

**ARTIFICIAL INTELLIGENCE PROGRAMMING PROJECT**

**Report 4 – Fundamental Algorithm**

– Hanoi, October 2021 –

**Table of Contents**

[IV. Fundamental Algorithms 3](#_heading=h.30j0zll)

[1. Overview 3](#_heading=h.1fob9te)

[2. Pseudo code](#_heading=h.3znysh7) 9

# IV. Fundamental Algorithms

## 1. Overview

* To tackle the keypoints detection issue, The Interception-Like algorithm will be implemented and for the area of improvement, we use the transfer learning method to improve the model.



**2. “Pseudo” code**

## 2.1. Convolutional network

## Convolutional network is made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function (e.g. SVM/Softmax) on the last (fully-connected) layer and all the tips/tricks we developed for learning regular Neural Networks still apply.[2]

* The convolutional network consists of 3 layers which are Convolutional Layer, Pooling Layer, and Fully-Connected Layer.

|  |
| --- |
| Fig.2: Example of Convolutional network |

1. **Convolutional Layer**

* The Conv layer is the core building block of a Convolutional Network that does most of the computational heavy lifting. The role of the ConvEnt is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.
* The objective of the Convolution Operation is to **extract the high-level features** such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving a network which has the wholesome understanding of images in the dataset. [3]
* In the convolutional layer, the extraction process often occurs in the beginning of the network and applies the ReLU activation. The ReLU activation involves only a comparison between its input and the value 0 and it also has a derivative of either 0 or 1, depending on whether its input is respectively negative or not (Fig.3).

|  |
| --- |
|  |
| Fig.3: Example of ReLU activation. |

#### Pooling layer

* Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training the model.
* There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel (Fig.4).

|  |
| --- |
| Fig.4: Types of pooling |

#### 

#### Fully connected layer

* The fully connected layer (FC) operates on a flattened input where each input is connected to all neurons. FC layers are placed at the end of CNN architectures and can be used to optimize objectives such as class scores (Fig.5). [3]

|  |
| --- |
| Fig.5: Fully connected layer |

## 2.2 Inception model

* Inception v3 is a widely-used image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. The model is the culmination of many ideas developed by multiple researchers over the years. It is studied by Christian Szegedy [5].
* The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batchnorm is used extensively throughout the model and applied to activation inputs. Loss is computed via Softmax.[Fig.6]

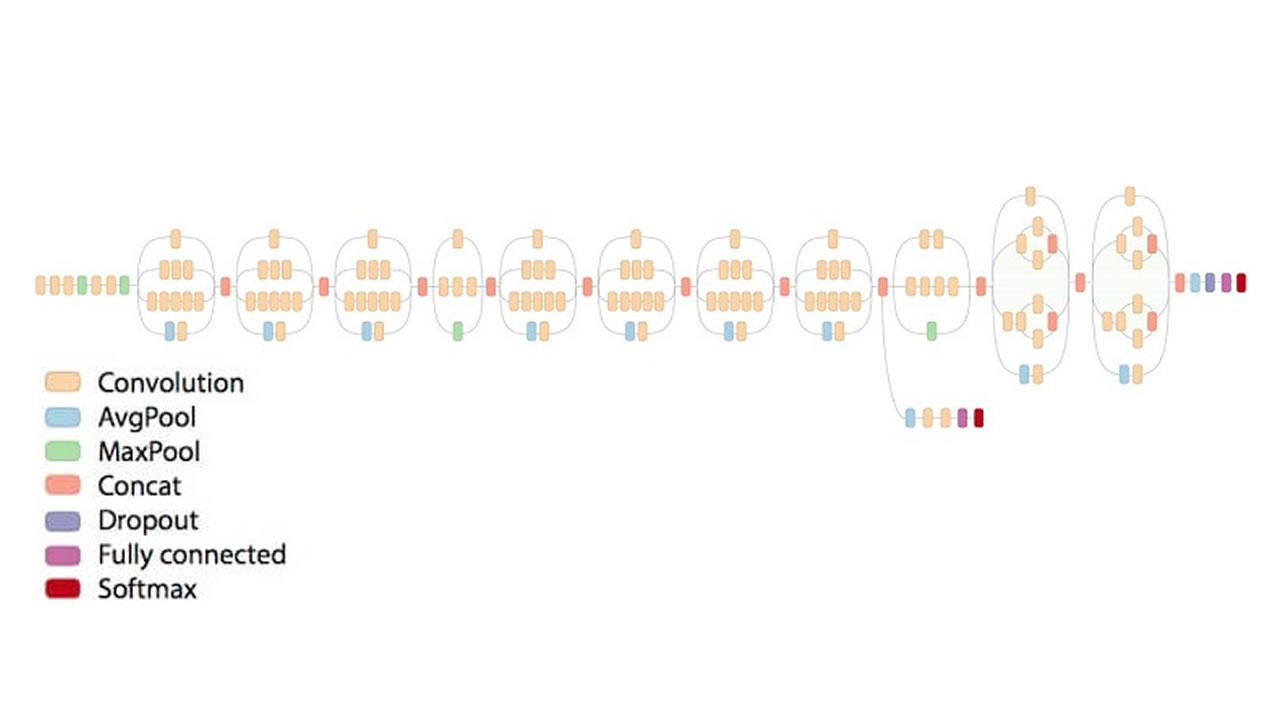
****

Fig.6: Inception v3 architecture

* The architecture of an Inception v3 network is progressively built, step-by-step

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. [Fig.7] is an example, 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.

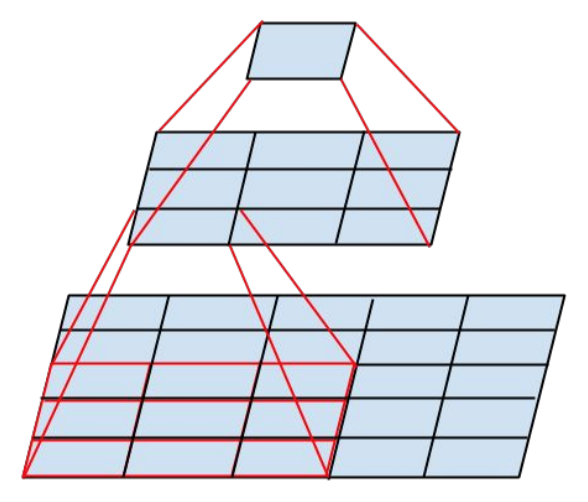


Fig.7:Smaller convolutions

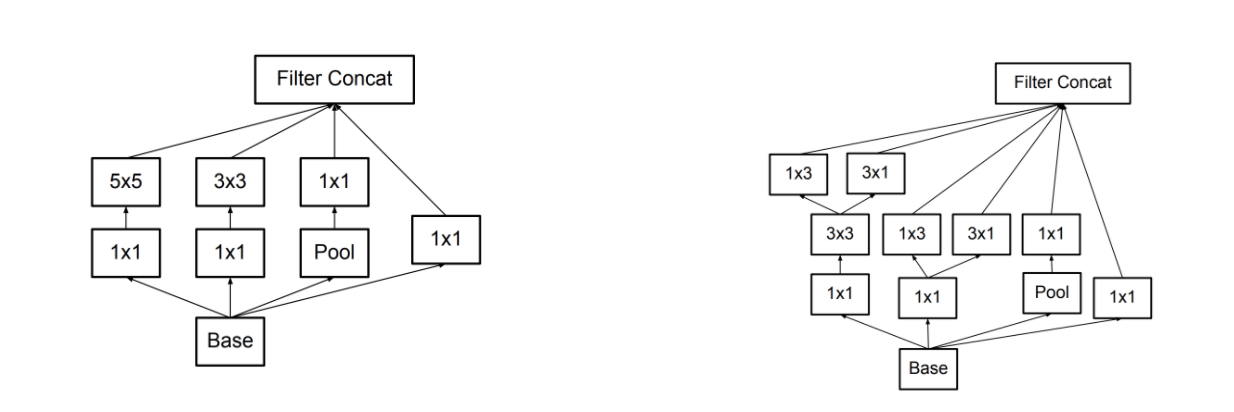
1. Asymmetric convolutions: A 3 × 3 convolution could be replaced by a 1 × 3 convolution followed by a 3 × 1 convolution. If a 3 × 3 convolution is replaced by a 2 × 2 convolution, the number of parameters would be slightly higher than the asymmetric convolution proposed.[Fig.8]

Fig.8: Example of Asymmetric convolutions

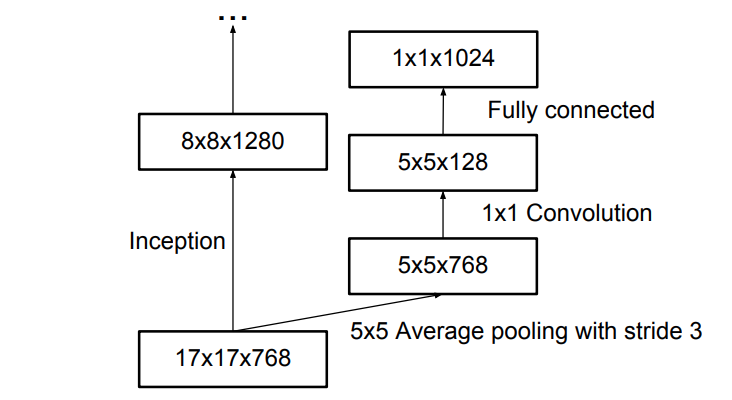
1. Auxiliary classifier: an auxiliary classifier is a small CNN inserted between layers during training, and the loss incurred is added to the main network loss. In GoogLeNet auxiliary classifiers were used for a deeper network, whereas in Inception v3 an auxiliary classifier acts as a regularizer.[Fig.9]

Fig.9: Auxiliary classifier’s example

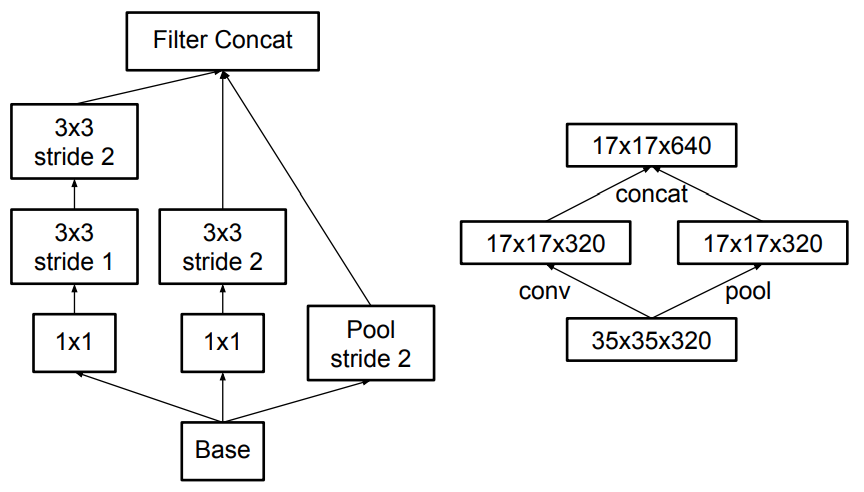
1. Grid size reduction: Grid size reduction is usually done by pooling operations. However, to combat the bottlenecks of computational cost, a more efficient technique is proposed in [Fig.10]:

Fig.10: A more efficient technique for grid size reduction

## 2.3 Transfer Learning

* Transfer learning is a machine learning technique where a model trained on one task is re-purposed on a second related task. More particularly, if we have significantly more data for task T1, we may utilize its learning, and generalize this knowledge (features, weights) for task T2 (which has significantly less data). In the case of problems in the computer vision domain, certain low-level features, such as edges, shapes, corners and intensity, can be shared across tasks, and thus enable knowledge transfer among tasks.
* There is a stark difference between the traditional approach of building and training machine learning models, and using a methodology following transfer learning principles. [1] (Fig.1)
* Transfer learning is related to problems such as multi-task learning and concept drift and is not exclusively an area of study for deep learning. Nevertheless, transfer learning is popular in deep learning given the enormous resources required to train deep learning models or the large and challenging datasets on which deep learning models are trained.
* In transfer learning, there are two major approaches which are develop model approach and pre-trained model approach. In this project, a pre-trained model will be applied.
* A pre-trained model is a model that was trained on a large benchmark dataset to solve a problem similar to the one that we want to solve. Accordingly, due to the computational cost of training such models, it is common practice to import and use models from published literature. In computer vision, there are numerous pre-trained models such as VGG-16, VGG-19, ResNet-50...

**Reference**

## [1][A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning](https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a). Dipanjan (DJ) Sarkar 2018

[2][CS231n Convolutional Neural Networks for Visual Recognition](https://cs231n.github.io/).

[3][A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way](https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53). Sumit Saha 2018

[4] CS230 Convolutional Neural Networks cheatsheet. Afshine Amidi and Shervine Amidi

[5]Rethinking the Inception Architecture for Computer Vision. Christian Szegedy 2015