Envisager- Sight For Blind And Visually Impaired

Nandanamudi Jyothikiran, Gatiganti Venkatesh, Nallmothu Naveena, Gumma Madhuri

***Abstract*— Image Recognition and the image classification has been one of the weighing gravitate towards computer vision and has been lagging in certainty. The field of Machine Learning has made a tremendous progress in object detection with a refined quality. With the usage of model like SVM, KNN, Bag of words we have improved our classification precision but still sparring with the amount of accuracy until deep convolutional neural networks came in existence with which we can exceed human performance in some domains. Here in this paper we introduced Tensorflow based model which uses deep convolutional network of 22 layers deep network which is trained with dataset of ILSVRC 2014 with a thousand classes to classify the image. We built an android application that can dynamically recognize and classify the images on the screen with an intuition of multi-scale processing.**

Keywords— Deep Neural Network, Tensorflow, ImageNet (ILSRVC 2014).

I. INTRODUCTION

In the recent years, with the advancement of deep learning the quality of image recognition and object detection made easier and has been improving at a dramatic pace [4]. We built an android application which uses Tensoflow open source library in which deep neural networks is built with the union of Softmax (multinomial logistic regression) to handle multiple classes and is named as Envisager. The softmax is used in the final layers of the deep neural networks.

With the ongoing friction between mobile and the level of computation it can perform the efficiency of algorithm is very important which determined the usage of power and memory. In this paper we focused on building an efficient deep convolutional neural network architecture[7] with the resource provided by the Tensorflow community in conjunction with the model proposed during ILSRVRC14 named inception.

In culmination [2], we demonstrated an android application which can identify and classify an image at an accuracy of human even exceeding it sometimes particularly in some domains.

II. RELATED WORK

A. Convolution Neural Networks

The typical architecture of convolutional neural networks is the union of convolutional layers(with normalization and max pooling layers followed by one or more fully connected networks. Often these architectures proved to give best results for various datasets such as MNIST, CIFAR and even ImageNet classification challenge [3, 10].

Inspired by the primary visual cortex [9] of the neuroscience model, to handle multiple scales filters of different sizes were used [9] similar to inception model [1] and in the Envisager (proposed model). Similar to inception model all the filters in all the layers are learned and theses inception layers were repeated many times resulting in a 22-layer model.

In the object detection, we used a model proposed by Girshick et al. [5] but instead used multi box prediction rather than single bounding box.

B. Motivation

In this hustle and bustle of modern life, even people with no disabilities find many things as a hindrance for completing their activities and daily routines. However, people who are visually disabled are facing many difficulties than the normal people without any disability. As we know, the social constructs are not always designed by keeping them people in mind. They do need support in crossing roads, detecting the object nearby and other obstacles. Even though they use canes, they cannot identify objects above their waists, which is why there is a need a smart way of identifying the object they come across.

However, with an android app like ours, people with smartphone can have it and we could help most of them to perform their daily activities without running into any trouble. Our main goal is to develop an application which would allow the blind people to take

the pictures through camera and then our system would detect the image captured and identify the object and give them the audio reply describing the object or the naming the object etc., Moreover, all the features can be used without spending a dime and all they required is to install the application.

III. PROPOSED PLAN

A. Architecturural Details

Our goal is to find the optimal local sparse network that can be integrated and should operate at a minimum level of computation on the mobiles. The method propsed by the Arora et al. [2] a layer by layer construction in which the correlation statistics of last layer is analyzed and clustered into unit groups of high correlation. The clusters of all layers are connected one to one forming a chain and the units received by the next layers is a part of image sent from previous layer. The local regions will be concentrated by the correlated units after diving the units into filter banks. The current incarcerations were restricted to filters sizes of 1x1, 3x3 and 5x5 like inception model to avoid patching alignment issues. Thus, this results in lots of clusters concatenated in a single region which is avoided by using 1x1 convolutions in the next layers.

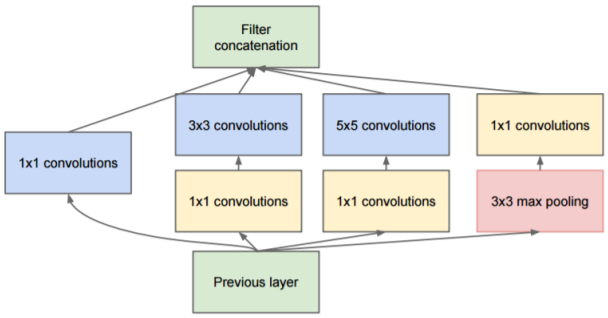

Fig 1 Inception module with dimension reduction

The network consists of modules stacked upon each other. For memory efficiency the stacked modules were used in the higher layers and the traditional convolutional in the lower layers. By using dimension reduction, it shields the large number of input filters in the last stage send to next layer, first reducing their dimension before convolving over them with a large patch size.

B. Envisager Model

We used a wider and deeper network with the addition of ensemble the results has been improved even though quality is inferior to the previous proposed models. The Envisager Incarcination of inception architecture is shown in the below table.

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Table 1. Envisager Incarcination of inception architecture

Recified Linear Activation is used in all the convolution even inside in the inception module. The receptive field of 224 x 224 is considered by taking RGB color channels in mean subtraction. “#3×3 reduce” and “#5×5 reduce” stands for the number of 1×1 filters in the reduction layer used before the 3×3 and 5×5 convolutions.

We build this application keeping computation efficiency and practicality in mind. There are 27 layers in total considering 5 pooling layers and remaining layers that contain parameters. If we consider individual building blocks there are almost 100 layers in total. The pooling improved the accuracy to almost 0.62%.

Due to depth of the network the backpropagation is a concern and there is discrimination in terms of features produced in the middle layers. So, by adding the auxiliary classifiers in the intermediate layers discrimination is expected in the lower layers as well and the gradient signal generated is strong which can be propagated back which results in regularization. The classifiers adds a lot of output during training time as it adds on to the total loss. So, during that time the auxiliary networks were neglected.

The Fig 2 shows the schematic view of deep neural network for our approach with

- A Convolution layer of size 1x1 with 128 filters for dimension reduction and rectified linear activation.
- Fully connected layer of 1024 units and rectified linear activations.
- For dropped outputs with 70% ratio in dropout layer.
- Softmax loss as the classifier in the linear layer.

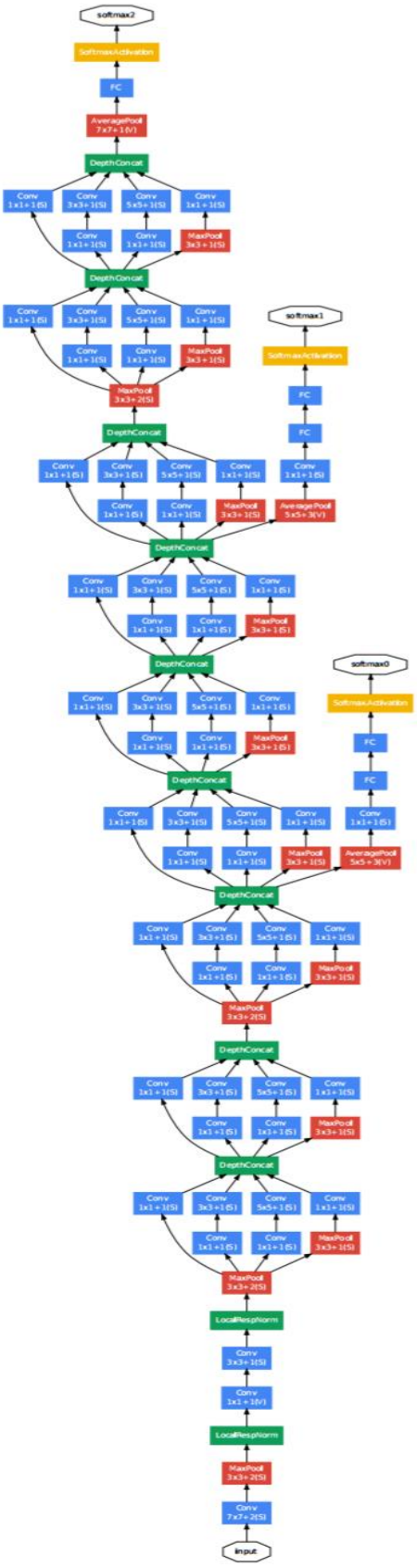

Fig 2. Schematic view of deep neural network

IV. IMPLEMENTATION

A. *Methodology of Training.*

The Inception model is used to classify the camera real frames in real time. We are displaying the top results on the image at the top. We used Bazel is fully supported by the tensorflow we preferred using in our approach to train and build.

Bazel is a build tool provided by Google which has advanced caching and parallel execution and has high scalability and is very flexible to operate.

As we developed the native application Android NDK is required to build it with a version of 12b and higher and Android SDK of with build tools for API level of above 21 is preferred. We used CPU based implementation here. The training module is of fixed learning rate schedule. During inference time, Polyak averaging [8] is used to create the final model.

We used the model generated using ImageNet data at 2015 ILSVRC14 competition which has 1000 classes at the output layer of the final layers of the softmax.

Type	**URL**
Github	https://github.com/nandanamudi/Big-Data-Analytics-and-Application---Envisager

Table 3 Github Repository and Youtube Video URL.

B. *Activity Diagram.*

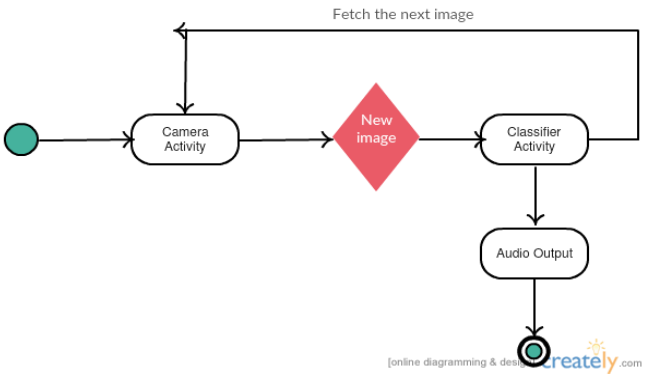

Fig 3. Activity diagram of Inception model

Live feed of an image frame is sent to the classifier using the camera activity and the classifier activity uses the trained model to classify the next image while doing so next image is fed to the classifier to classify and this is a continuation process and is dynamic.

V. RESULTS AND EVALUATION

A. *Evaluation plan.*

a. Datasets

ImageNet dataset has been one of the popular datasets with thousands of images in each node with hierarchy. Even though there are only nouns available for now, there is almost thousand images per node. The database is organized according the hierarchy used by the WordNet.

b. System Specifications

12 GB Ram, Intel 5000 CPU, Tensorflow Version 1.0, Android Studio 2.3, Gradle version3.3, Pycharm.

B. *Evaluation & Results.*

1. Previous Models methodologies

a. Clarify API:

Clarifai API accepts a video, image or an audio as an input from the Android application and it will break down the media to analyze it and to use the information to configure and extract the words the media may contain. The API uses machine learning techniques which would improve with time and it gives the output to perform any further actions.

b. Apache Spark API:

Apache Spark is an open source platform for scalable MapReduce computing on clusters. The main goal of the Spark framework is speeding up Image classification by parallelizing the machine learning to a high-performance computing cluster. We integrate open source SDK library to Spark API. To classify the image/video which is taken as the input from Android APP we use random forest classification algorithm in Spark.

c. Comparative Results:

	Clarify API	Spark API	Inception
Accuracy	99% (Trained classes)	31%	92%
Computational time	Low	High	Low
Memory Usage	Low	High	Medium

Table 4. Comparitive Results between clarify, Spark api and Inception

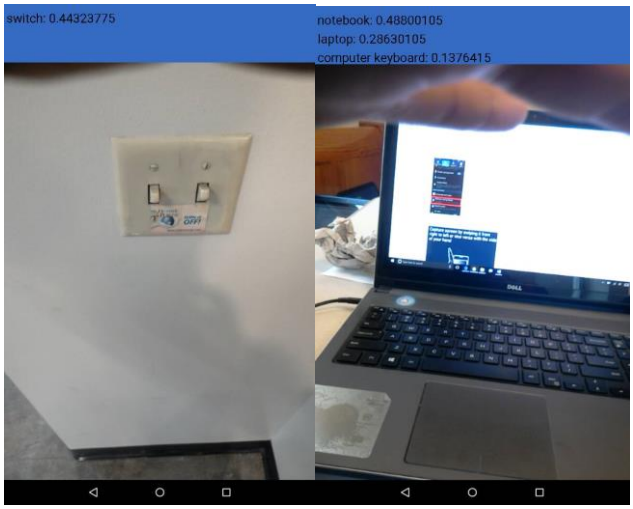

Fig 4: Test Images using android application

VI. LIMITATIONS

Since, we intend to develop an android application for this model, the tensorflow doesn't support gradle completely so we are forced to used bazel and we have used a CPU processor to develop this model which took a long time to do so.

Since the camera2 api supports only to API level above 21 and further we are unable to provide service to lower levels.

VII. CONCLUSION

Our results to yield gave a solid proof that by using densely building blocks of inception modules for approximating the expected optimal sparse structure for improving the deep neural networks for computer vision. The main advantage is computational requirements required to run this application. The response time is very low when compared to other developed comparative models.

With the usage of more dense depth and width network accuracy can be improved but it leads to the expensive methodology.

VIII. FUTURE WORK

We intend to integrate Google Maps with the application taken which would be helpful while crossing the roads, navigation etc.,

We also intend to improve the accuracy further considering all the possible changes.

IX. REFERENCES

[1] Wei Liu, Yangqing Jia, Christian Szegedy, Pierre Sermanet, Scott Reed, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. Dragomir Anguelov, Going Deeper with convolutions. arXiv:1409.4842v1 [cs.CV] 17 Sep 2014

[2] Sanjeev Arora, and Aditya Bhaskara, Rong Ge, Tengyu Ma. Provable bounds for learning some deep representations. CoRR, abs/1310.6343, 2013.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. In Advances in Neural Information Processing Systems 25, pages 1106–1114, 2012. Imagenet classification with deep convolutional neural networks.

[4] J. S. Denker, D. Henderson, W. Hubbard, and L. D. Jackel. Y. LeCun, B. Boser, R. E. Howard, Backpropagation applied to handwritten zip code recognition. Neural Comput., 1(4):541–551, December 1989.

[5] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate semantic segmentation and object detection. In Computer Vision and Pattern Recognition, 2014. CVPR 2014. IEEE Conference on, 2014.

[7] Qiang Chen, Min Lin, and Shuicheng Yan. Network in network. CoRR, abs/1312.4400, 2013.

[8] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. SIAM J. Control Optim., 30(4):838–855, July 1992.

[9] M. Bileschi, Maximilian Riesenhuber, Thomas Serre, Lior Wolf, Stanley and Tomaso Poggio. Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Intell., 29(3):411–426, 2007.

[10] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David J. Fleet, Bernt Schiele, Tinne Tuytelaars, Tomas Pajdla, and editors, ´ Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I, volume 8689.