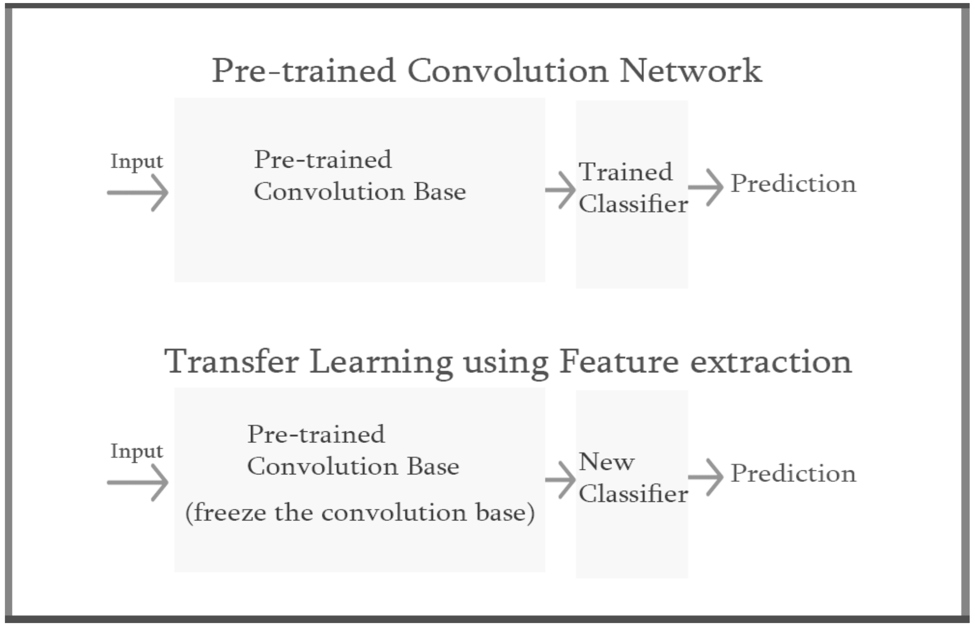
How Transfer Learning Works

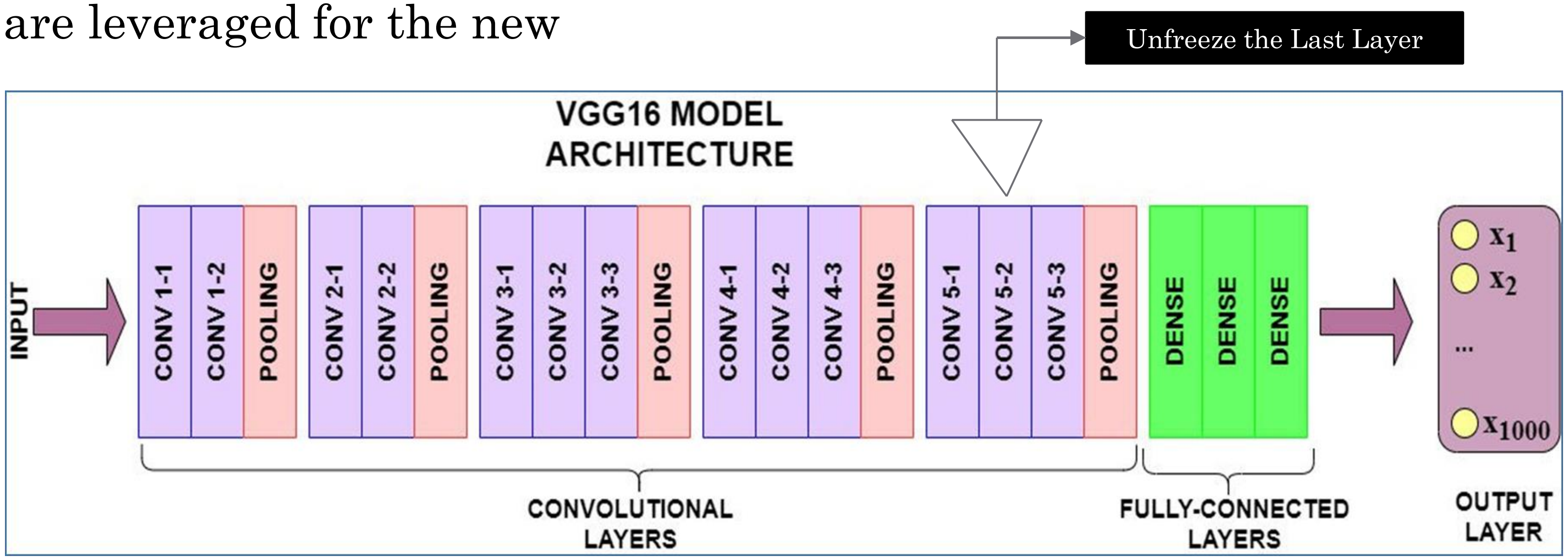
Pre-trained Model:

- Begin with a model that has already been trained on a large dataset for a specific task (e.g., image classification using ImageNet).
- This pre-trained model has learned useful features and patterns from the data.

Application to a New Task:

- Add new task-specific layers (e.g., an output layer) on top of the base model.
- Fine-tune the entire model on the new task using a smaller dataset.
- The base model's features are leveraged for the new task.





Kinds of Transfer Learning

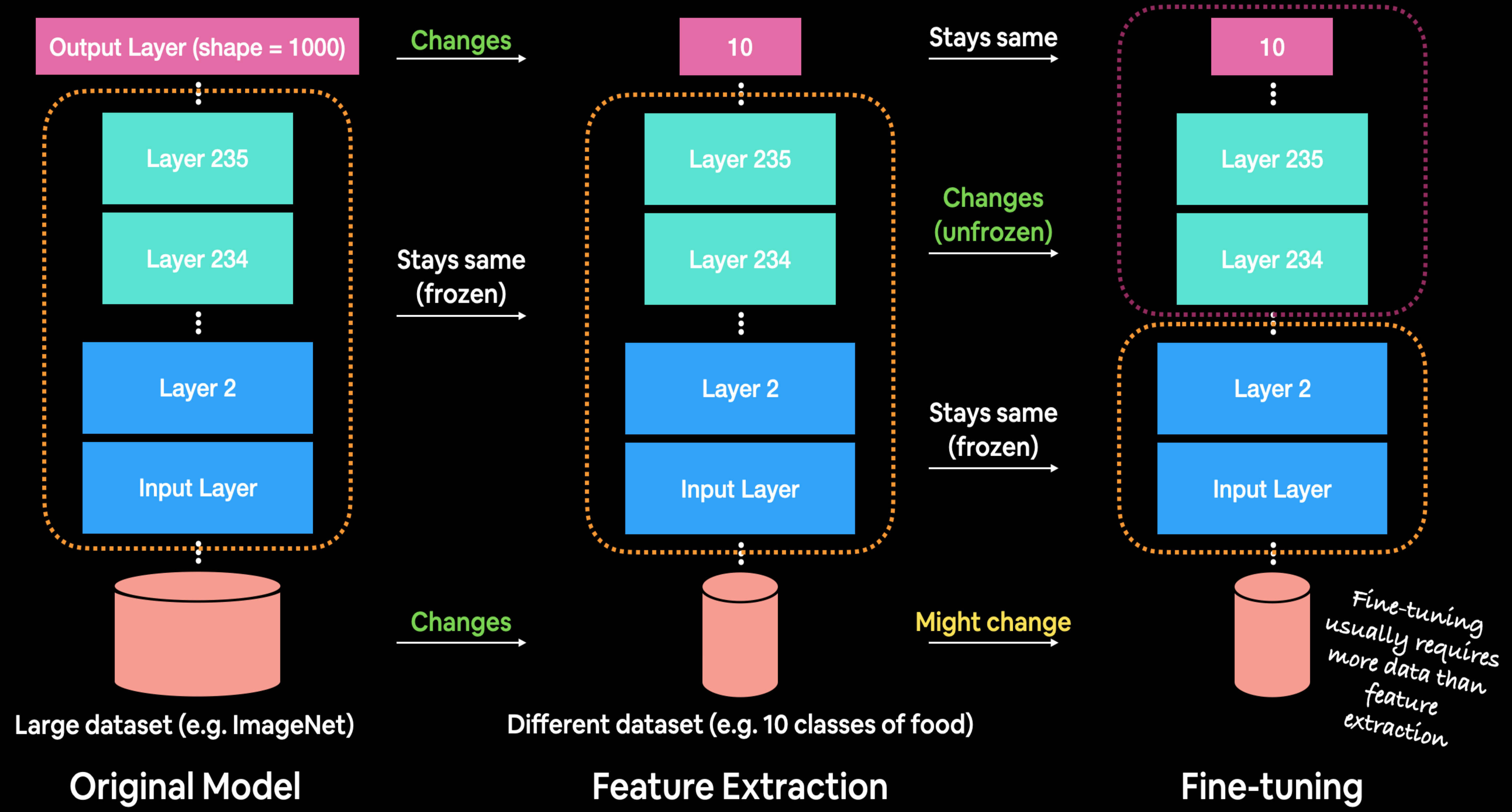
Feature Extraction:

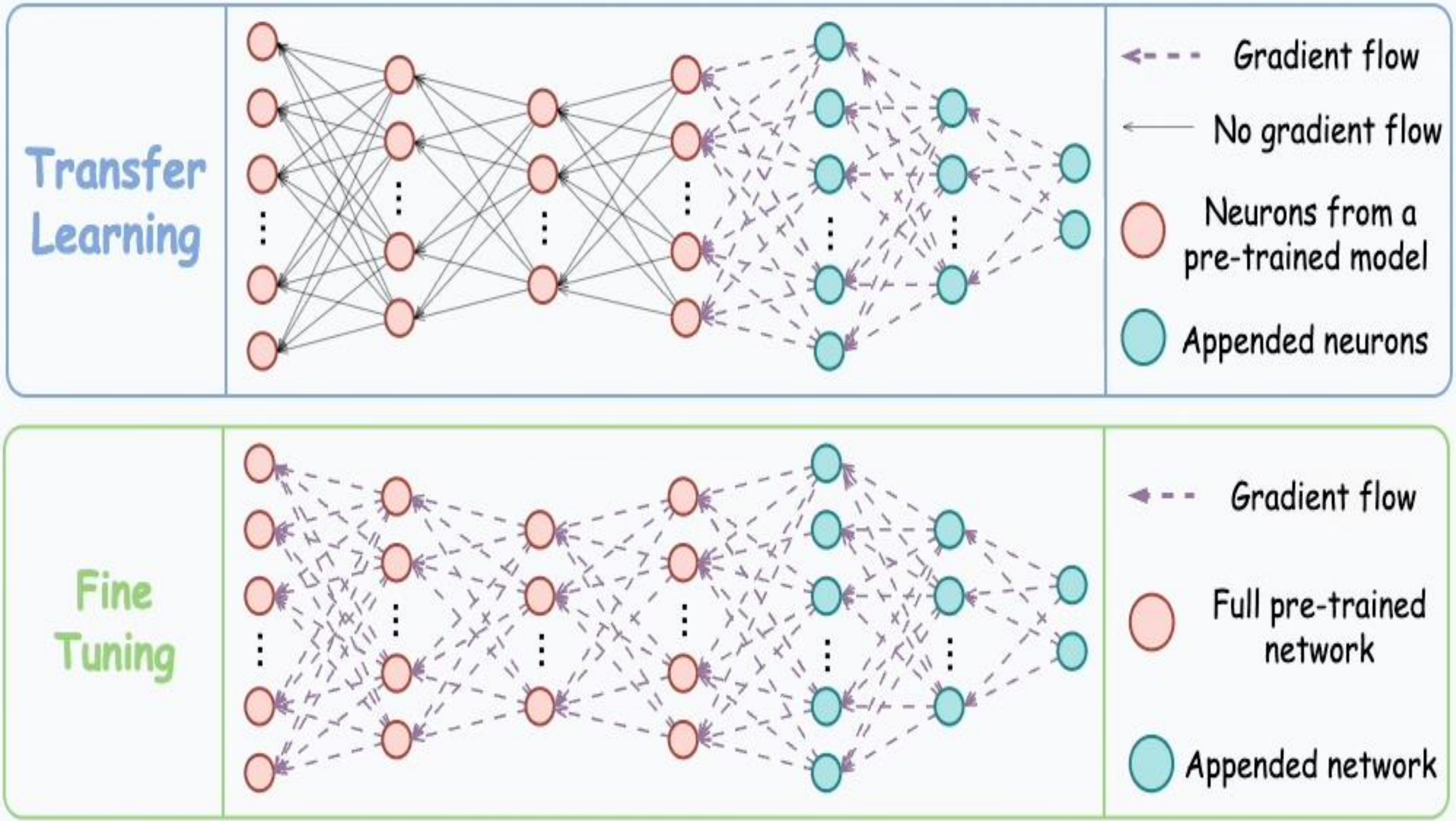
- Use the base model as a fixed feature extractor.
- Extract features from intermediate layers (e.g., convolutional layers in a CNN).
- Feed these features to a new classifier (e.g., a fully connected layer).
- Commonly used when data for the new task is limited.

Fine-Tuning:

- Fine-tune layer in the base model for the new task.
- Adjust the weights of the base model using the new task's data.
- Useful when the new task is closely related to the original task.

Kinds of Transfer Learning

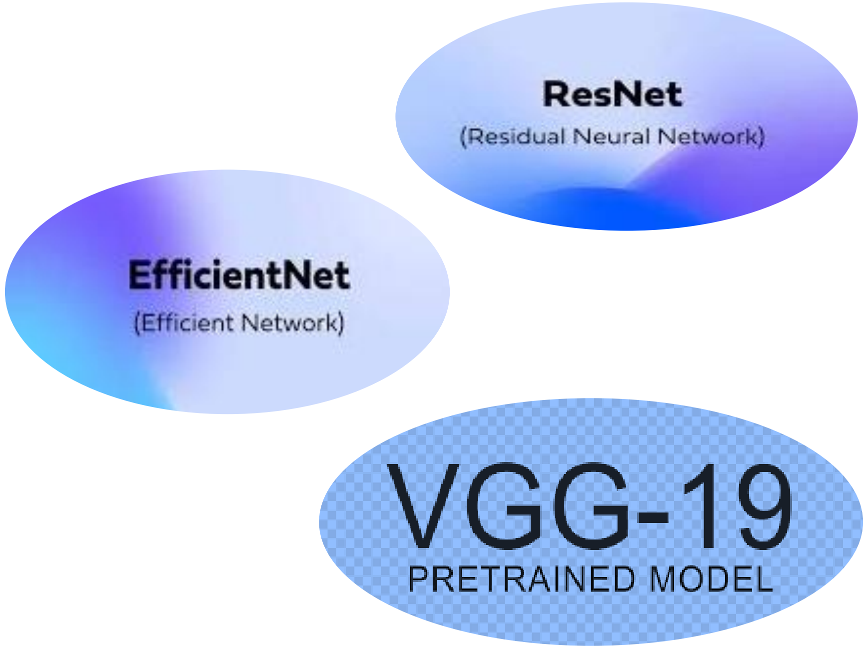



Transfer Learning
Gradient flow
No gradient flow
Neurons from a pre-trained model
Appended neurons
Fine Tuning
Gradient flow
Full pre-trained network
Appended network

Pre-trained Models

A **pre-trained model** is a deep learning model that has been trained on a large dataset and can be fine-tuned for a specific task.

- Pre-trained models serve as a starting point for developing deep learning models.
- They provide initial weights and biases that can be fine-tuned for specific tasks.
- Reuse lower layers of a pre-trained model for feature extraction, training only the final layers specific to your project.



Benefits:

- Save time and resources.
- Achieve higher accuracy with pre-learned features.
- Access models trained on large datasets (e.g., ImageNet with 14 million images).

Efficiency

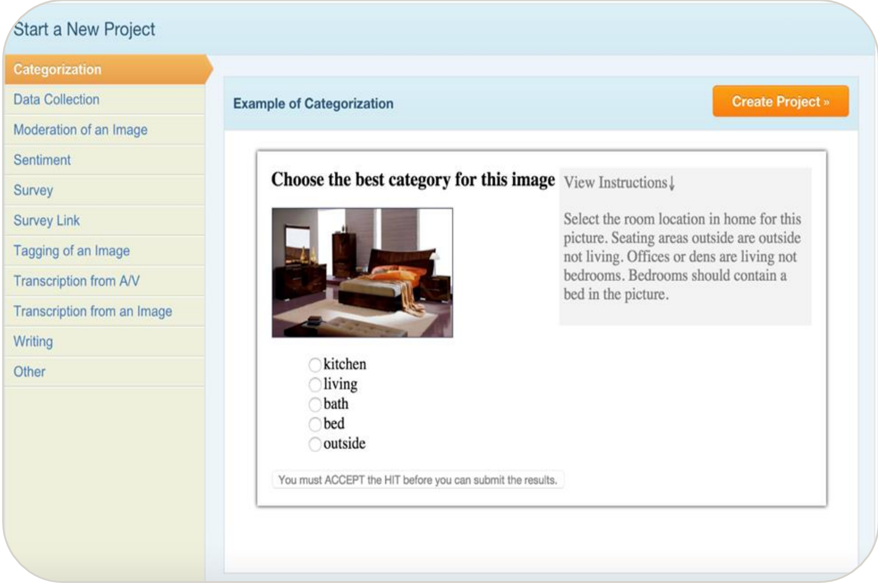
Performance

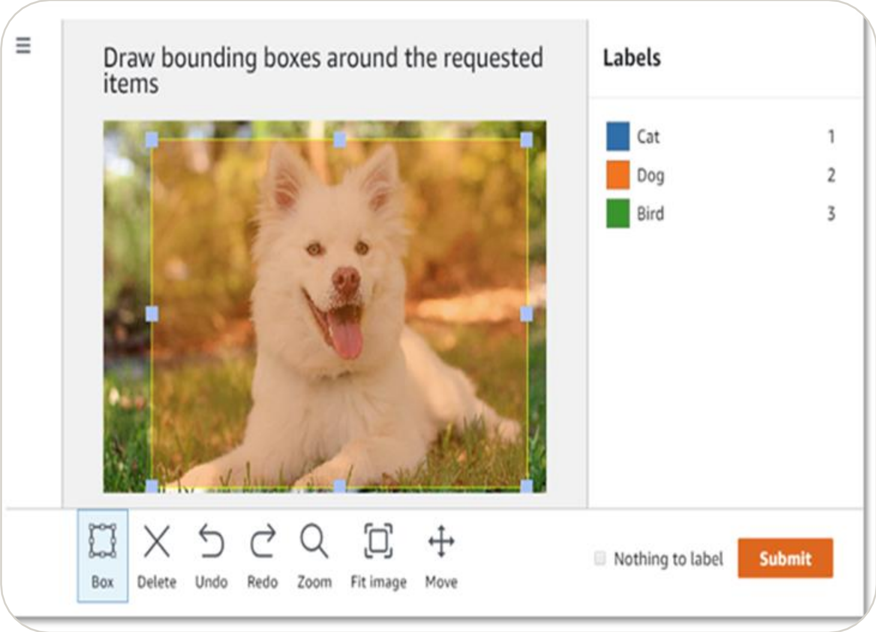
Accessibility

ImageNet

- The ImageNet dataset contained over **14 million** labeled images.
- The dataset contains more than 20,000 categories.
- A prominent computer scientist, **Fei-Fei Li,** co-founded the ImageNet project in 2009 (initiated in 2006) along with other researchers. Their work on ImageNet significantly advanced computer vision research and deep learning development.
- There are **1 million** Images with bounding box labelling as well for the purpose of **Object Localization** task, where the goal is to identify not only the object but also its precise location within the image.

- They used Amazon Mechanical Turk to help with the classification of images.
- Amazon Mechanical Turk (MTurk) is great for crowdsourcing tasks using images.





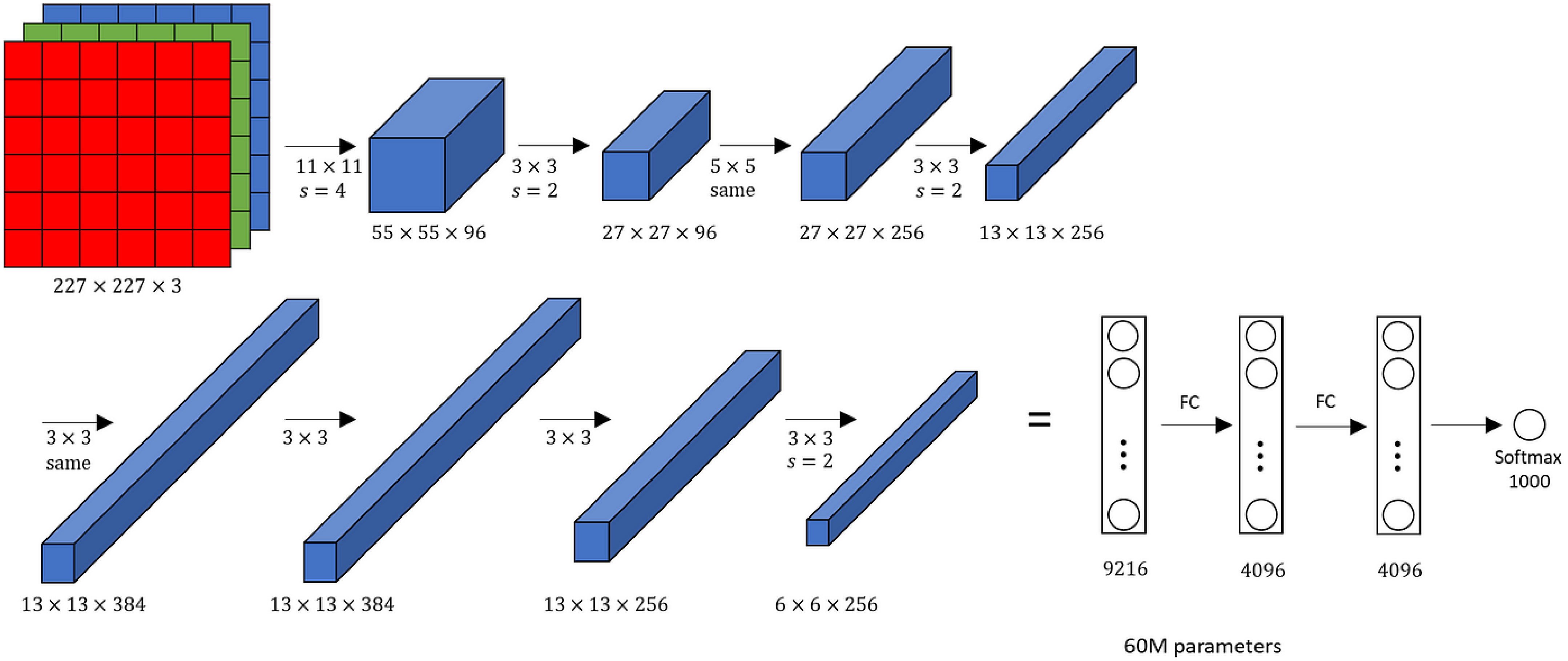
ImageNet Competition

- The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) starts in 2010 , and the goal was to Highlight the best model for classification to the research Community.
- The dataset which were used in the challenge was a **subset** of ImageNet which consists of around **1.2 million** images from **1000 classes.**
- At first The peoples were use **ML Algorithms**.
- The Error rate was **28%** for the first time.
- In **2011** the error rate reduces to **25%** using ML algorithms.
- The Revolution in **2012** when **Geoffrey Hinton** Participated in this challenge with his CNN based Model **AlexNet.**
- It's effective implementation of deep learning algorithm, ReLU as an activation function and as well as uses GPU instead of CPU, significantly improved performance.
- AlexNet achieved a significantly lower error rate in the ILSVRC 2012 competition compared to previous approaches.

Winning Models of the ImageNet Competition

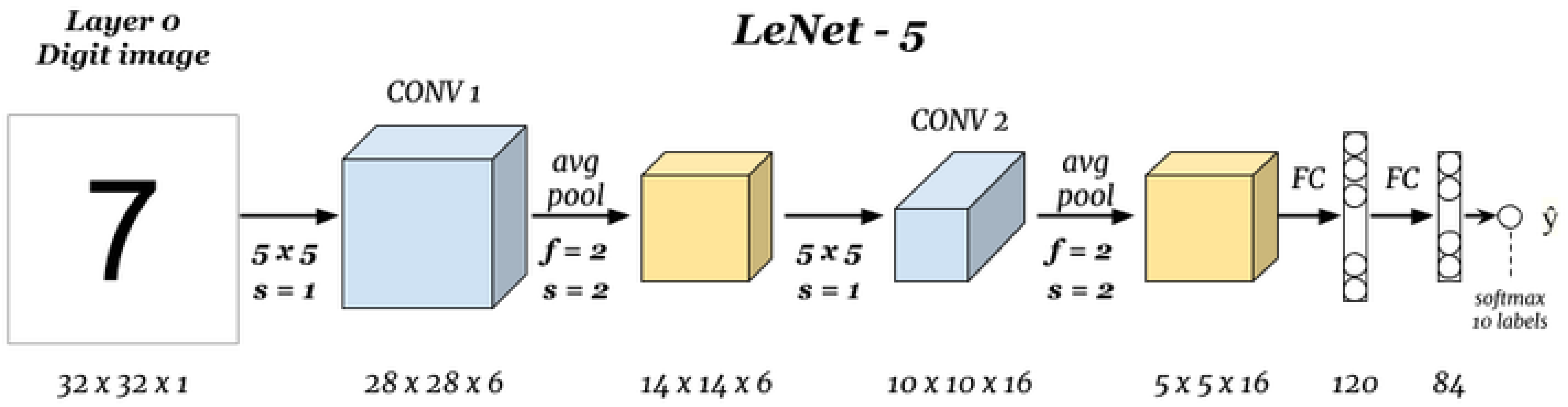
YEAR	WINNER	TOP 5 ERROR RATE %
2012	ALEXNET	15.3
2013	ZFNET	11.2
2014	INCEPTION V1 (GoogLeNet) VGG NET (Runner up)	6.67 7.3
2015	ResNet	3.57
2016	ResNeXt	4.1
2017	SENet	2.251

AlexNet Architecture



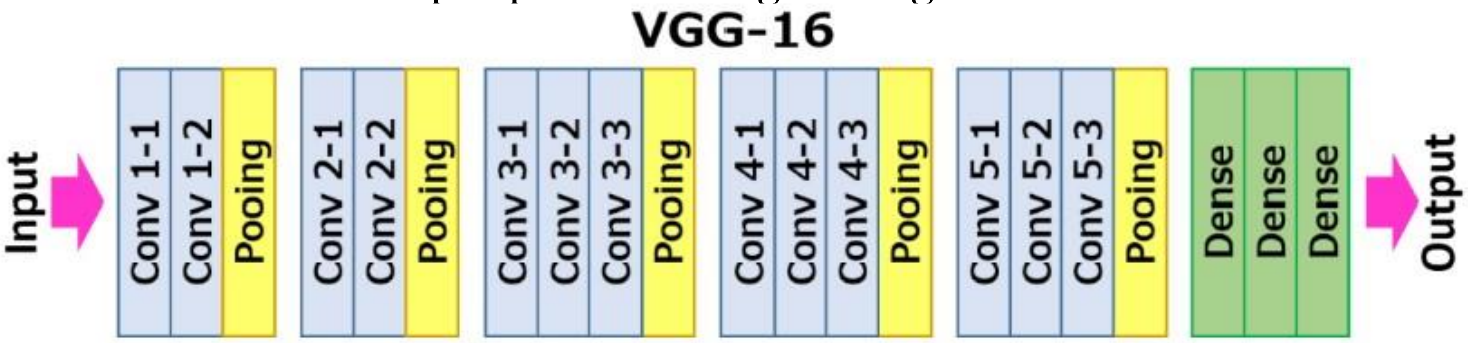
LeNet5 Architecture

- LeNet-5 is the earliest CNN architecture, developed by Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner in 1998.
- It was primarily designed for handwritten digit recognition tasks, particularly recognizing digits in postal codes on letters

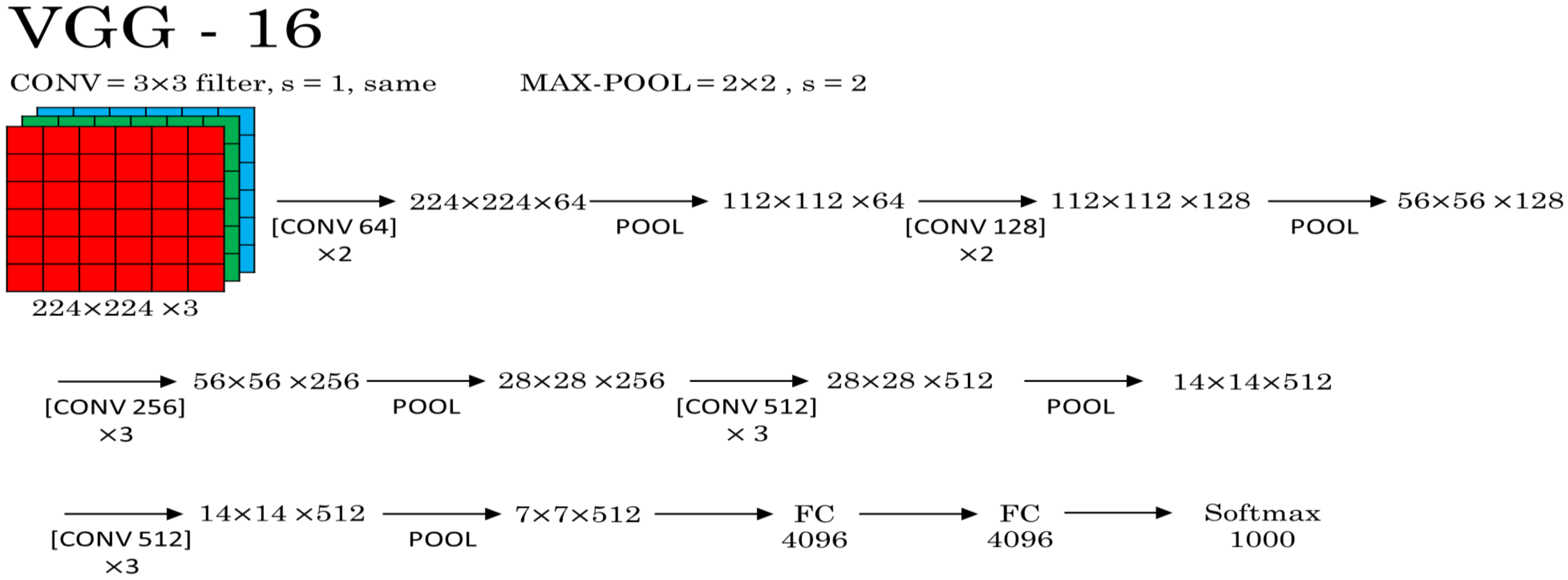


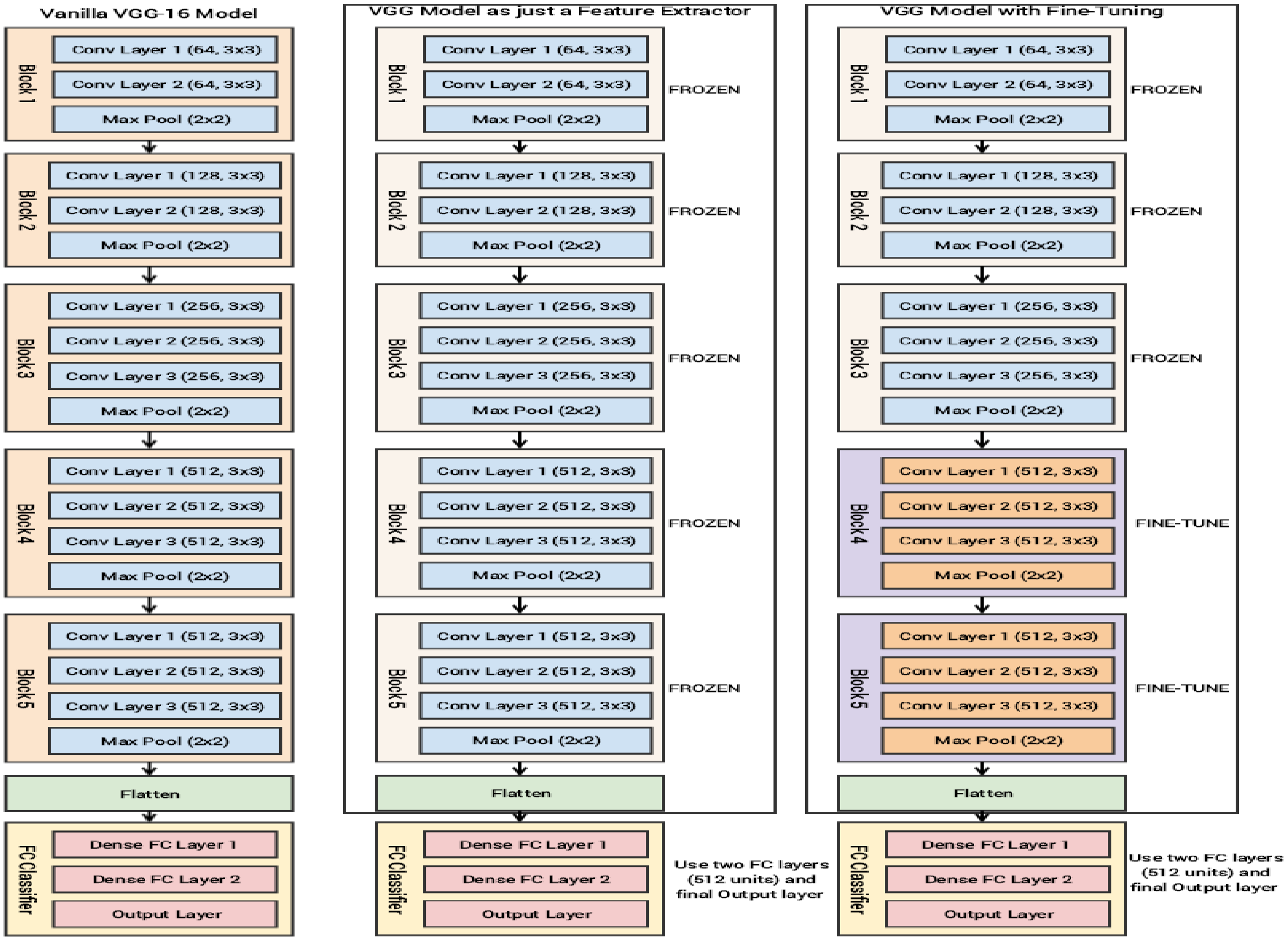
VGG16/19 Architecture

- VGG-16 model, created by the Visual Geometry Group at the University of Oxford.
- The VGG-16 model is a 16-layer (convolution and fully connected) network built on the ImageNet database, which is built for the purpose of image recognition and classification.



VGG16



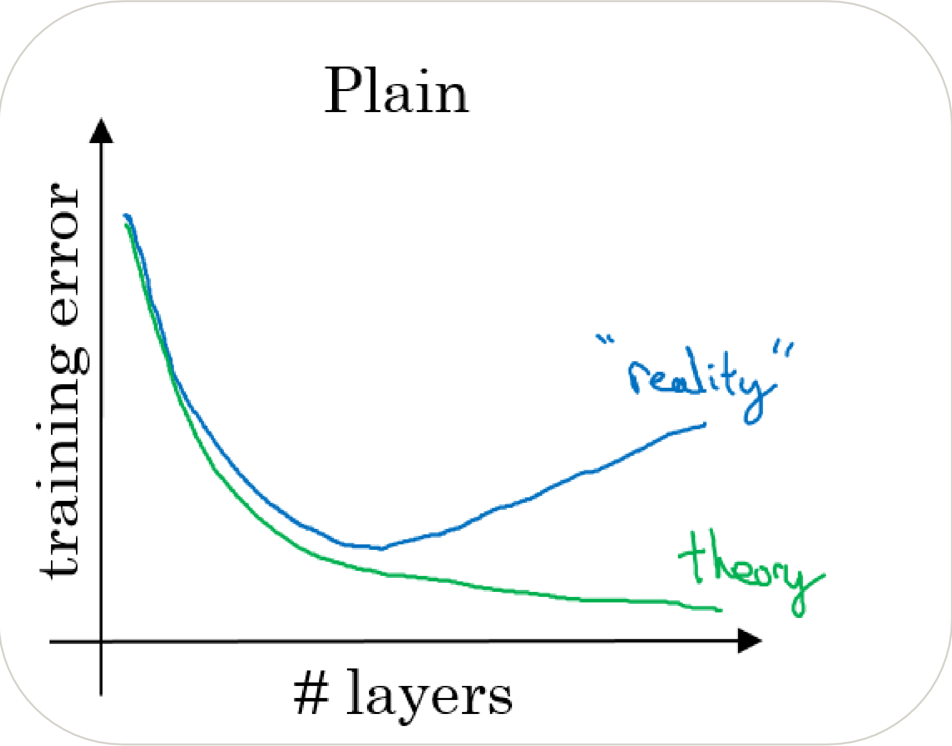

Vanilla VGG-16 Model
Block 1
Conv Layer 1 (64, 3x3)
Conv Layer 2 (64, 3x3)
Max Pool (2x2)
Block 2
Conv Layer 1 (128, 3x3)
Conv Layer 2 (128, 3x3)
Max Pool (2x2)
Block 3
Conv Layer 1 (256, 3x3)
Conv Layer 2 (256, 3x3)
Conv Layer 3 (256, 3x3)
Max Pool (2x2)
Block 4
Conv Layer 1 (512, 3x3)
Conv Layer 2 (512, 3x3)
Conv Layer 3 (512, 3x3)
Max Pool (2x2)
Block 5
Conv Layer 1 (512, 3x3)
Conv Layer 2 (512, 3x3)
Conv Layer 3 (512, 3x3)
Max Pool (2x2)
Flatten
FC Classifier
Dense FC Layer 1
Dense FC Layer 2
Output Layer
VGG Model as just a Feature Extractor
FROZEN
Use two FC layers (512 units) and final Output layer
VGG Model with Fine-Tuning
FROZEN
FINE-TUNE
Use two FC layers (512 units) and final Output layer

Problems with Very Deep Networks

- **Initial Belief:**
  - Increasing the number of layers in a neural network would consistently improve accuracy.
- **Practical Issues:**
  - No generalization
  - Vanishing gradient
  - Exploding gradients
  - Hampered training and limited performance

Vanishing Gradient

Exploding Gradients



Vanishing/Exploding Gradients Problems:

In deep neural networks, during backpropagation, gradients can become extremely small.
This happens as they are propagated back through the network.

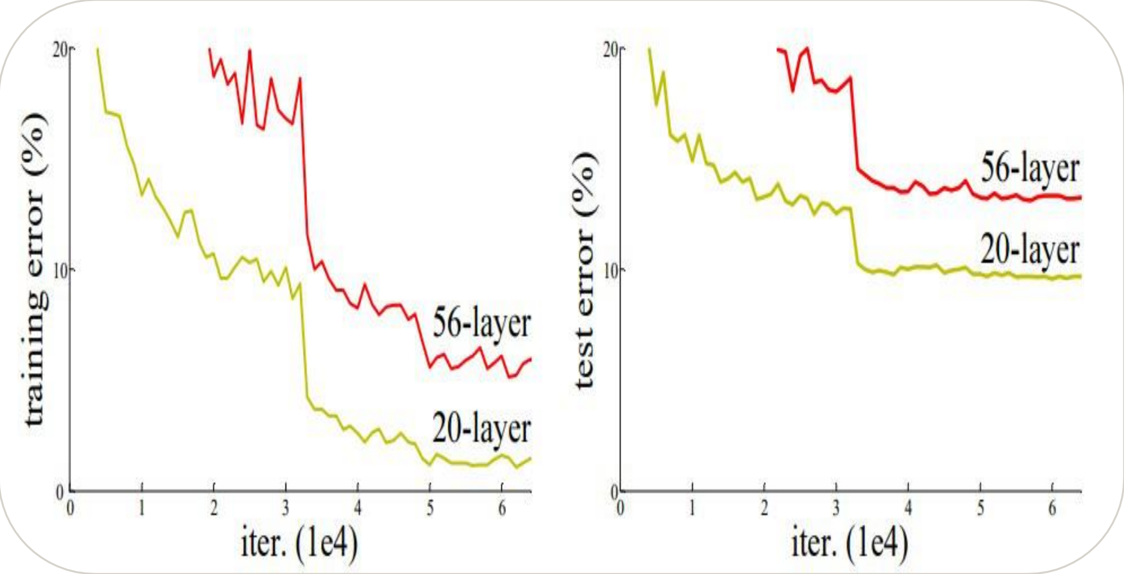
Vanishing Gradient Problem:

- During backpropagation, gradients can become extremely small.
- Lower layers' weights remain unchanged, hindering convergence..

Exploding Gradients Problem:

- During backpropagation, Gradients grow larger, leading to unstable training.
- Results in excessively large weight updates.
- Often observed in recurrent neural networks.

It makes learning difficult, particularly in deeper layers.



Comparison of 20-layer vs 56-layer architecture

Solution

Residual Learning

Skip Connections:

- A shortcut connection in a neural network that bypasses some layers and directly feeds the output of an earlier layer to a later layer.
- Helps in bypassing any layer that may degrade performance, effectively regularizing the network.

Residual Blocks:

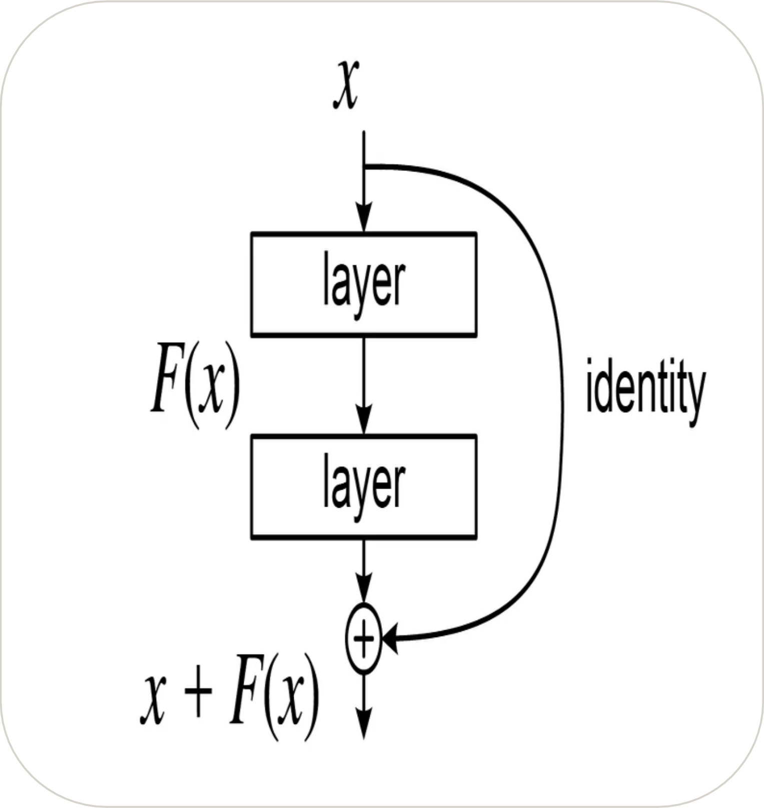
- Introduce skip connections to address vanishing/exploding gradient issues.
- Form the basic building blocks of ResNet.

Residual Mapping:

- Instead of learning the underlying mapping $H(x)$, the network learns the residual $F(x)$ where $F(x) := H(x) - x$
- Allows the network to fit the function $H(x) := F(x) + x$.

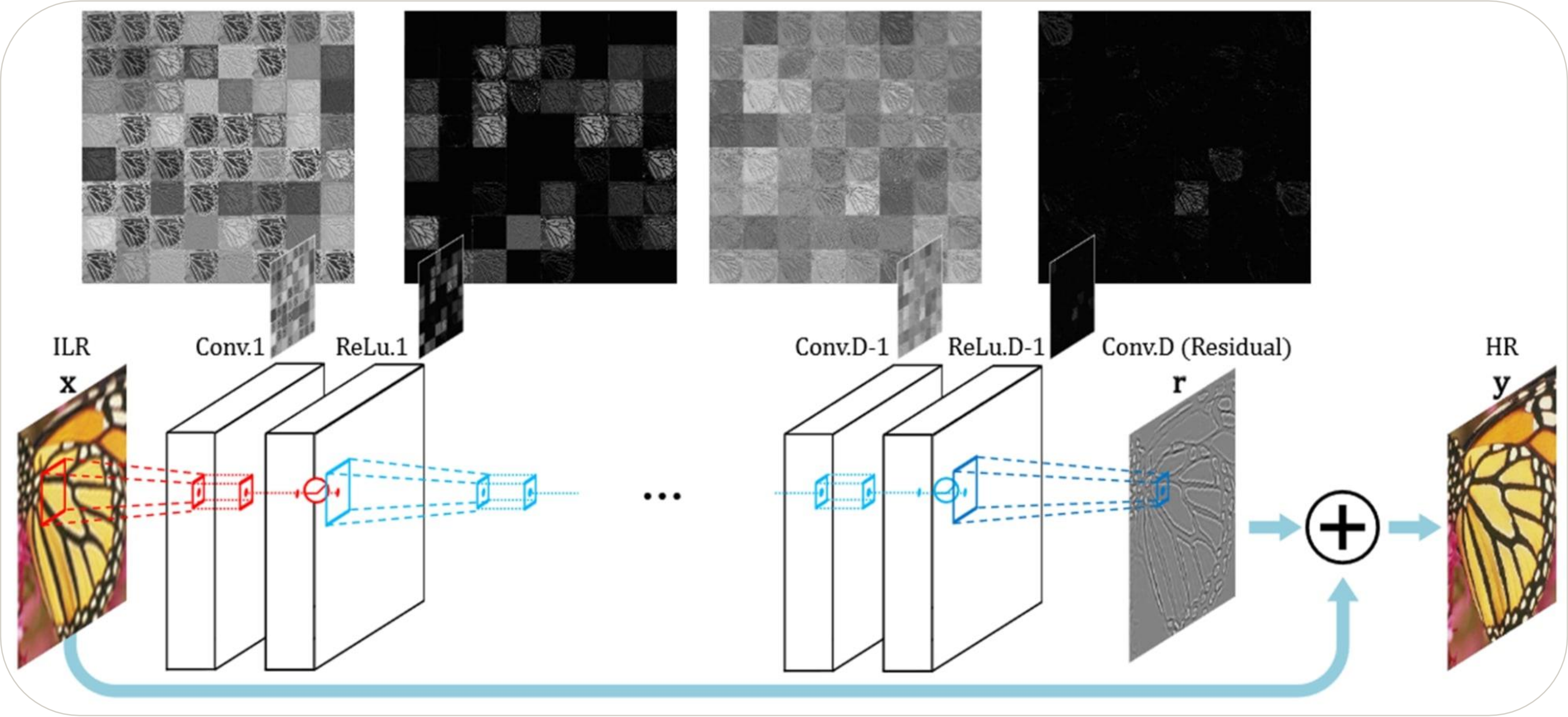
Advantages:

- Mitigates vanishing/exploding gradient problems.
- Enables training of very deep neural networks.
- Improves performance and generalization.



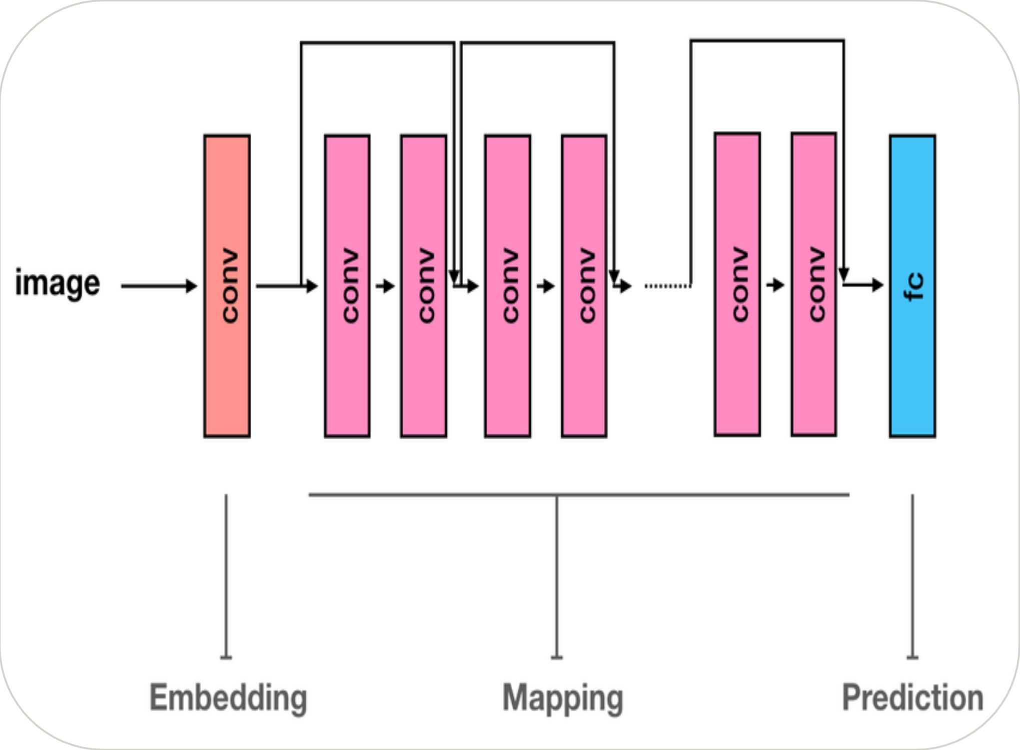
A Residual Block in a deep Residual Network. Here the Residual Connection skips two layers

Residual Learning



ResNet

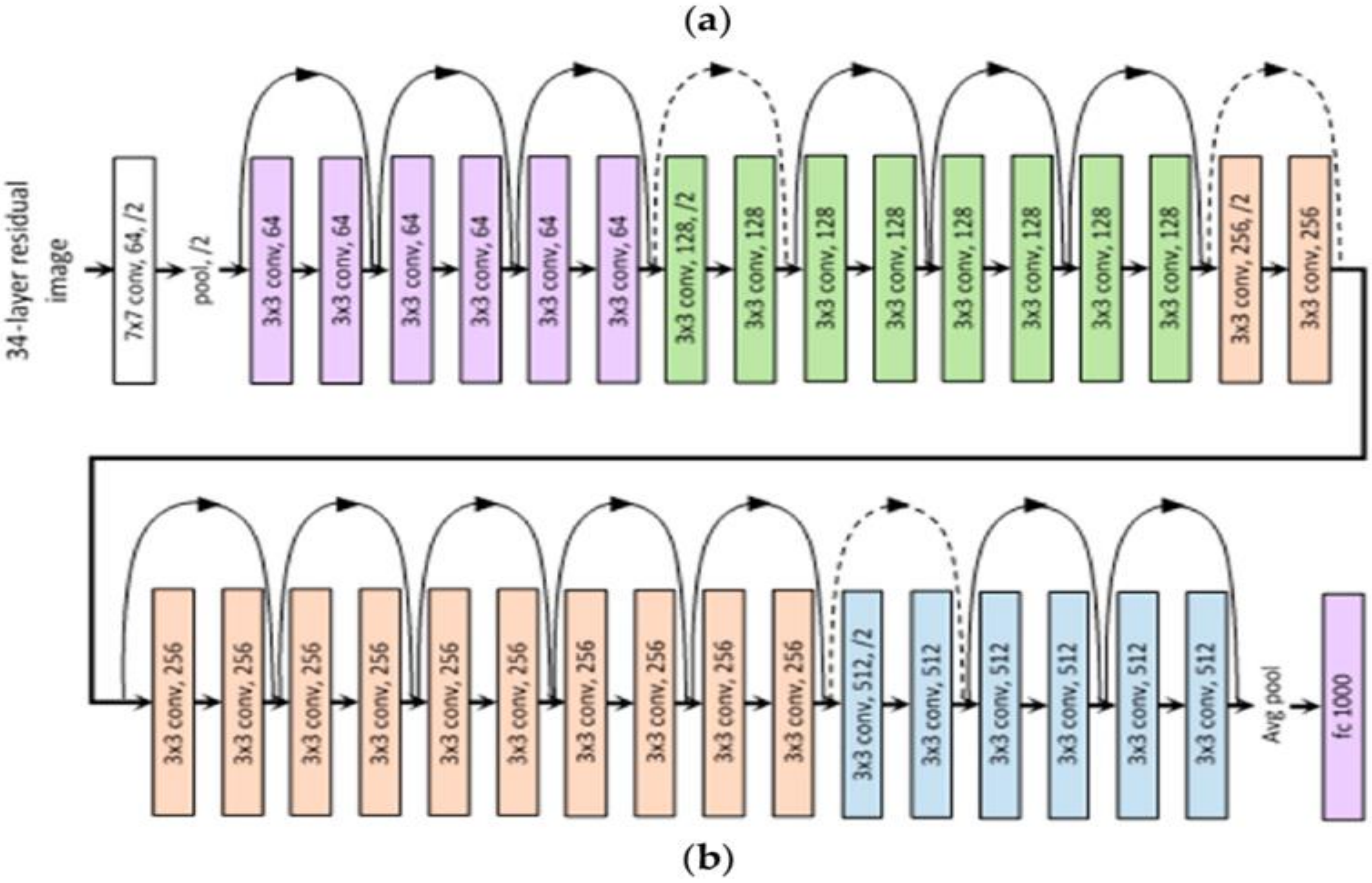
- ResNet was proposed in 2015 by researchers at Microsoft Research introduced a new architecture called Residual Network.
- A Residual Neural Network (ResNet) is an Artificial Neural Network (ANN) of a kind that stacks residual blocks on top of each other to form a network.
- ResNet first introduced the concept of skip connection.
- Winner of the ImageNet Challenge in 2015 with an error rate of 3.57%.
- ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer



Variants of ResNet architecture

- Resnet-18, Resnet-34, Resnet-50, Resnet-101, Resnet- 152. The number after all the model is the number of layers in the model.

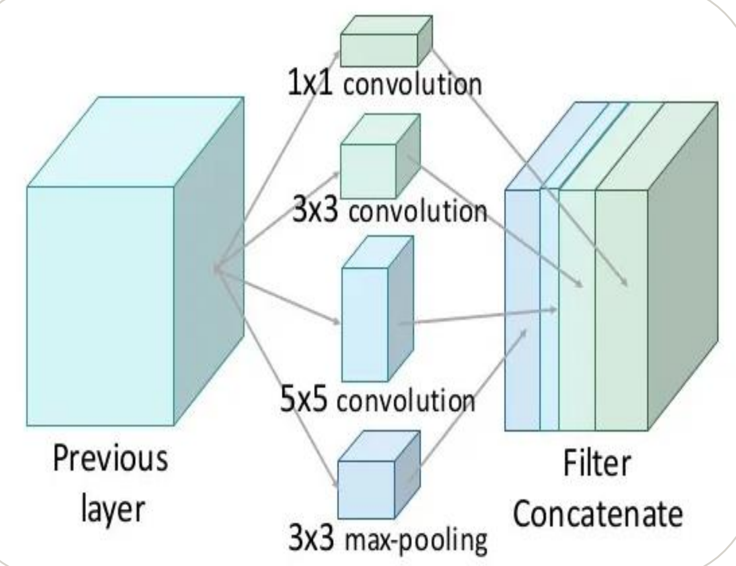
ResNet-34 Layered architecture

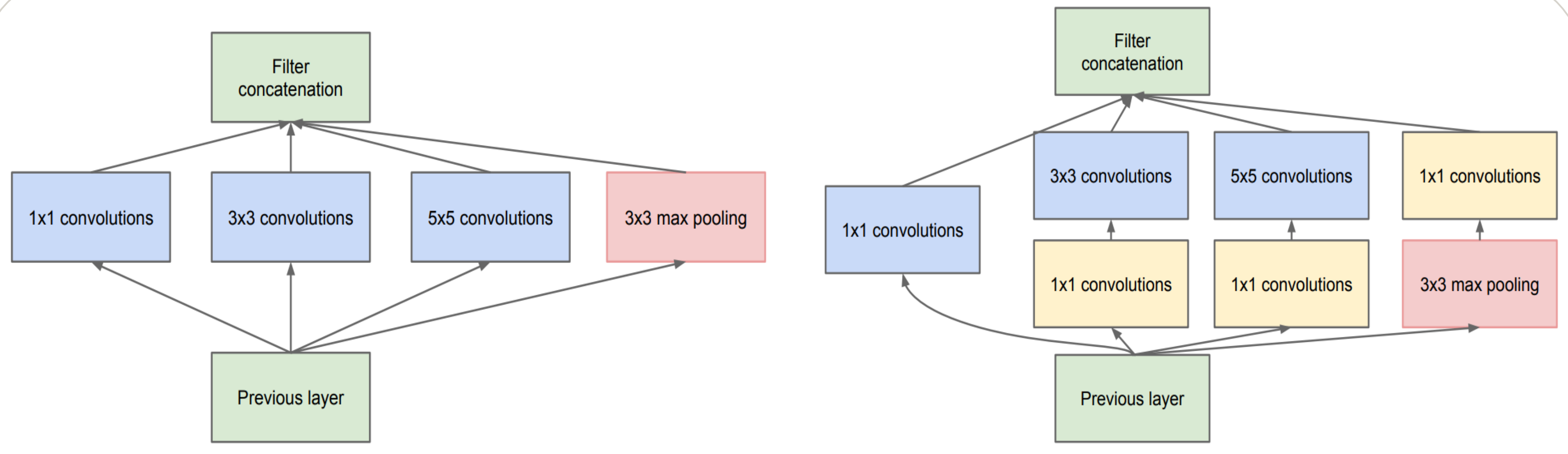


Inception

Inception Module:

- Utilizes multiple convolutional filters (1x1, 3x3, 5x5) and pooling operations within the same module.
- While some networks like VGG16 focus only on 3x3 or LeNet5 on 5x5, Inception makes sure to grab all kinds of features.
- By using various filter sizes, Inception can pick up both small and big details in the data.
- Captures information at different scale.





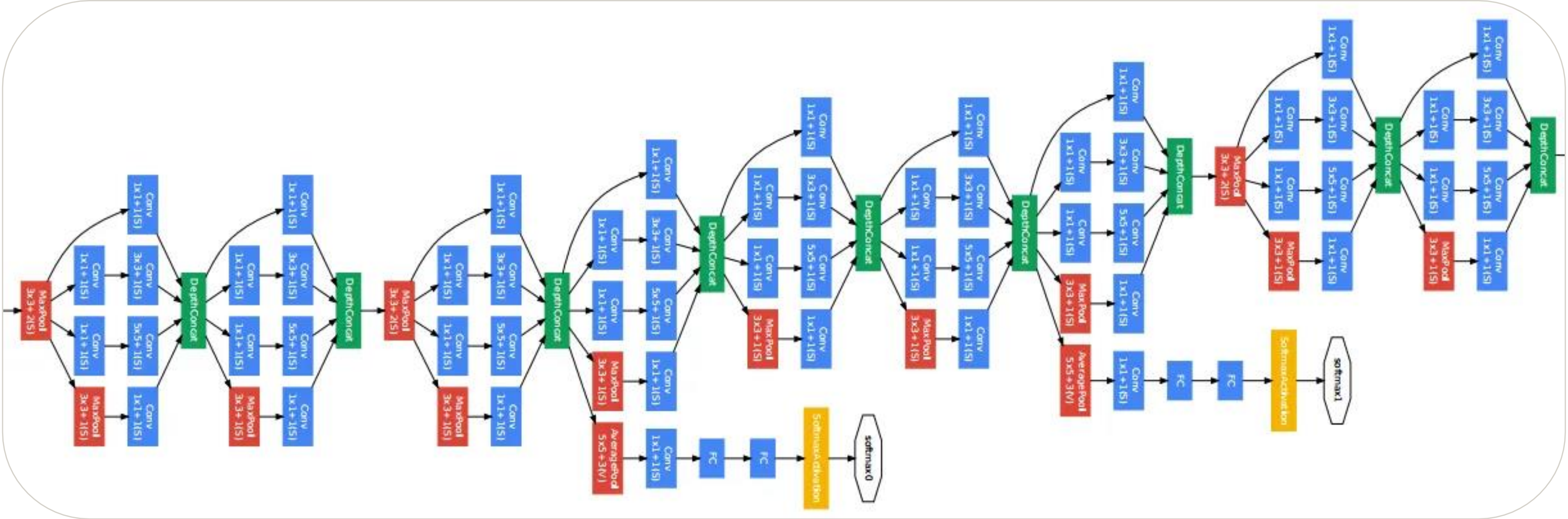
(a) Inception module, naïve version (b) Inception module with dimension reductions

Inception Pre-trained Models:

- Inception-v1 (GoogLeNet), Inception-v2, Inception-v3, Inception-v4.

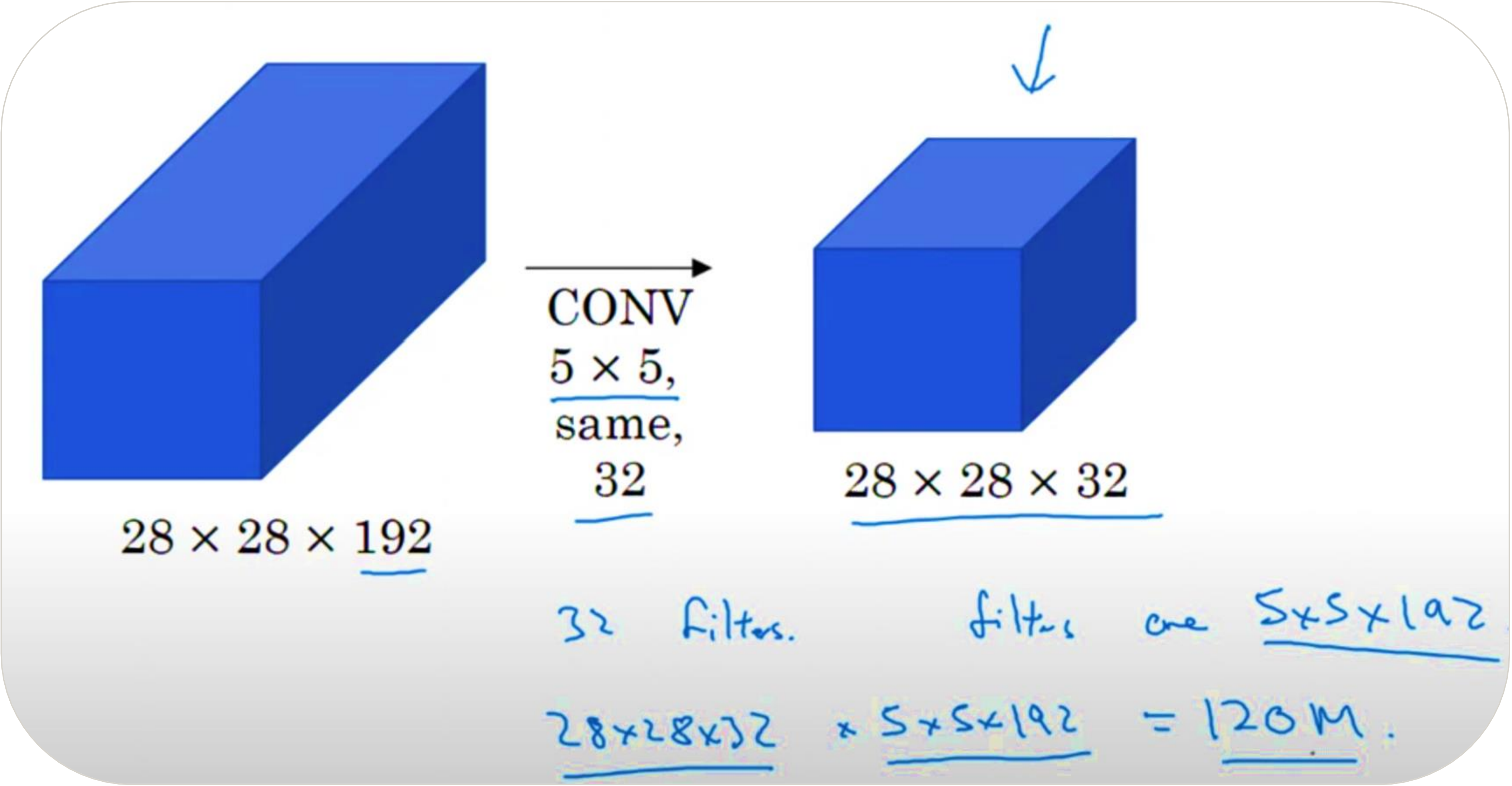
Notable Achievements:

- Inception-v1 (GoogLeNet) won the 2014 ImageNet Challenge with a top-5 error rate of 6.67%.



GoogLeNet, 2014

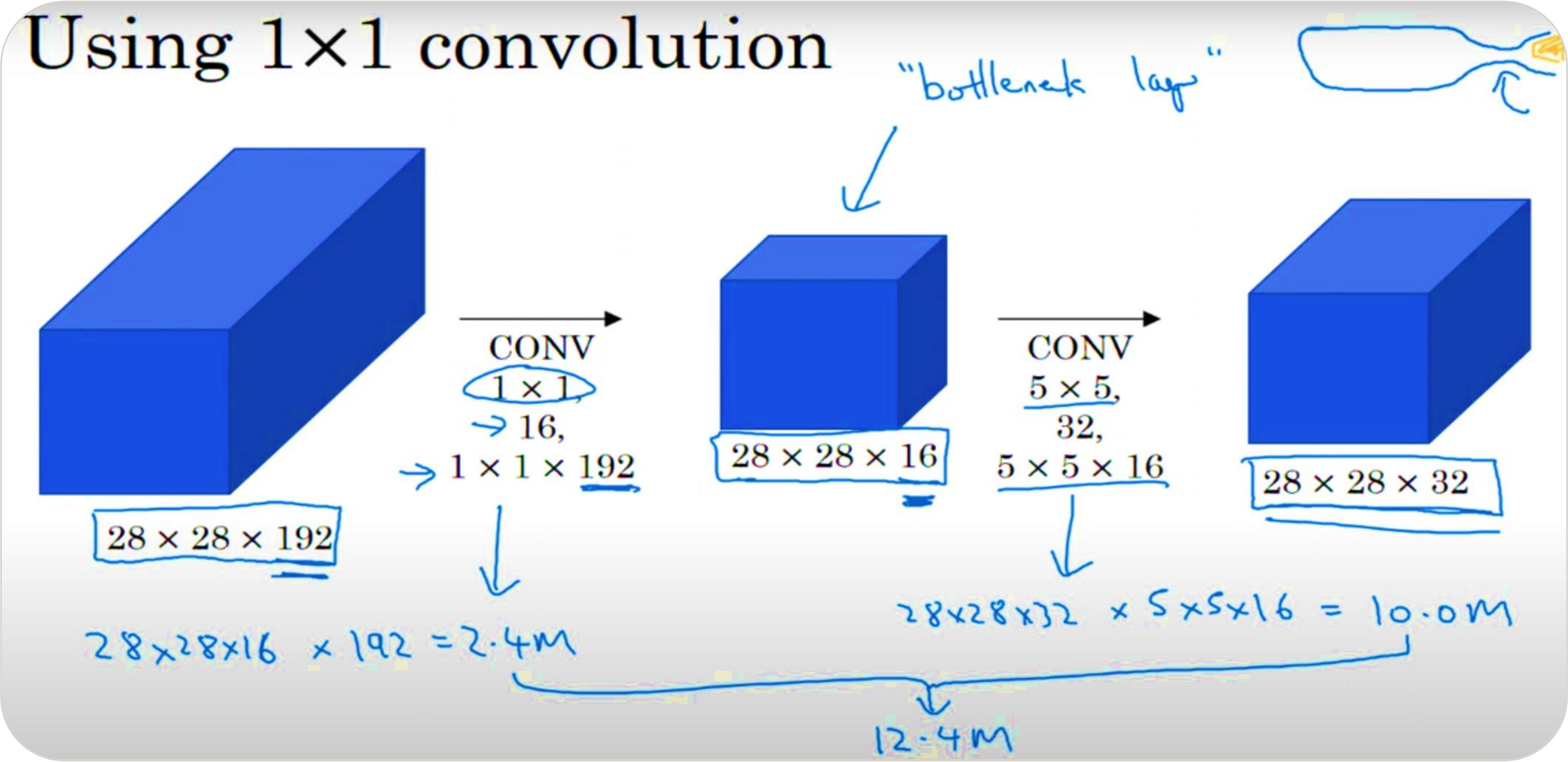
The Problem of Computational Cost



Solution

Less Parameters means Less Computational Cost.

- Add 1*1 Conv before 3*3
- Add 1*1 Conv before 5*5
- And Add 1*1 Conv after the 3*3 Maxpool layer.



1x1 Convolution

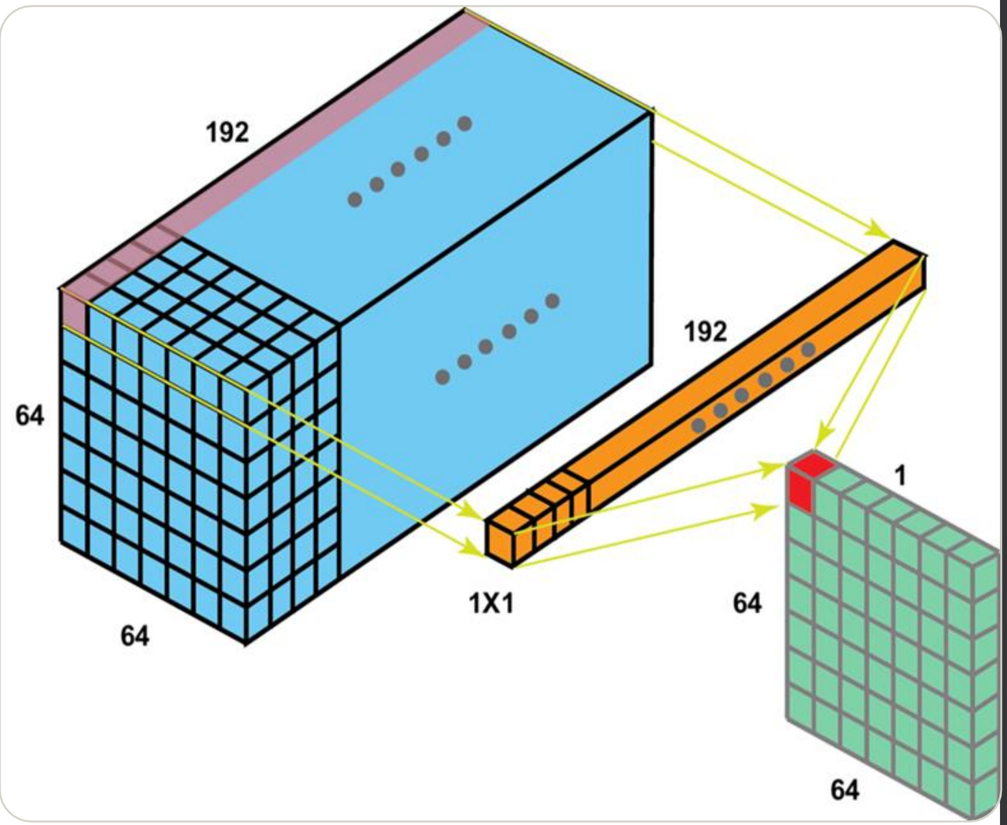
- A 1x1 convolution applies a single 1x1 filter to each pixel in the input volume.
- It processes each pixel individually but across all channels (depth), combining the information from different channels.

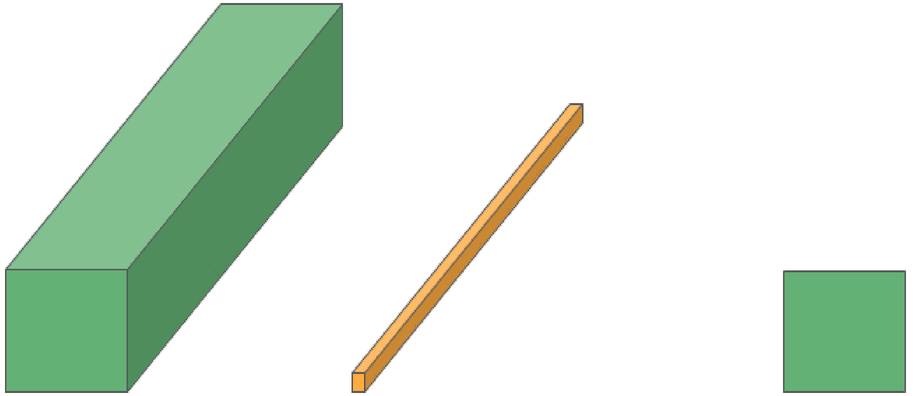
Purpose:

- Reduces the number of channels while retaining spatial dimensions.
- Enables efficient dimensionality reduction and computational cost savings.

Applications

- **ResNet:** Used in bottleneck blocks for efficiency.
- **MobileNet:** Part of depthwise separable convolutions.
- **Inception Modules:** Reduces dimensions before expensive convolutions.





EfficientNet

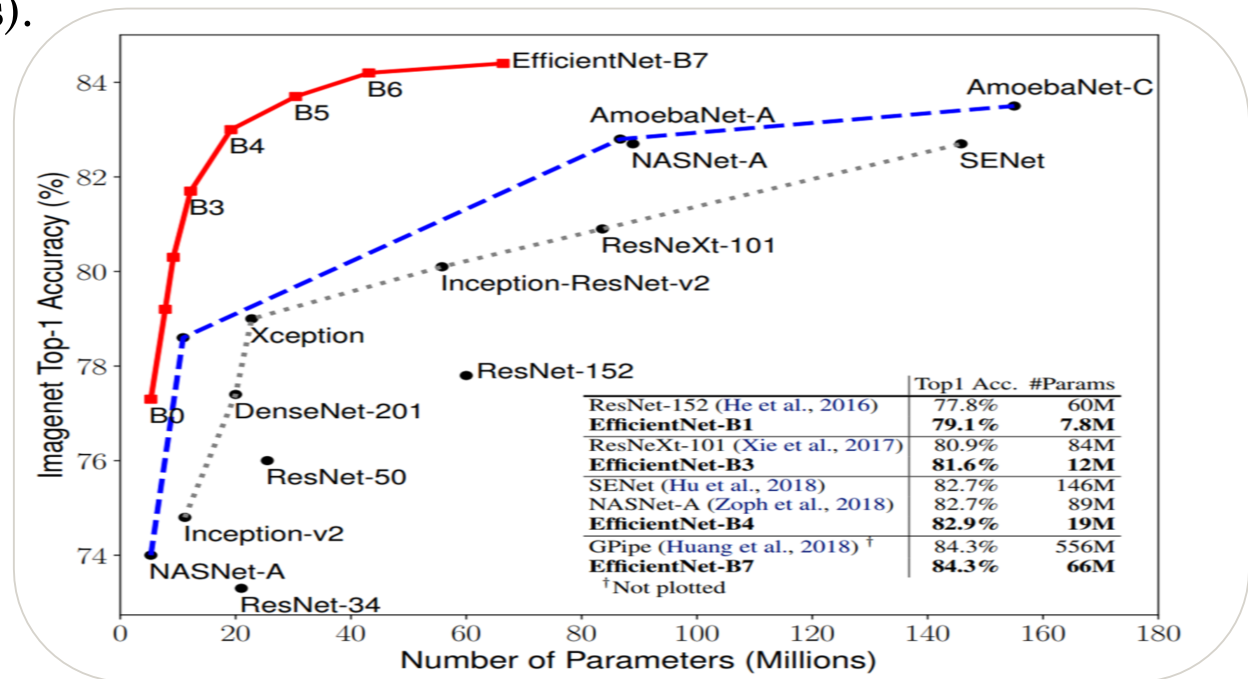
- Introduced in 2019 by a team of researchers at Google AI,
- The most powerful CNN architecture
- EfficientNet is built upon a concept called compound scaling.
- **Compound scaling** optimizes model depth, width, and resolution for optimal efficiency.

Applications:

- Image classification, object detection, semantic segmentation

EfficientNet Variants

- **EfficientNet B0-B7:** A family of EfficientNet models with varying complexities.
- **B0: Most lightweight** (5.3 million parameters).
- **B7: Most complex** (6.1 billion parameters).



MobileNet

Developed by Google researchers.

Purpose:

- Designed for mobile and embedded vision applications.
- Focuses on efficient, lightweight models suitable for devices with limited computational resources.

Key Features:

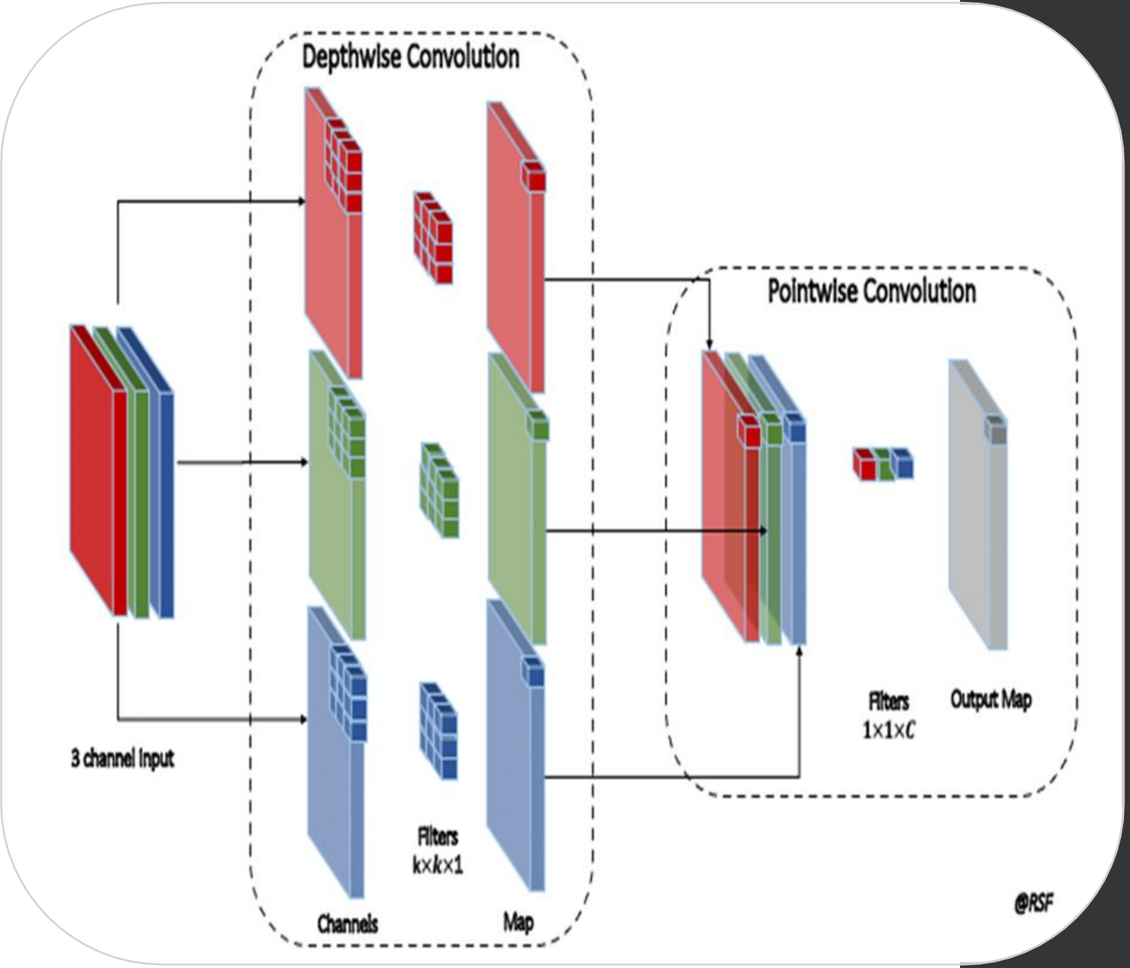
- Reduces computational cost and model size.
- Fewer parameters compared to traditional convolutional networks.
- Maintains competitive accuracy with optimized speed and efficiency.

Applications:

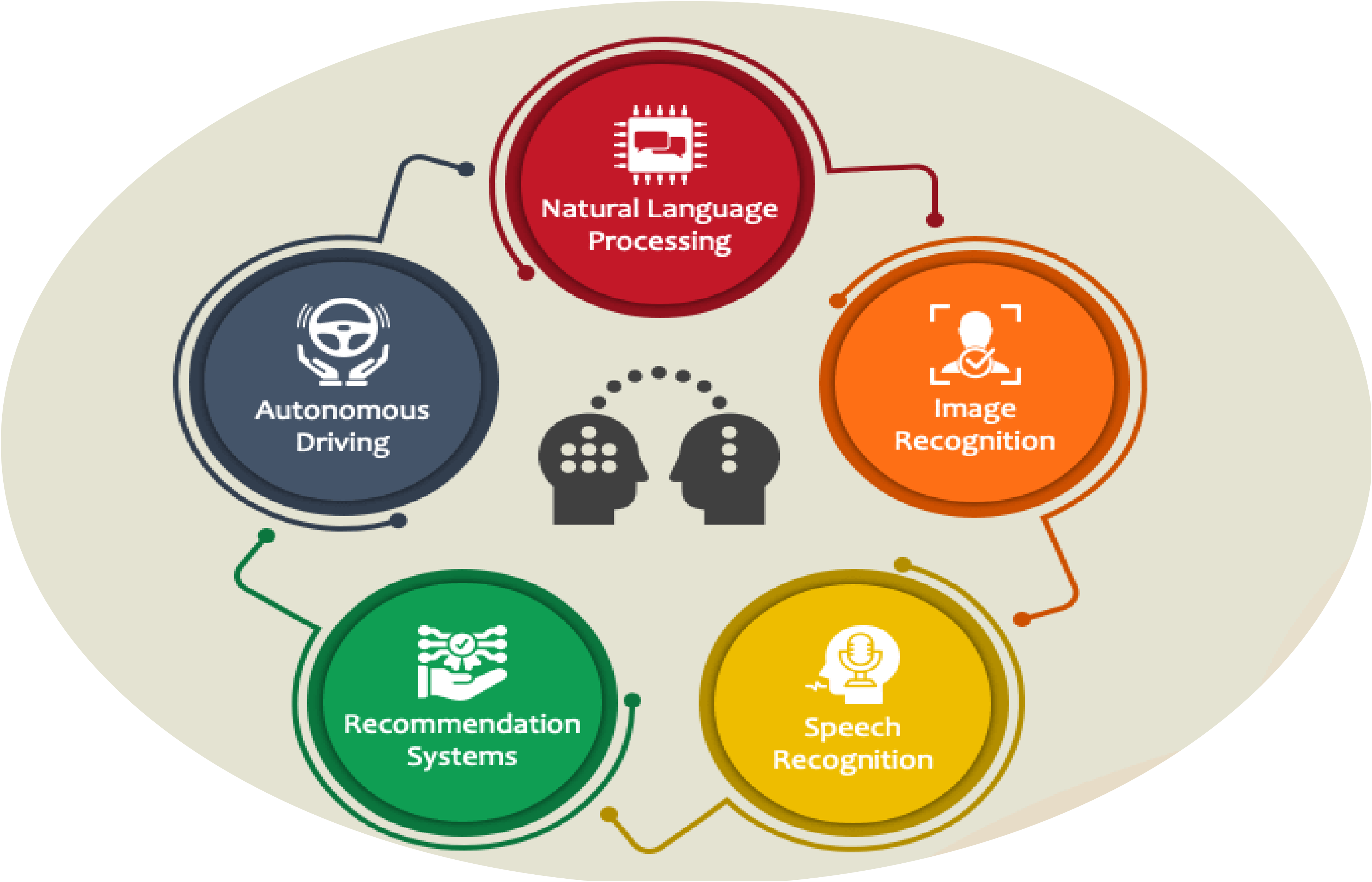
- Real-time object detection and image classification on mobile devices.
- Deployment in IoT devices and augmented reality applications.

Pre-trained Models:

- MobileNetV1, MobileNetV2, MobileNetV3.
- Pre-trained on ImageNet, available for transfer learning.



Applications of transfer Learning



Applications of transfer Learning

Image Classification

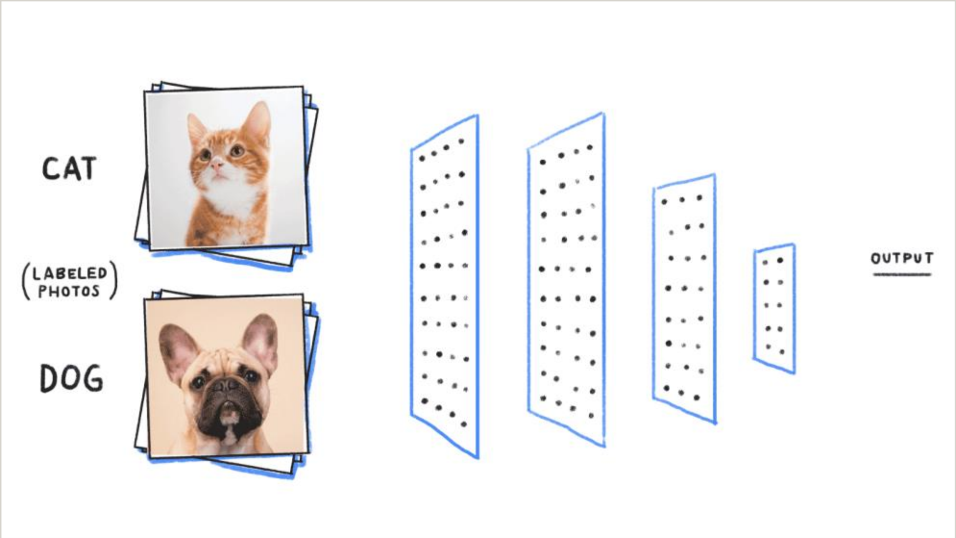
A core application of transfer learning in computer vision.

Pre-trained Models

- Leverage powerful models like ResNet, VGG, and Inception.
- Trained on massive datasets like ImageNet.
- Fine-tune models for specific domains.

Applications:

- For examples, Identifying species in wildlife photography or diagnosing medical conditions from imaging data.



Object Detection:

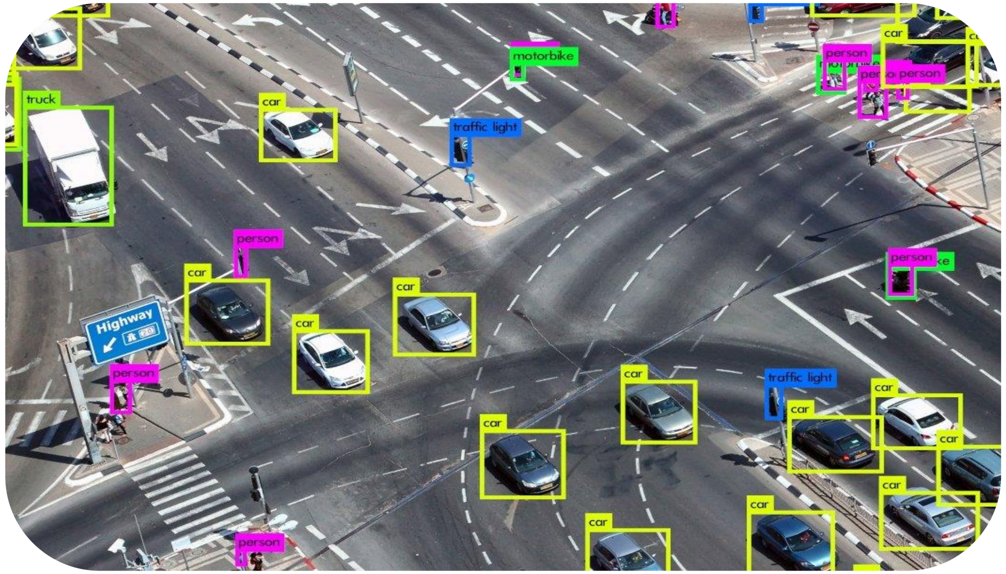
Detect and localize objects in images or videos.

Pre-trained Models:

- Utilize pre-trained models like YOLO, Faster R-CNN, SSD for feature extraction.
- Add layers for bounding box prediction and class identification.

Applications:

- Pedestrian detection for self-driving cars.



Applications of transfer Learning

Image Segmentation

Segmenting images into distinct regions corresponding to objects or parts of objects.

Pre-trained Models:

- U-Net, DeepLab, FCN.

Applications with Transfer Learning

- **Medical Imaging:** Identify tumors or other abnormalities.
- **Autonomous Vehicles:** Differentiate between roads, sidewalks, and vehicles.

Face Recognition:

Identify and verify faces in images or videos.

Pre-trained Models:

- Utilize pre-trained models like FaceNet, VGGFace for feature extraction.
- Add layers for face identification and verification.

Applications:

- Security systems for access control.
- Social media tagging.
- User authentication for devices and apps.

References:

- *https://www.researchgate.net/figure/Change-in-the-number-of-model-parameters-and-training-time-with-increased-number-of_fig3_335865710*
- *https://blog.devgenius.io/all-you-need-to-know-about-transfer-learning-4a7e3cbaf1dd*
- *https://medium.com/@nutanbhogendrasharma/transfer-learning-using-feature-extraction-in-deep-learning-afd97380c96c*
- *https://dev.mrdbourke.com/tensorflow-deep-learning/04_transfer_learning_in_tensorflow_part_1_feature_extraction/*
- *https://encord.com/glossary/pre-trained-model-definition/*
- *https://en.wikipedia.org/wiki/ImageNet*
- *https://blog.mturk.com/tutorial-how-to-label-thousands-of-images-using-the-crowd-bea164ccbefc*
- *https://medium.com/@siddheshb008/alexnet-architecture-explained-b6240c528bd5*
- *62.3 million learnable parameters.*
- *https://www.philschmid.de/getting-started-with-cnn-by-calculating-lenet-layer-manually*
- *https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918*
- *https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/*
- *https://viso.ai/deep-learning/resnet-residual-neural-network/*
- *https://www.researchgate.net/figure/A-schematic-view-of-ResNet-architecture-15-decomposed-into-three-blocks-embedding_fig1_333475917*

References:

- *https://medium.com/@siddheshb008/resnet-architecture-explained-47309ea9283d*
- *https://velog.io/@snoop2head/Going-Deeper-with-Convolutions-GoogleNet-Inception*
- *https://towardsdatascience.com/an-intuitive-guide-to-deep-network-architectures-65fdc477db41*
- *https://www.youtube.com/watch?app=desktop&v=pf-HUcqdCr4*
- *https://www.baeldung.com/wp-content/uploads/sites/4/2020/06/3D_1D_cropped.gif*
- *https://arxiv.org/abs/1905.11946*
- *https://deeplobe.ai/image-segmentation-the-most-interesting-applications/*
- *https://medium.com/analytics-vidhya/how-facial-recognition-systems-work-edcbb227e614*

Thank You