**Image Recognition and Extraction of deep features using GoogLeNet**

Sowmya Ch, Kukkadapu Sushma, Sai Sruthi K

**Keywords :** *Inception v1; GoogLeNet; Image Recognition; Keras; Theano; Convolution Neural Networks; ImageNet Large-Scale Visual Recognition Challenge.*

**Abstract**

Image Recognition is one of the hallmark tasks of Computer Vision. It defines a context for recognition of objects. A deep neural network architecture named as ‘Inception’ achieved new state of art for classification and recognition in the ImageNet Large-Scale Visual Recognition (ILSVR). One of the incarnations of ILSVR is called ‘GoogLeNet’. It is 22 layered deep network.

GoogLeNet was one of the first models that introduced the idea of ‘Convolutional Neural Network’ (CNN) layers need not be always stacked up sequentially. Thus, the evolution of Inception module is defined as a creative structuring of layers like a network inside a network which leads to improved performance of algorithms in recognition and extraction of deep features of images.

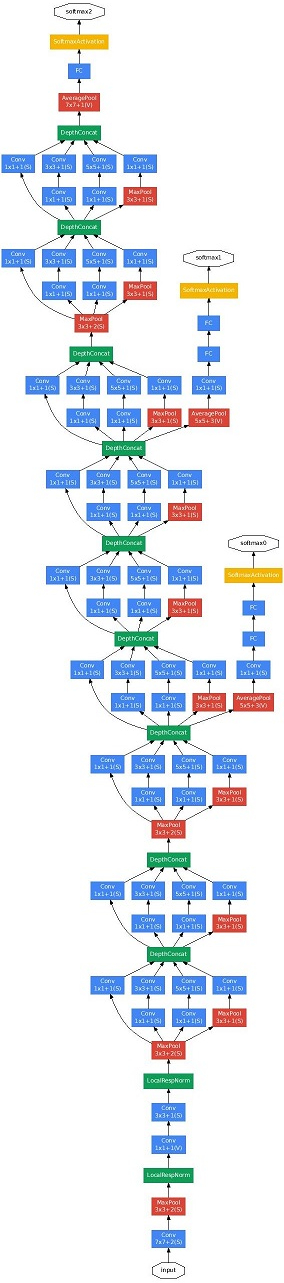
With the ever increasing computational power, the need for processing images is going beyond the two-dimensional. To go beyond is to extract deep features from that image. For such a deeper vision, a deep learning library called ‘Keras’ is used in our project simulation. This library runs on top of Theano as its backend. ‘Theano’ is a Python library that allows optimizing and evaluating mathematical expressions which involve multidimensional arrays efficiently. These strong libraries are responsible in strengthening the image recognition accuracy. Thus the concept of Inception will set the future of Computational Intelligence in Image Recognition.

**1. Introduction**

In the last three years, the object classification and detection capabilities have dramatically improved due to advances in deep learning and convolutional networks. One encouraging news is that most of this progress is not just the result of more powerful hardware, larger datasets and bigger models, but mainly a consequence of new ideas, algorithms and improved network architectures.

**1.1 GoogLeNet**

The network is 22 layers deep when counting only layers with parameters (or 27 layers if we also count pooling). The overall number of layers (independent building blocks) used for the construction of the network is about 100. The exact number depends on how layers are counted by the machine learning infrastructure. The linear layer enables us to easily adapt our networks to other label sets. Moving from fully connected layers to average pooling improved the top-1 accuracy by about 0.6%, however the use of dropout remained essential even after removing the fully connected layers.

****

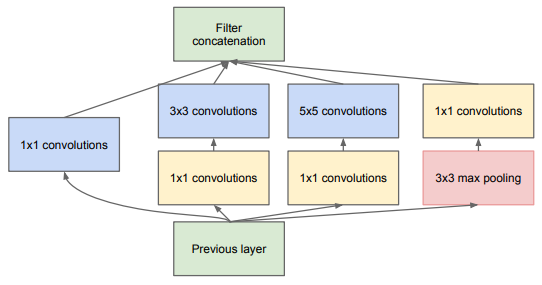
**Figure 1.1 : GoogLeNet**

The exact structure of the extra network on the side, including the auxiliary classifier, is as follows:

* An average pooling layer with 5×5 filter size and stride 3, resulting in an 4×4×512 output for the (4a), and 4×4×528 for the (4d) stage.
* A 1×1 convolution with 128 filters for dimension reduction and rectified linear activation.
* A fully connected layer with 1024 units and rectified linear activation.
* A dropout layer with 70% ratio of dropped outputs.
* A linear layer with softmax loss as the classifier (predicting the same 1000 classes as the main classifier, but removed at inference time).

**1.2 Inception module Architecture**

The main idea of the Inception architecture is to judiciously reduce dimension wherever the computational requirements would increase too much otherwise. This is based on the success of embeddings: even low dimensional embeddings might contain a lot of information about a relatively large image patch. However, embeddings represent information in a dense, compressed form and compressed information is harder to process. The representation should be kept sparse at most places and compress the signals only whenever they have to be aggregated en masse. That is, 1×1 convolutions are used to compute reductions before the expensive 3×3 and 5×5 convolutions. Besides being used as reductions, they also include the use of rectified linear activation making them dual-purpose. The final result is depicted in Figure 1.2 In general, an Inception network is a network consisting of modules of the above type stacked upon each other, with occasional max-pooling layers with stride 2 to halve the resolution of the grid. For technical reasons (memory efficiency during training), it seemed beneficial to start using Inception modules only at higher layers while keeping the lower layers in traditional convolutional fashion. A useful aspect of this architecture is that it allows for increasing the number of units at each stage significantly without an uncontrolled blow-up in computational complexity at later stages. This is achieved by the ubiquitous use of dimensionality reduction prior to expensive convolutions with larger patch sizes. Furthermore, the design follows the practical intuition that visual information should be processed at various scales and then aggregated so that the next stage can abstract features from the different scales simultaneously. The improved use of computational resources allows for increasing both the width of each stage as well as the number of stages without getting into computational difficulties. One can utilize the Inception architecture to create slightly inferior, but computationally cheaper versions of it.



**Figure 1.2 Inception Module**

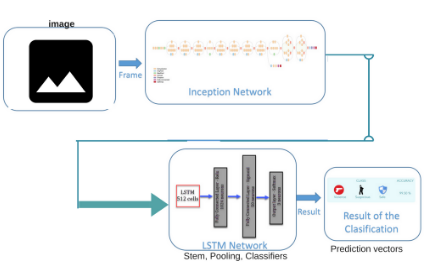
**2. Image Recognition using GoogLeNet in Keras**

The project simulation is done based on existing research work on “Going Deeper with Convolutions”. The research is about implementation of deeper and efficient neural network called ‘Inception’. Inception module is network inside a network. GoogLeNet architecture is implemented using Inception network. That’s why GoogLeNet has won Image Net Challenge and attained less error rate of about 6% only. Now, all that matters is the accuracy. Hence, to simulate the proposed results lively, we implemented GoogLeNet architecture in Keras model to predict the outputs as image class label name and the deep features it contains.

The simulations are carried out with overall structure of the model which has few basic sections: the stem, the inception modules, the auxiliary classifiers, and finally the output classifier. GoogLeNet contains nine of these modules, sequentially stacked, with two max pooling layers along the way to reduce the spatial dimensions. Due to the depth of this architecture, the authors added two auxiliary classifiers branching from the main network structure. The purpose of these classifiers is to amplify the gradient signal back through the network, attempting to improve the earlier representations of the data. Finally, we get to the output classifier, which performs an average pooling operation followed by a softmax activation on a fully connected layer.

In total, the network uses the standard operations: convolution, pooling, normalization, and fully-connected layers.

The architecture of project (Figure 2.1) typically depicts the way the simulation happens. First the image is inputted to the model, immediately the image dimensions are reduced to a fixed dimension to maintain uniformity, given any image type. Next the image is passed down to layers of convolutional networks which act as network inside a network i.e, Inception model. Filters will be applied on the input image while passing through these convolutional layers which calculates weights. Next, all these weights are pooled up through Long Short-Term Memory (LSTM) Network layer. Finally the weight as a single value is outputted to GoogLeNet image labels. This single-weight value is matched with its respective class label of 1000 pre-trained class labels in GoogLeNet. Then, the class name of the input image is given as output. Finally, the image is recognised and it’s deep features are predicted.



**Figure 2.1 Architecture followed for simulations**

**3. Conclusions**

The image recognition is done using Inception model, where the image is passed over layers of convolutional neural nets and also been applied deeper filters to view the deepest features of the image. An input image is taken and on filters like 1x1, 3x3, 5x5 the deep feature extraction is done. The output displayed is based on the class label of which input image belong to. Hence, the output of the simulation is image’s class type out of 1000 classes which are pre-trained in GoogLeNet, along with the deep features.

Scope of study can be enlarged by considering more complex images i.e, image with multiple objects. This code can be integrated with Arduino microprocessor for real time applications like training robots for image recognition. In our project we only simulated the results of a research paper and inception model is implemented for extraction of deeper features. Our simulations are basis of ground truth that pre-trained set of 1000 classes of images i.e, GoogLeNet has highest accuracy in recognition of images. Further the layers of inception model can be increased to attain highest level of accuracy which can be done by changing the GoogLeNet architecture.

**References**

[1] Szegedy Christian, Wei Liu. IEEE Reference, Available:<http://ieeexplore.ieee.org/document/7298594/>

[2] Going Deeper with convolutions. <https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf>

[3] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In CVPR, 2014.

[4] Gist link to Joelouis’s experiment using GoogLeNet in Keras <http://joelouismarino.github.io/blog_posts/blog_googlenet_keras.html>