**ABSTRACT**

**Image recognition** is one of the hallmark tasks of computer vision, allowing the definition of a context for object recognition.  A deep convolutional neural network architecture codenamed as **‘Inception’** is the one that achieves the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition (ILSVR). One particular incarnation of ILSVR is called **‘GoogLeNet’**, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.

**‘We Need To Go Deeper’** This phrase can indicate that further investigation is needed with image processing today and it results in going beyond the two-dimensional and going deeper to see what is actually in the image. It is the real computer vision. For such a deeper vision, a deep learning library called **‘Keras’** which is a python library that runs on top of Theano is implemented. **‘Theano’** is a Python library that allows defining, optimizing, and evaluating mathematical expressions involving multi-dimensional arrays efficiently which bolsters image recognition accuracy. GoogLeNet was one of the first models that introduced the idea that **Convolution Neural Network** (CNN) layers did not always have to be stacked up sequentially. Coming up with the Inception module, it is a creative structuring of layers that can lead to improved performance and computationally efficiency in recognition of images. Thus the concepts of GoogLeNet in Keras will set the computational intelligence stage for some amazing architectures that could advance the fields of computer vision and image recognition.

**LIST OF FIGURES**

**S.No. Fig No. Fig Name Page No.**

1. 1.1 GoogLeNet in Keras 2
2. 1.2 Stem Architecture 3
3. 1.3 Depth concatenation Architecture 3
4. 1.4 Output classifier 3
5. 2.1 Configuration of Theano as backend 9
6. 2.2 Project Architecture 11
7. 3.1 ConvNet internal layer Architecture 14
8. 3.2 Convolution network flow 14
9. 3.3 Max pooling of spatial dimensions 17
10. 3.4 Architecture of a neuron 18
11. 3.5 1D convolutions 19
12. 3.6 General pooling 21
13. 3.7 LSTM network architecture 25
14. 3.8 Inception model architecture 29
15. 4.1 Running the project file 80
16. 4.2 Running on theano as backend 80
17. 4.3 Output I 81
18. 4.4 Output II 81
19. 4.5 Output III 81
20. 4.6 Output IV 81
21. 5.1 Unit testing life cycle 83

**LIST OF TABLES**

**S.No. Table Page No.**

1. Software requirement specifications I 12
2. Software requirement specifications II 12
3. GoogLeNet Layers 31
4. Unit test cases I 84
5. Unit test cases II 84