Operation Research Presentation

MRNET For Knee Diagnosis

By: Mohamed Said Ibrahem ID: 57
By: Mohamed Abo Bakr Al-Seddik ID: 51

Contents

- MRNET Using AlexNet.
- Data Generation.
- VGG Model.
- ResNet Model.
- Inception V3 Model.
- Training The DataSet.
- Transfer Learning.
- Accuracy Calculation.

MRNET Using AlexNet.

The primary building block of MRNet is a **convolutional neural network (CNN)**, mapping a 3-dimensional MRI series to a probability.

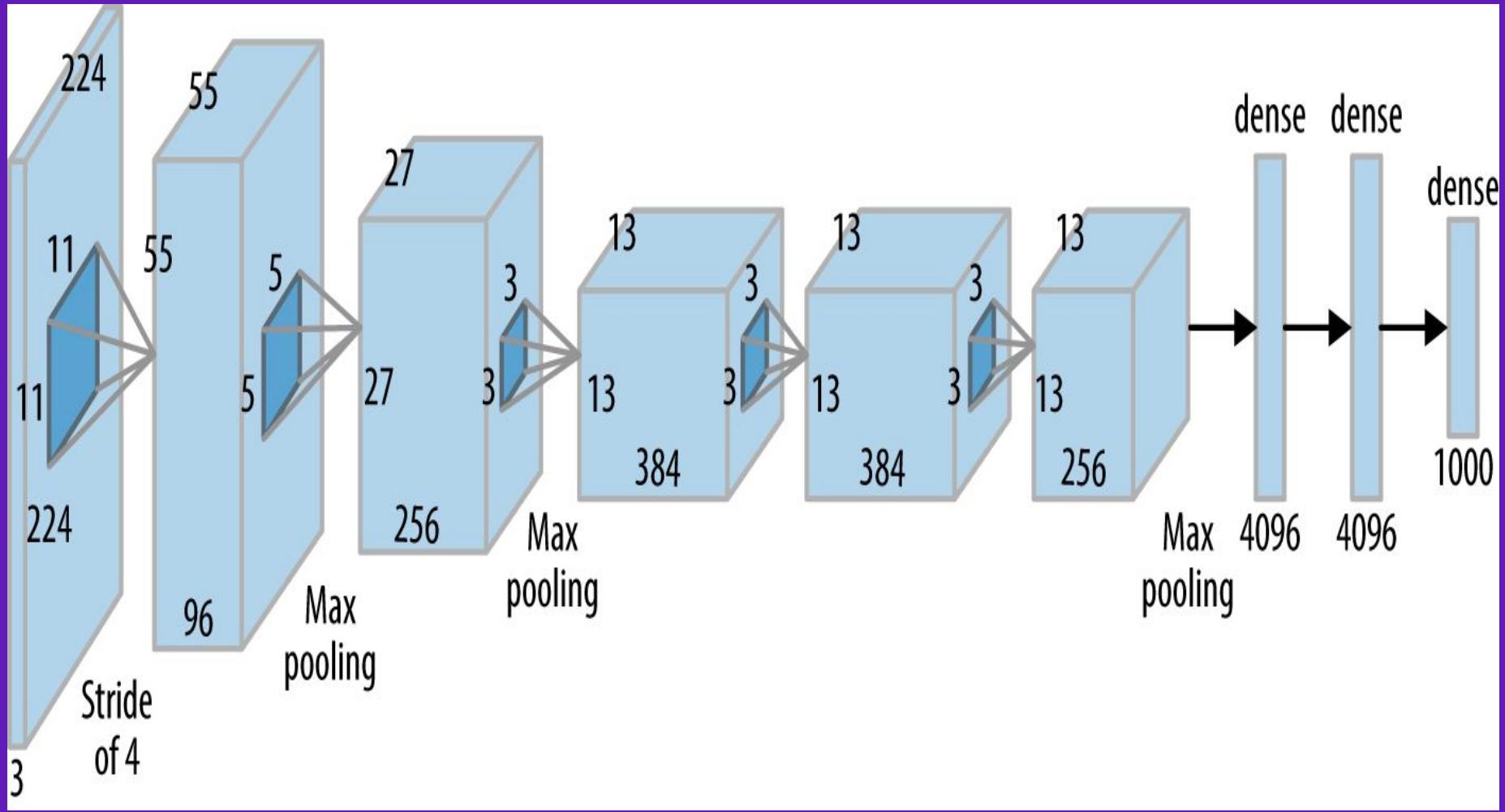
AlexNet architecture consists of **five convolutions** and **three fully-connected** layers.

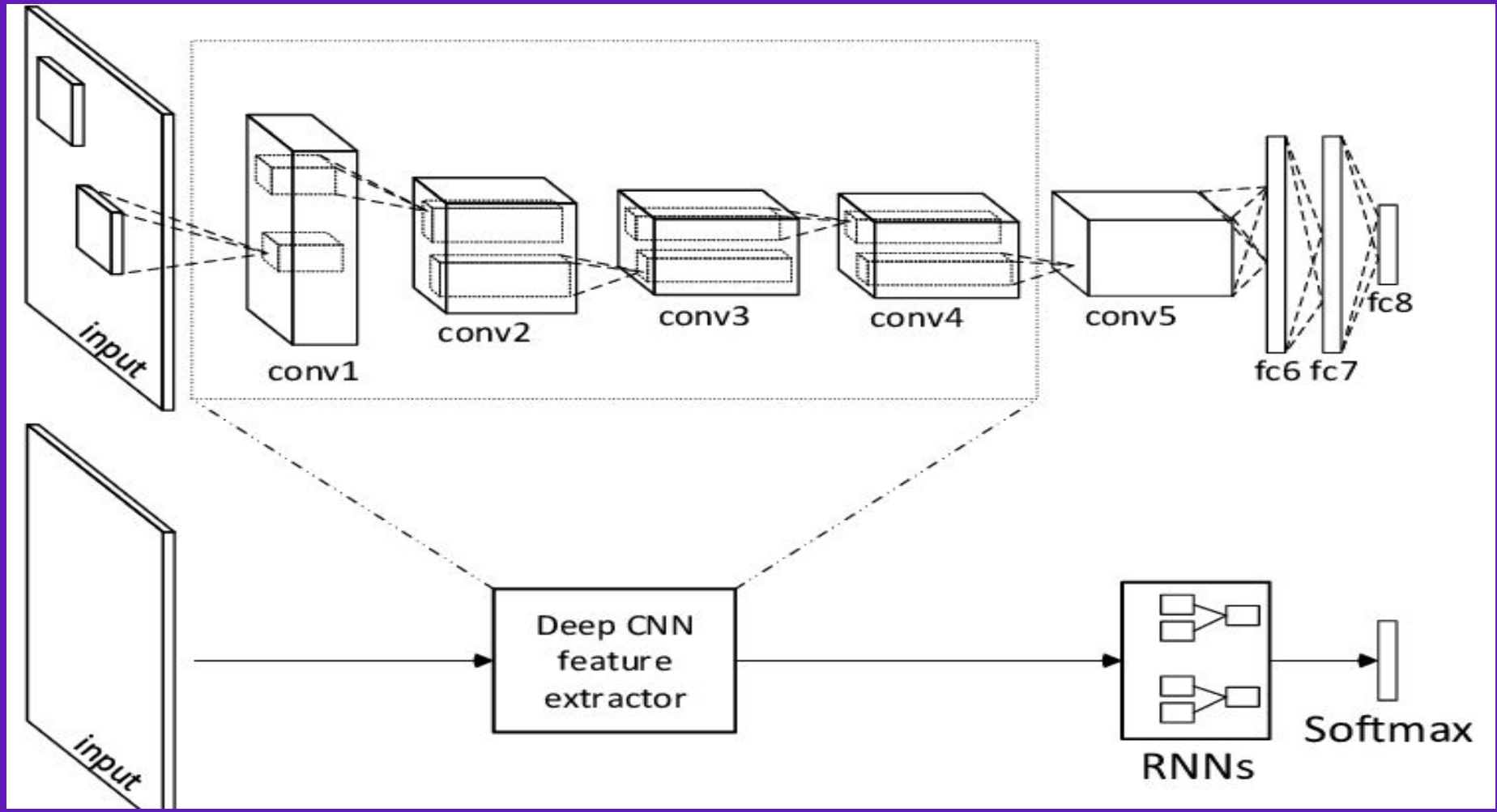
To build the MRNET model, We used an average pooling CNN layer after the AlexNet layer to reduce these features to s x 256 at the end we use **Max-Pooling** to across slices to obtain a 256-dimensional vector, which is passed to a fully connected layer to obtain a prediction probability.

We trained a different MRNet for each task **(abnormality, anterior cruciate ligament tear, meniscal tear)** and series type **(sagittal, coronal, axial)**, resulting in 9 different MRNets.

MRNET Architecture Using AlexNet.

Five convolutions layers,
Three fully-connected layers,
Average Pooling and Max
Pooling at the end.


224
224
3
11
11
Stride
of 4
55
55
96
5
5
Max
pooling
27
27
256
3
3
Max
pooling
13
13
384
3
3
13
13
384
3
3
13
13
256
Max
pooling
dense
4096
dense
4096
dense
1000


input
conv1
conv2
conv3
conv4
conv5
fc6 fc7
fc8
input
Deep CNN feature extractor
RNNs
Softmax

Data Generation.

If you want to load a dataset but there is not enough memory in your machine. As the field of machine learning progresses, this problem becomes more and more common. Today this is already one of the challenges in the field of vision where large datasets of images and video files are processed.

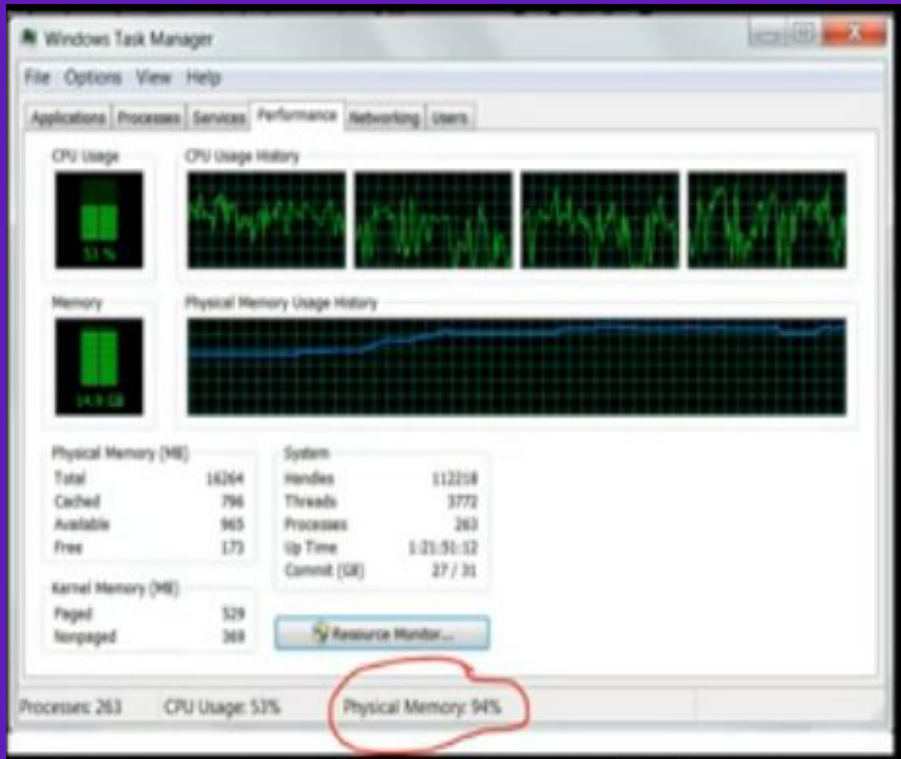
Goal: to generate dataset on multiple cores in real time and feed it right away to the deep learning model.

Data Generation.

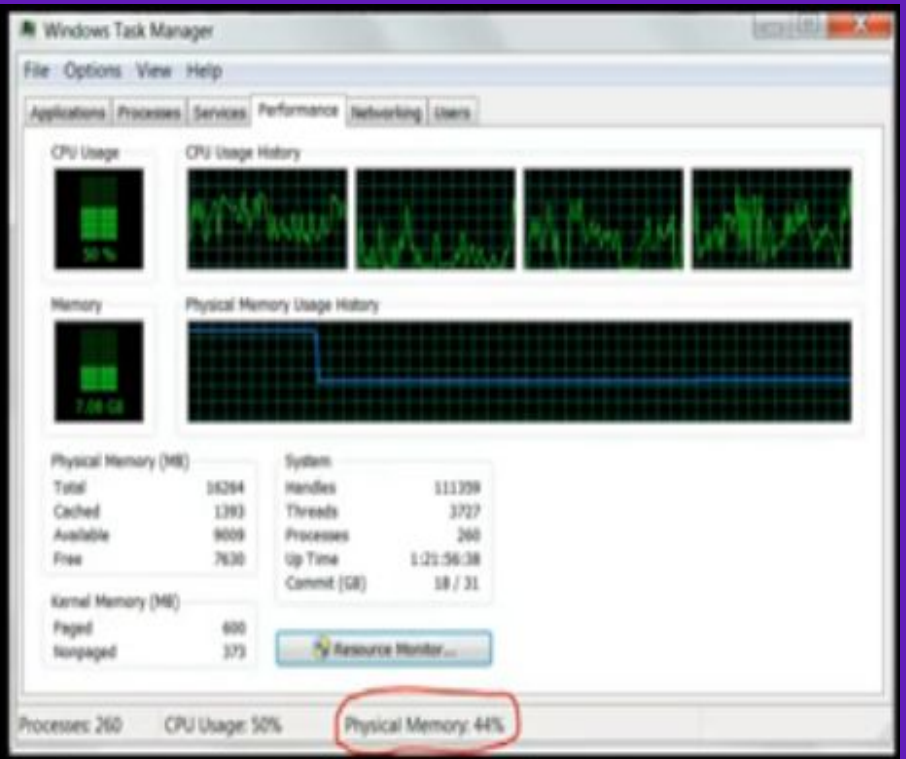
Note That: We will not be able to train data more if we keep misuse our memory (RAM), {in normal data loading}.

So it is so important to use dynamic loading such in Data Generator.

Main Idea: Instead of loading everything at once, generator will load parts of data instead of loading the whole data into memory.



The memory usage (94% and increasing), when using normal function as the entire data sequence is stored in list



The memory usage(only 44%), when using generator function as the yield iterates over the sequence returning element by element from the sequence rather than storing entire sequence in a list.

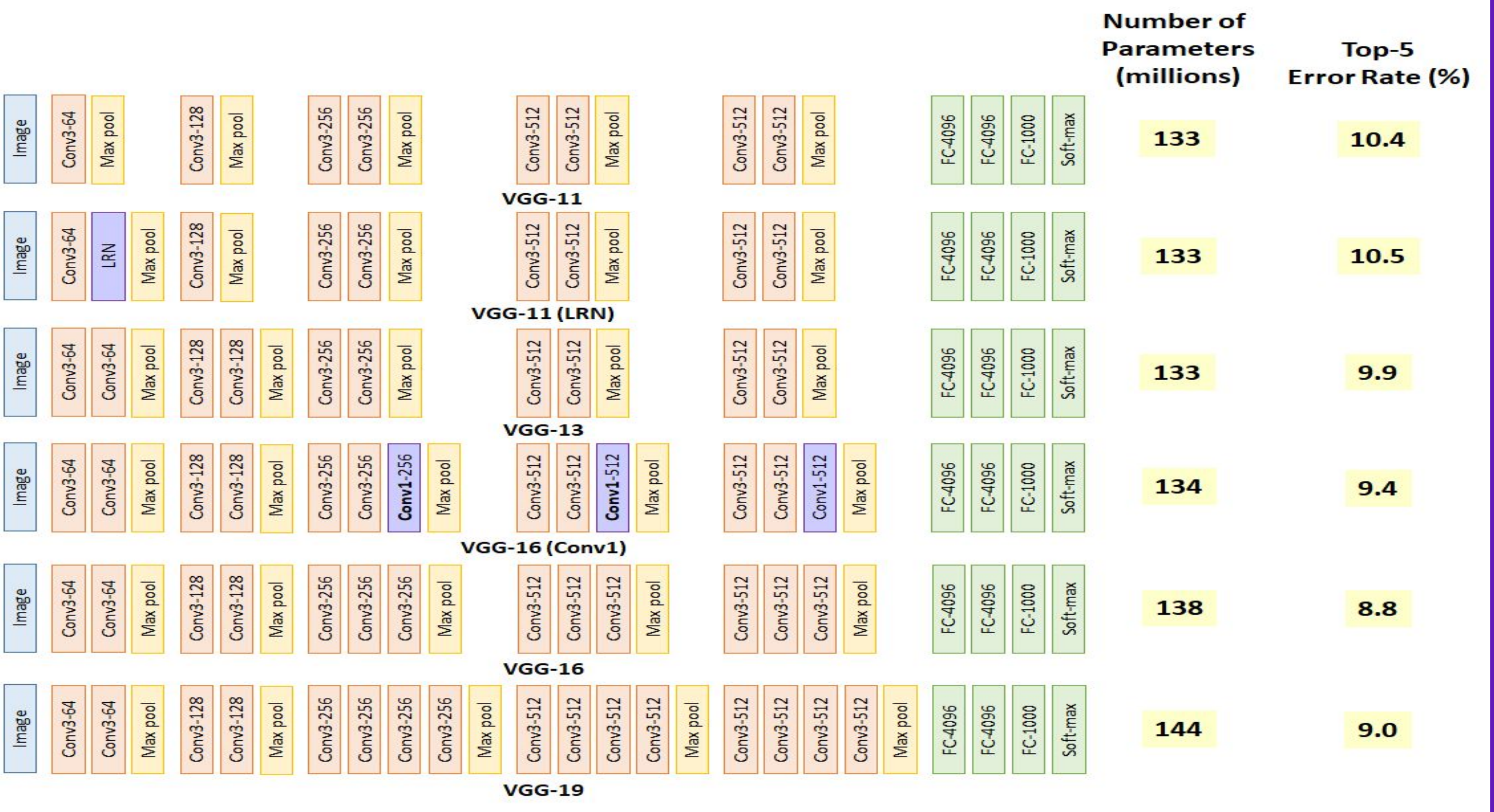
VGG Model.

VGG is a convolutional neural network model. The model achieves 92.7% top-5 test accuracy in ImageNet which is a dataset of over 14 million images belonging to 1000 classes.

VGG stands for **(Visual Geometry Group).**

Note That : We implement VGG model from scratch using the help of Keras Application Repository.

In the next Fig, Here is a Different VGG Layer Structures Using Single Scale (256) Evaluation.



Number of Parameters (millions)
Top-5 Error Rate (%)
Image
Conv3-64
Max pool
Conv3-128
Max pool
Conv3-256
Conv3-256
Max pool
Conv3-512
Conv3-512
Max pool
Conv3-512
Conv3-512
Max pool
FC-4096
FC-4096
FC-1000
Soft-max
133
10.4
VGG-11
Image
Conv3-64
LRN
Max pool
Conv3-128
Max pool
Conv3-256
Conv3-256
Max pool
Conv3-512
Conv3-512
Max pool
Conv3-512
Conv3-512
Max pool
FC-4096
FC-4096
FC-1000
Soft-max
133
10.5
VGG-11 (LRN)
Image
Conv3-64
Conv3-64
Max pool
Conv3-128
Conv3-128
Max pool
Conv3-256
Conv3-256
Max pool
Conv3-512
Conv3-512
Max pool
Conv3-512
Conv3-512
Max pool
FC-4096
FC-4096
FC-1000
Soft-max
133
9.9
VGG-13
Image
Conv3-64
Conv3-64
Max pool
Conv3-128
Conv3-128
Max pool
Conv3-256
Conv3-256
Conv1-256
Max pool
Conv3-512
Conv3-512
Conv1-512
Max pool
Conv3-512
Conv3-512
Conv1-512
Max pool
FC-4096
FC-4096
FC-1000
Soft-max
134
9.4
VGG-16 (Conv1)
Image
Conv3-64
Conv3-64
Max pool
Conv3-128
Conv3-128
Max pool
Conv3-256
Conv3-256
Conv3-256
Max pool
Conv3-512
Conv3-512
Conv3-512
Max pool
Conv3-512
Conv3-512
Conv3-512
Max pool
FC-4096
FC-4096
FC-1000
Soft-max
138
8.8
VGG-16
Image
Conv3-64
Conv3-64
Max pool
Conv3-128
Conv3-128
Max pool
Conv3-256
Conv3-256
Conv3-256
Conv3-256
Max pool
Conv3-512
Conv3-512
Conv3-512
Conv3-512
Max pool
Conv3-512
Conv3-512
Conv3-512
Conv3-512
Max pool
FC-4096
FC-4096
FC-1000
Soft-max
144
9.0
VGG-19

ResNet Model.

ResNet stands for Residual Networks.

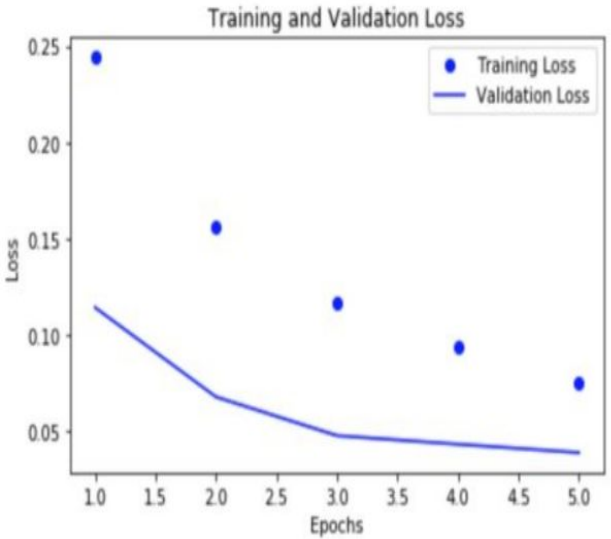
It is a classic neural network used as a backbone for many computer vision tasks. This model was the winner of ImageNet challenge in 2015.

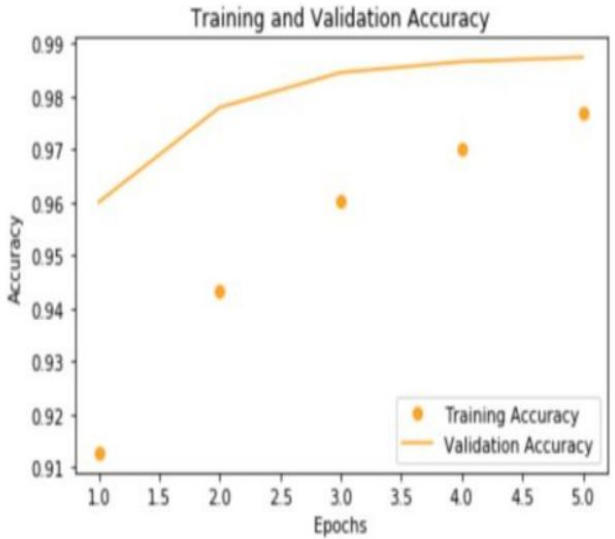
ResNet first introduced the concept of **skip connection**, so we can add the original input to the output of the convolution block.

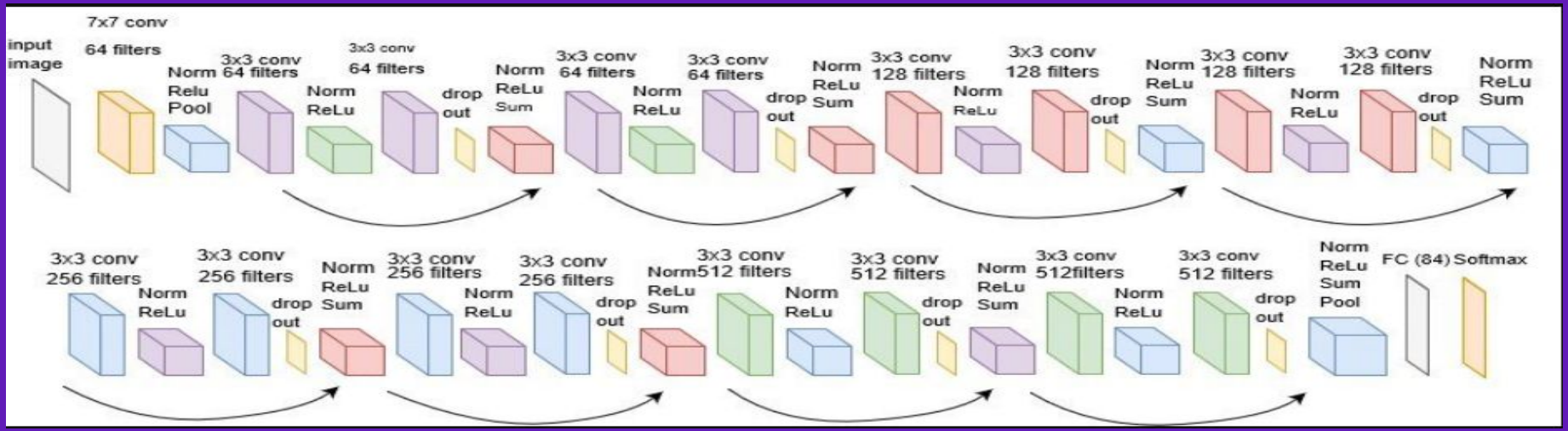
ResNet Model.

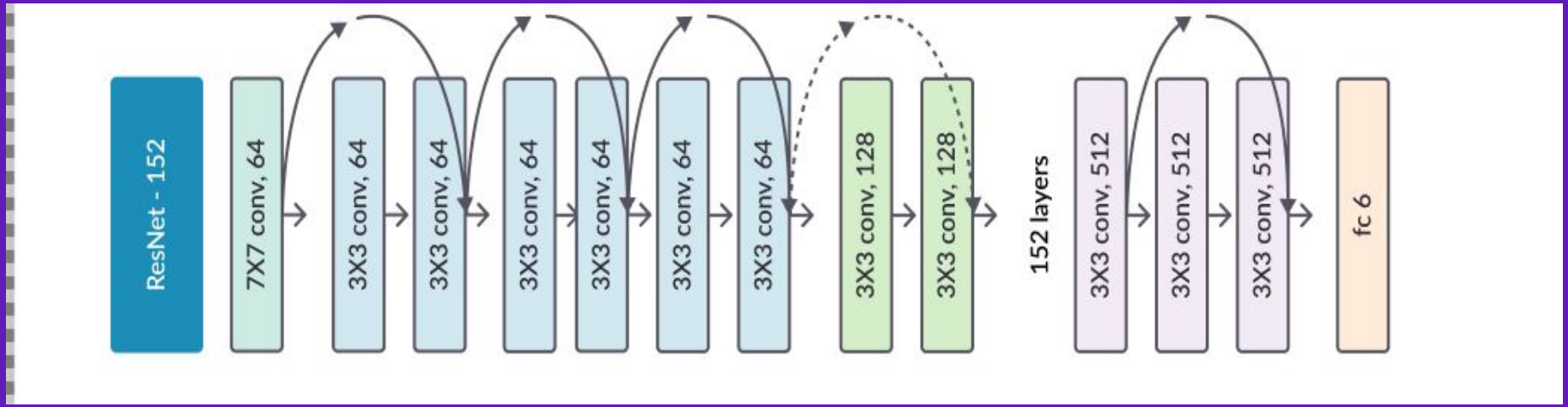
Note that: ResNet is the <u>Best Model</u> in the three models.

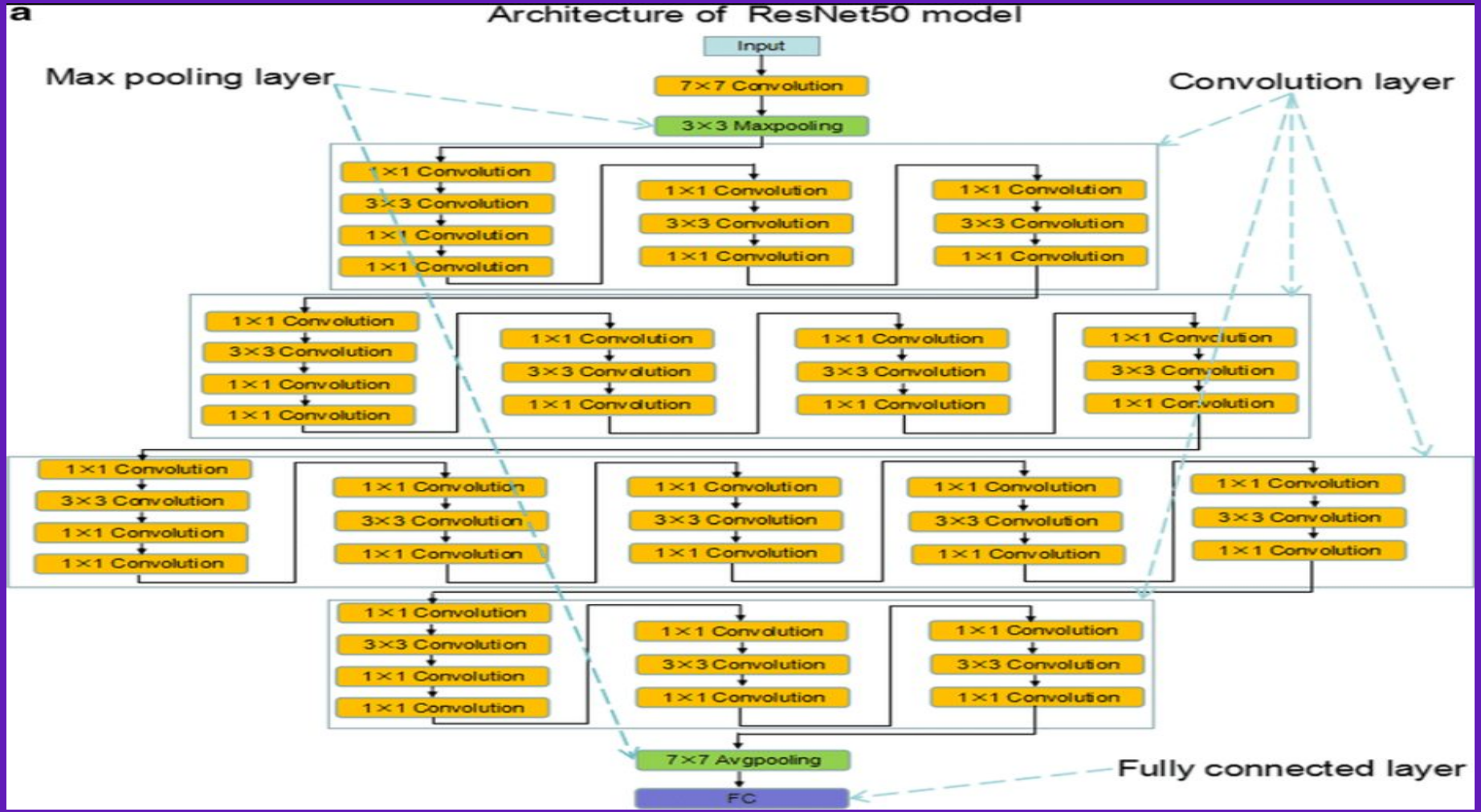
RestNet Validation and Accuracy Comparison:






input image
7x7 conv
64 filters
Norm
Relu
Pool
3x3 conv
64 filters
Norm
ReLu
3x3 conv
64 filters
drop
out
Norm
ReLu
Sum
3x3 conv
64 filters
Norm
ReLu
3x3 conv
64 filters
drop
out
Norm
ReLu
Sum
3x3 conv
128 filters
Norm
ReLu
3x3 conv
128 filters
drop
out
Norm
ReLu
Sum
3x3 conv
128 filters
Norm
ReLu
3x3 conv
128 filters
drop
out
Norm
ReLu
Sum
3x3 conv
256 filters
Norm
ReLu
3x3 conv
256 filters
drop
out
Norm
ReLu
Sum
3x3 conv
256 filters
Norm
ReLu
3x3 conv
256 filters
drop
out
Norm
ReLu
Sum
3x3 conv
512 filters
Norm
ReLu
3x3 conv
512 filters
drop
out
Norm
ReLu
Sum
3x3 conv
512filters
Norm
ReLu
3x3 conv
512 filters
drop
out
Norm
ReLu
Sum
Pool
FC (84)
Softmax


ResNet - 152
7X7 conv, 64
3X3 conv, 64
3X3 conv, 64
3X3 conv, 64
3X3 conv, 64
3X3 conv, 64
3X3 conv, 64
3X3 conv, 128
3X3 conv, 128
152 layers
3X3 conv, 512
3X3 conv, 512
3X3 conv, 512
fc 6

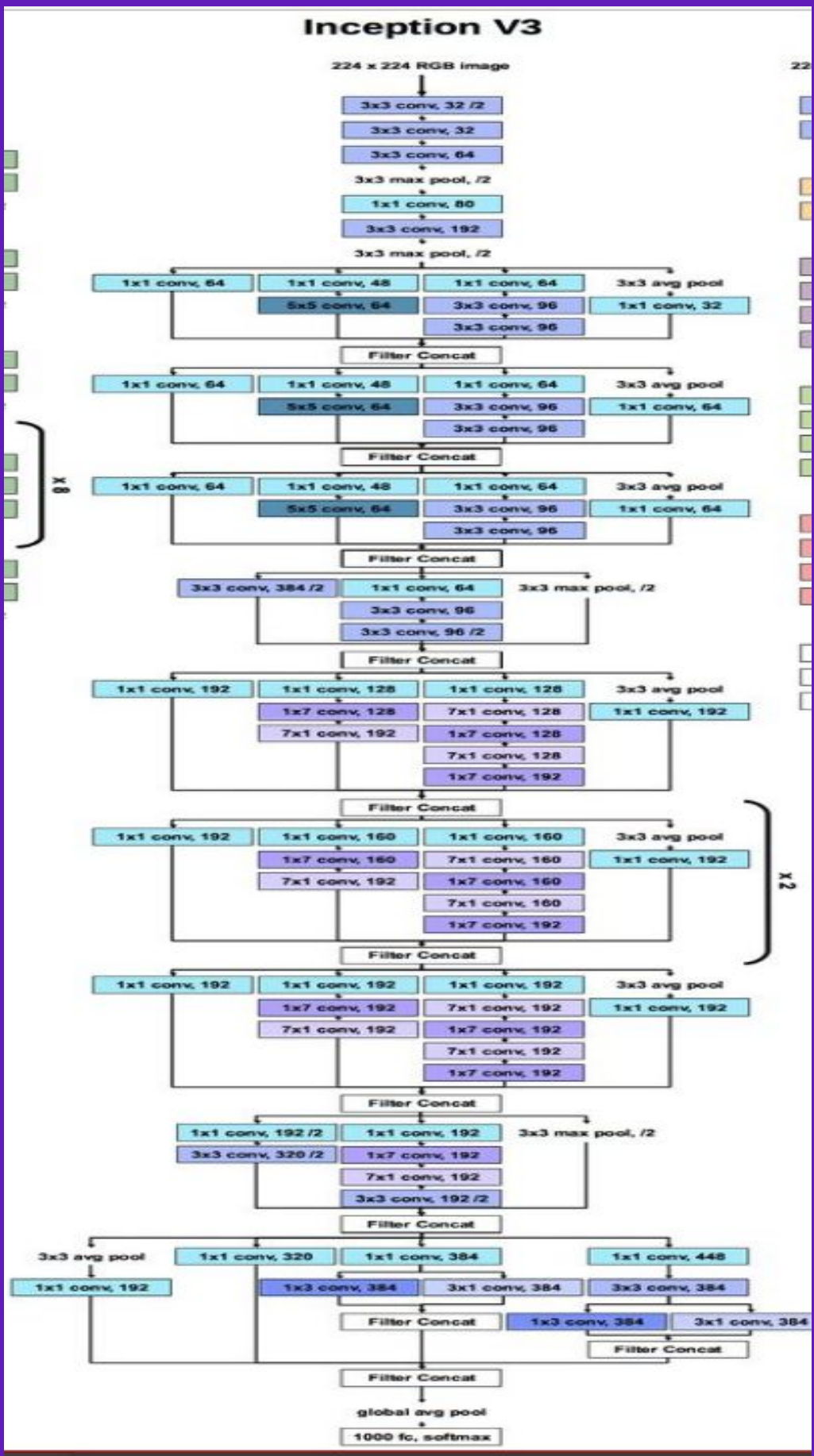

a
Architecture of ResNet50 model
Input
7×7 Convolution
3×3 Maxpooling
Max pooling layer
Convolution layer
1×1 Convolution
3×3 Convolution
1×1 Convolution
7×7 Avgpooling
FC
Fully connected layer

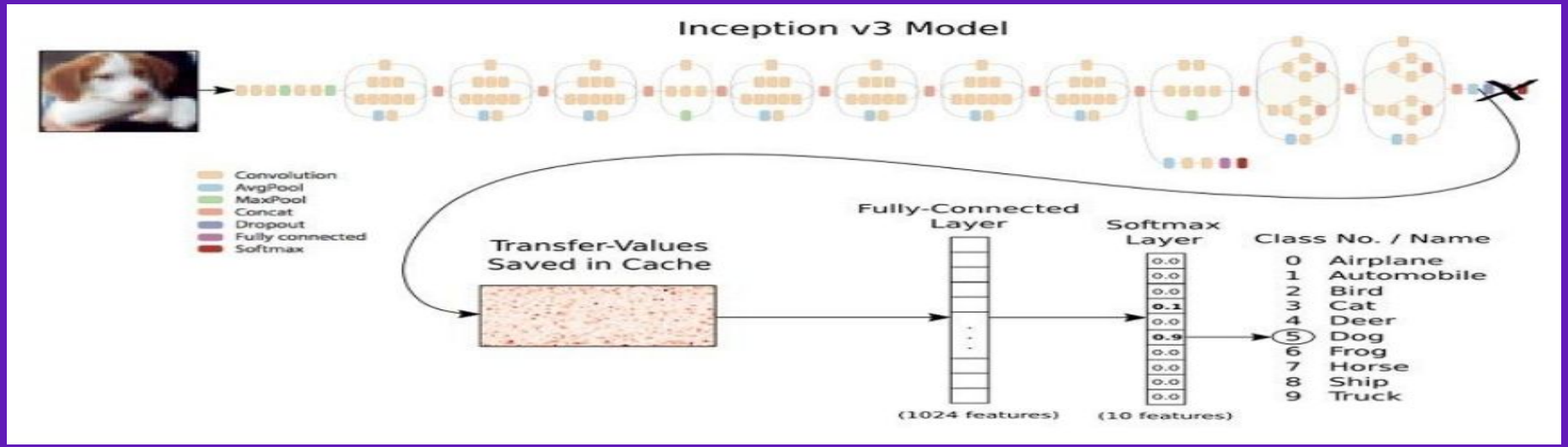
Inception V3 Model.

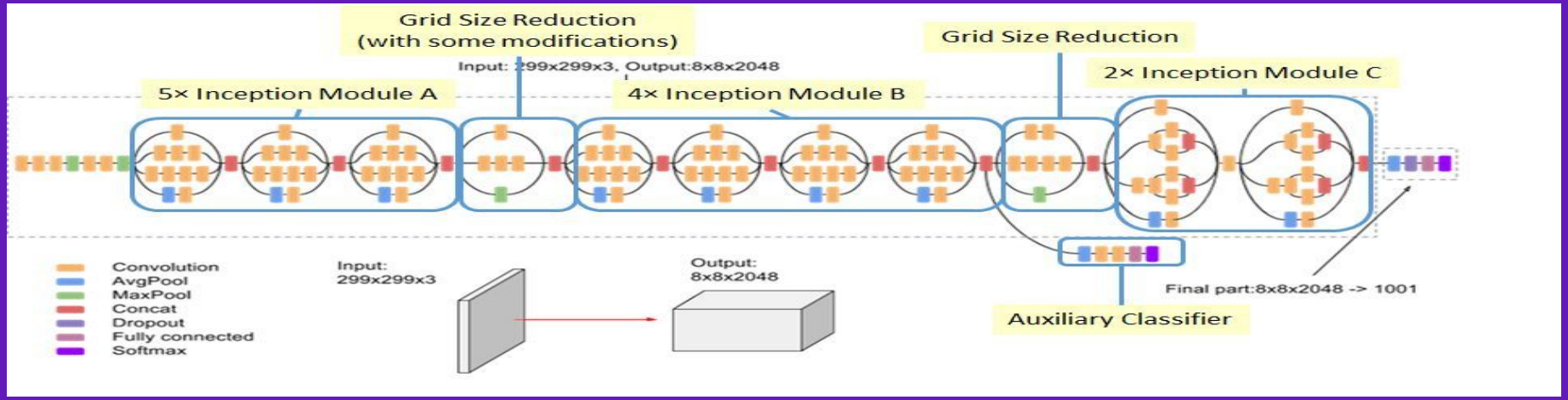
Inception-v3 is a convolutional neural network that is 48 layers deep.

It is the third edition of Google's Inception Convolutional Neural Network.

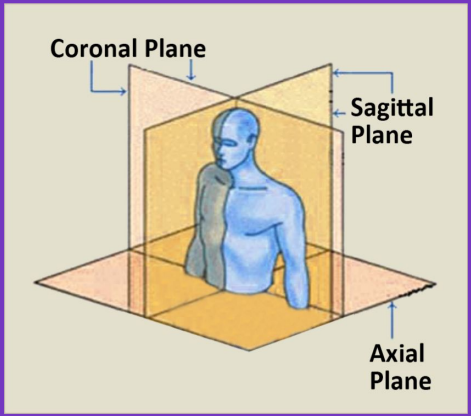
Note that: the input image format for this model is different than for the VGG16 and ResNet models (299x299 instead of 224x224), and that the input preprocessing function is also different (same as Xception).


Inception V3
224 x 224 RGB image
3x3 conv, 32 /2
3x3 conv, 32
3x3 conv, 64
3x3 max pool, /2
1x1 conv, 80
3x3 conv, 192
3x3 max pool, /2
1x1 conv, 64
1x1 conv, 48
5x5 conv, 64
1x1 conv, 64
3x3 conv, 96
3x3 conv, 96
3x3 avg pool
1x1 conv, 32
Filter Concat
1x1 conv, 64
1x1 conv, 48
5x5 conv, 64
1x1 conv, 64
3x3 conv, 96
3x3 conv, 96
3x3 avg pool
1x1 conv, 64
Filter Concat
x 8
1x1 conv, 64
1x1 conv, 48
5x5 conv, 64
1x1 conv, 64
3x3 conv, 96
3x3 conv, 96
3x3 avg pool
1x1 conv, 64
Filter Concat
3x3 conv, 384 /2
1x1 conv, 64
3x3 conv, 96
3x3 conv, 96 /2
3x3 max pool, /2
Filter Concat
1x1 conv, 192
1x1 conv, 128
1x7 conv, 128
7x1 conv, 192
1x1 conv, 128
7x1 conv, 128
1x7 conv, 128
7x1 conv, 128
1x7 conv, 192
3x3 avg pool
1x1 conv, 192
Filter Concat
1x1 conv, 192
1x1 conv, 160
1x7 conv, 160
7x1 conv, 192
1x1 conv, 160
7x1 conv, 160
1x7 conv, 160
7x1 conv, 160
1x7 conv, 192
3x3 avg pool
1x1 conv, 192
Filter Concat
x 2
1x1 conv, 192
1x1 conv, 192
1x7 conv, 192
7x1 conv, 192
1x1 conv, 192
7x1 conv, 192
1x7 conv, 192
7x1 conv, 192
1x7 conv, 192
3x3 avg pool
1x1 conv, 192
Filter Concat
1x1 conv, 192 /2
3x3 conv, 320 /2
1x1 conv, 192
1x7 conv, 192
7x1 conv, 192
3x3 conv, 192 /2
3x3 max pool, /2
Filter Concat
3x3 avg pool
1x1 conv, 192
1x1 conv, 320
1x1 conv, 384
1x3 conv, 384
3x1 conv, 384
Filter Concat
1x1 conv, 448
3x3 conv, 384
1x3 conv, 384
3x1 conv, 384
Filter Concat
Filter Concat
global avg pool
1000 fc, softmax


Inception v3 Model
Convolution
AvgPool
MaxPool
Concat
Dropout
Fully connected
Softmax
Transfer-Values
Saved in Cache
Fully-Connected
Layer
(1024 features)
Softmax
Layer
(10 features)
Class No. / Name
0 Airplane
1 Automobile
2 Bird
3 Cat
4 Deer
5 Dog
6 Frog
7 Horse
8 Ship
9 Truck


Grid Size Reduction
(with some modifications)
Input: 299x299x3, Output:8x8x2048
Grid Size Reduction
5× Inception Module A
4× Inception Module B
2× Inception Module C
Convolution
AvgPool
MaxPool
Concat
Dropout
Fully connected
Softmax
Input:
299x299x3
Output:
8x8x2048
Final part:8x8x2048 -> 1001
Auxiliary Classifier

Training The DataSet.



We have mainly three models (VGG, ResNet, Inception) + ALexNet Model.

We Train our DataSet in our three direction (Axial, Coronal, Sagittal) for each of the three injury (Abnormality, ACL, Meniscus).

We divide the training split of the dataset into training part (90%) and validation part (10%) using validation_split parameter.

Transfer Learning.

What is Transfer Learning?

TL is to start training your network starting from a random state (Pre-Trained Weights) on famous datasets like ImageNet which is (14 million images).

We Used ImageNet Weights to do transfer learning with **ResNet** (as it is the best model).

Accuracy Calculation.

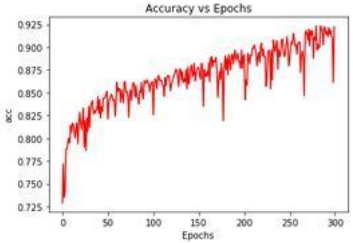
Here is the overall accuracy for the dataset classes:

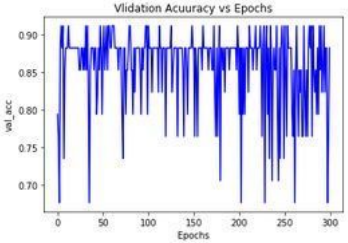
1- Abnormal : 85.08496701055103 %

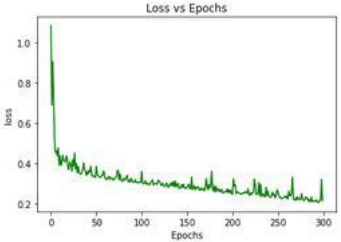
2- ACL : 55.398693366183174 %

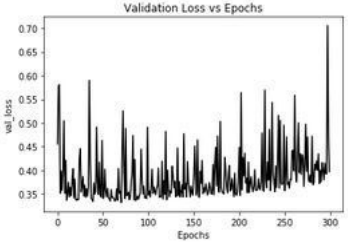
3- Meniscus : 81.12091491619746 %

We calculate and plot (accuracy - validation accuracy - loss - validation loss).

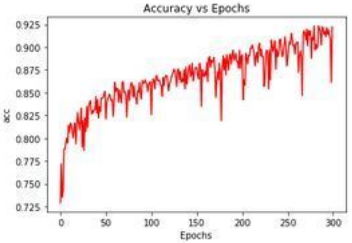


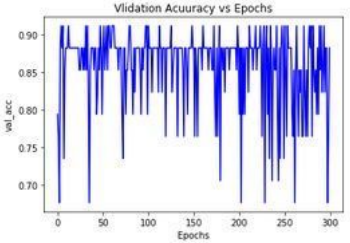


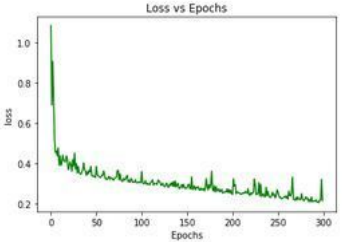


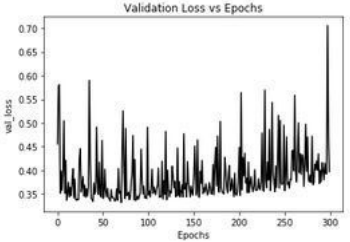


Example of some plots shown the Accuracy Vs Epochs for both validation and loss









Many Thanks To

Advisors

Prof. Marwan A. Torki.

Prof. Ayman Khalafallah.