RSU JOURNAL OF BIOLOGY

AND

APPLIED SCIENCES

ISSN: 2811 – 1451

RSUJBAS

ABOUT US

Rivers State University Journal of Biology and Applied Science (RSUJBAS) publications is a quarterly, open access, international journal for all academic research in science discipline. Microbiology, botany, zoology, environmental biology, chemistry, physics, mathematics, computer science, biochemistry medical laboratory sciences and other applied science related areas. RSUJBAS is a platform set for elites to influence, contribute and communicate to the global environment through their various academic researches. We synergistically engage our noble effort to contribute to the knowledge development, discoveries and innovations in all fields of study. In RSUJBAS we publish research papers on current academic issues with standard scientific reviews. RSUJBAS publishes original research articles, review articles, case studies, short communications, survey report, comparative studies and many more.

Aims and Scope

Rivers state University Journal of Biology and Applied Sciences aims to publish high quality papers that communicate fundamentals and contemporary discoveries both theoretical and practical. Most importantly RSUJBAS seeks to establish a platform for communicating emerging trends in various discipline such as Microbiology, Botany, Zoology, Environmental Biology, Chemistry, physics, Mathematics, Computer Sciences, Biochemistry, Medical Laboratory, Sciences, and other applied sciences related areas.

Description:

- Area of concentration: All science academic disciplines
- Frequency of publishing: Quarterly
- Mode of publishing: both online and print publication
- Language of publication: English
- Double blinded Review Process
- Zero Level Plagiarism Tolerance

Why Publish with us

Low Article Processing Charge (ACP) to promote the research work
Easy and Rapid review process
Instant publication upon acceptance
Dedicated editorial and review team for fast review process
RSUJBAS provides hard copies of publication every quarterly

EDITORIAL BOARD

All Correspondence to
Prof Sam Wenedu (Editor -in -Chief)
Department of Microbiology, Rivers State University
Edictor.ibasya@yoo.com

Or

OLUCHI DICKSON
Publication Manager
Dicksonoluchi87@gmail.com

CONSULTING EDITORS

GUIDELINE FOR MANUSCRIPTS

Manuscripts should be typewritten on an A4sheet having B1.5=line spacing throughout the text. The margins should be 2B54cm (1 inch) in all sides and page number should be consecutively on the bottom of the page. The manuscript should be written in Times New Romans using ‘12’ font size.

For original research paper, the manuscript should be arranged in the following order: Tittle page, Abstract, Keywords Introduction, Materials and Methods Results, Discussion, Acknowledgement, References, Tables with legends and supplementary materials

The tittle page should contain the title, the name(s) of the author(s), the name(s) and address (es) of the instruction(s) where the work was carried out, including a valid e-mail address from the corresponding author along with telephone numbers. The title of the manuscript should be specific and concise but sufficiently informative.

The Abstract should not exceed 250 words and it should contain brief summary of the findings including brief introduction, methodology, results, and conclusions,

The keywords should have a minimum of five and maximum of seven words.

The introduction should provide a clear statement of the problem and indicates aim of the study citing relevant literature to support background statements.

The Materials and Method should include the methods and methodology of the research.

The results should be presented in the form of tables of figures. It should be presented with clarity and precision. Statements used to present results should be written in the past tense. Detailed interpretation of data should not be included in the results but should be put into the Discussion section.

The Discussion should interpret the results clearly and concisely, and should integrate the research findings of this and past studies on the topic. Highlight the significant/unique findings of the research under conclusion.

The acknowledgment of people, grants or funds should be brief.

TABLE OF CONTENTS

LIST OF CONTRIBUTORS

Naluba Nwiekpigi Goddy
Department of Geography and Environmental Studies
Faculty of Social Sciences
Ignatius Ajuru University of Education
P.M.B 5047, Rumuolumeni, Port Harcourt.
Email: naluba.goddy@iaue.edu.ng

Ephraim Esuu-Ikup Amon
Department of Geography and Environmental Studies
Faculty of Social Sciences
Ignatius Ajuru University of Education
P.M.B 5047, Rumuolumeni, Port Harcourt.

Kpeno Amon Ephraim
Department of Adult Education and Community Development
Faculty of Education
Ignatius Ajuru University of Education
P.M.B 5047, Rumuolumeni, Port Harcourt.
Email: kpenoephraim@gmail.com

Ugwoke, F. N
Department of Computer Science,
Michael Okpara University of Agriculture, Umudike
ndidi.ugwoke@gmail.com

Etuk, E.A
Department of Computer Science,
Michael Okpara University of Agriculture, Umudike
etuk.enefiok@mouau.edu.ng

Iroegbu C.
Department of Electrical and Electronics Engineering,
Michael Okpara University of Agriculture, Umudike
iroegbu.chibuisi@mouau.edu.ng

Nwabueze, S.I.
Department of Computer Science,
Renaissance University,
Ugbawka, Enugu.

Mc-Kelly Tamunotena Pepple
Federal Polytechnic of Oil and Gas Bonny,
Department of Electrical/Electronic Engineering Technology
mckelly2014@gmail.com, 07030335027

Efiyeseimokumo Sample Ikeremo
University of Africa, Toru-Orua,
Faculty of Basic and Applied Science,
Department of Computer Science
samplefiye@gmail.com

Asogwa S.C.,
Department of Computer Sc.,
Michael Okpara University of Agriculture,
Umudike, Nigeria
Asogwa.samuel@mouau.edu.ng

Okorie Nnanna Kalu
Department of Computer Sc.,
Michael Okpara University of Agriculture,
Umudike, Nigeria

COMPARISON OF GOOGLE NET CONVOLUTIONAL NEURAL NETWORK TO VISUAL GEOMETRY GROUP CONVOLUTIONAL NEURAL NETWORK

MC-KELLY TAMUNOTENA PEPPLE
Federal Polytechnic of Oil and Gas Bonny, Department of Electrical/Electronic Engineering Technology
mckelly2014@gmail.com, 07030335027

&

EFIYESEIMOKUMO SAMPLE IKEREMO
University of Africa, Toru-Orua, Faculty of Basic and Applied Science, Department of Computer Science
samplefiye@gmail.com
Email: g.onuodu@gmail.com

ABSTRACT
Human being has always had the dream of making computer operate with human intelligence in other to solve problem. But it has always been a difficult task to train a system to operate with human intelligence. But AlexNet was able to achieve an impressive result in the ImageNet Large Scale Visual Recognition Challenge using convolutional neural network, which motivated researchers in the development of more improved and robust models and led to the development of Google Net, Visual Geometry Group and other models. In this paper, Google Net which is based on the enhancement of the inception model is compared to Visual Geometry Group which is a very deep convolution in terms of architectural design and performance efficiency. A single inception model of Google Net and a single block of Visual Geometry Group were also implemented to show the differences in their architectural design and mode of operation. The comparison shows that Google Net has a simpler architecture and took lesser time to train. It also outperformed VGG in the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with a top-5 error of just 6.67% compared to 7.32% of Visual Geometry Group (VGG).

Key words: *Neural Network, Convolution layer, Max pooling, Average pooling Complex cell (C-cell), Simple cell (S-cell), Visual Geometry Group (VGG), Google Net, ImageNet Large Scale Visual Recognition Challenge (ILSVRC)*

INTRODUCTION

Human being is able to analyze intricate design easily. For instance, a young child is able to recognize a lot of faces and objects around them in spite of the random motion of the object and the distance of the object from them. But the greatest challenge over the years is to train a computer to analyze and easily recognize intricate and dynamic designs of objects around them. This difficulty has compelled scientist to design algorithms which can solve a specific problem but is not able to solve a similar problem with little modifications. This makes it difficult to produce a robust system that can replicate the function of the brain and recognize modifications in complex systems.

Blakemore and Cooper (1970) came up with an experiment where they trained young domestic cat in a simulated environment that only has black and white stripes placed horizontally as a result, the cats were not able to sense vertical stripes which proved that mammals develop their ability to recognize things from experience and training because the regular cats can recognize both vertical and horizontal stripes. Hence, the focus is to train computers to be adaptive like the human brain.

Hubel and Wiesel (1959) explained simple cells and complex cells in the human optical cerebral cortex. They postulated that both type of cells is utilized in pattern detection. They declared that a simple cell responds to edges and bars of a specific pattern of an image and that a complex cell does the same thing and also responds when the edges and bars are shifted. Hence, it the disparity between a complex cell and a simple cell is that simple cell only detects an edge in a specific location while complex cells detect multiple edges in different locations.

Hubel and Wiesel (1962) postulated that complex cells accomplished spatial invariance by combining the results of many simple cells with the same orientation (e.g vertical bars) but different receptive fields (e.g different vertical locations on an object). Hence, this ability enables the complex cell to collect information on any part of an object.

This gave birth to the idea of summing simple detectors to create an advanced detector that is able to detect complex object. This is the foundation through which the convolutional neural network was built on.

Fukushima (1975) design an algorithm to simulate the adaptive nature of the human brain which was impressive in its patten recognition abilities. The mathematical model is a multilayer network without supervised training or desired training output pattern, hence, only the most potent firing neurons are trained, this was likened to elite education. The architecture of the cognitron is shown in Figure 1, it depicts a network of neural layers of like structure stacked one after another. The l-th layer U_l is made up of neurons $u_l(n)$ which can be excited and repressive neurons $v_l(n)$, where n = (nx, ny) represents 2-dimentional co-ordinates which indicates the position of a cell.

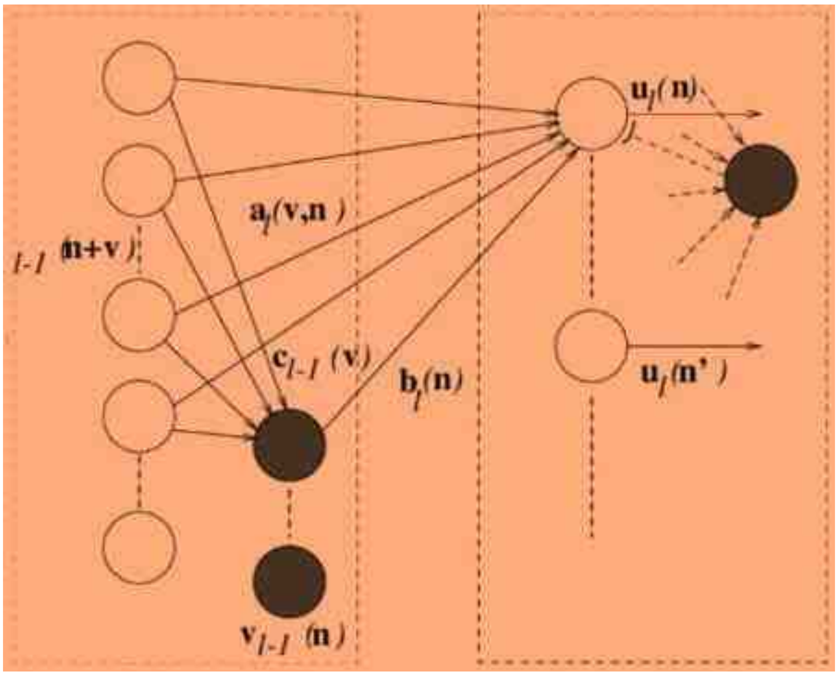

Figure 1 The Architecture of cognitron

Fukushima (1984, 1986, 1987) developed the neocognitron which is also a multilayer network but similar to the visual system model developed by Hubel and Wiesel (1962, 1965, 1977). The neocognitron improved on its pattern recognition capabilities when compared to the cognitron. The neocognitron is made up of layers and each layer is made up of two planes of cell arrays which react to like pattern.

The architecture of the neocognitrone is shown in Figure 2. It utilizes the extraction of feature, pooling layers and convolution neural network towards identification of image. The architecture of the neocognitron was based on optical neural network of a vertebrate. It consists of alternate layers of simple cells (S-cells) and complex cells (C-cells). It operates by the continuous extraction of feature by the S-cells and adjustment to positional displacement by C-cells. The process incorporates local features extracted at lower levels into more global features. It was applied in handwritten letters detection and other pattern detection projects. This algorithm introduced the study of convolution neural networks (CNN).

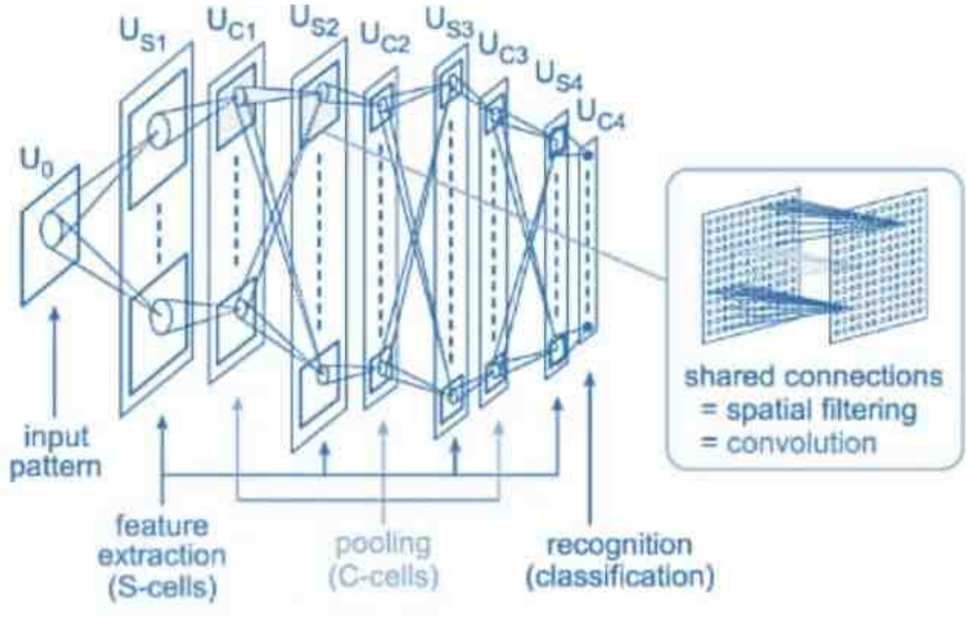

Figure 2 The architecture of Neocognitron

Le Cun et al (1989 to 1998) used a Convolutional Neural Network (CNN) which is a Deep learning algorithmic program which is used to identify and recognize images. The design is modelled like the operation of the neurons in the human brain which uses back propagation algorithm which was which was tested on its capabilities to recognize characters which were hand-written and those printed by machines and it gave an impressive result. It is made up of layers which consists of parameters that can learn. It also consists of sets of convolution layers with combining average pooling layers. The number of parameters which can be trained is 60000 which is a massive improvement when compared to the neocognitron which does not train parameters.

Figure 3 depicts the architecture of the LeNet-5 CNN which consists of 7 layers which is made up of 2 sets of convolutional layers, 2 sets of average pooling (subsampling) layers, a convolution layer and 2 fully connected layers.

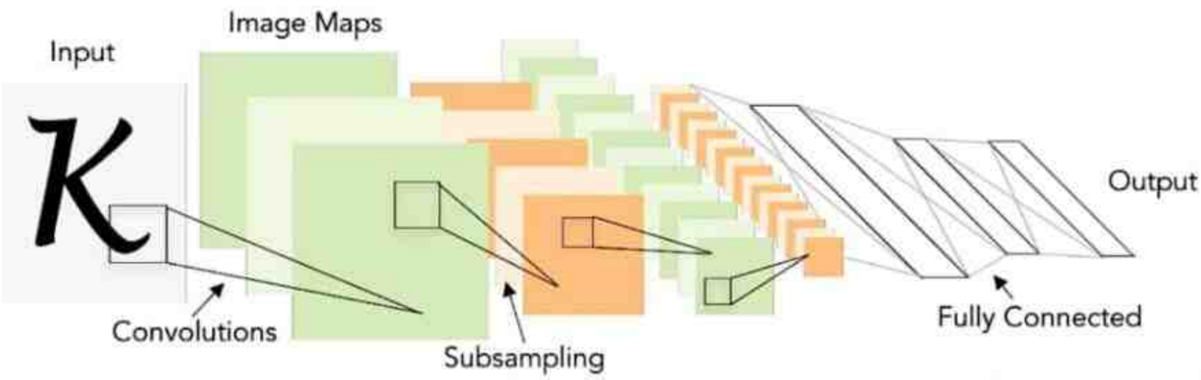

Figure 3 The architecture of LeNet-5

AlexNet is a convolutional neural network algorithm that was developed by Krishevsky et al (2012) in his research on the paper called Imagenet Classification with Deep Convolutional Neural Networks. AlexNet which was the first algorithm to win the ImageNet Large Scale Visual Recognition Challenge by achieving a top 5 error of 15.3% in 2012, which is 10.8% lower than the second best. This was based on the model's ability to categories object photographs into 1000 distinguished categories. Prior to AlexNet, categorizing objects at this level by computer vision was seen to be very difficult to accomplish, but the great success achieved by AlexNet motivated the development of other enhanced models using Convolutional Neutral Network. AlexNet demonstrated that deep and robust models can be designed to solve complex problems through supervised pre-training techniques.

The significance in the architecture of AlexNet is the application of linear activation function (ReLU) to account for the nonlinearity within each convolution layer, rather than the S-shaped logistic functions or Tanh. The output layer was activated by a softmax function.

AlexNet applied average pooling method instead of the max pooling method applied in LeNet-5

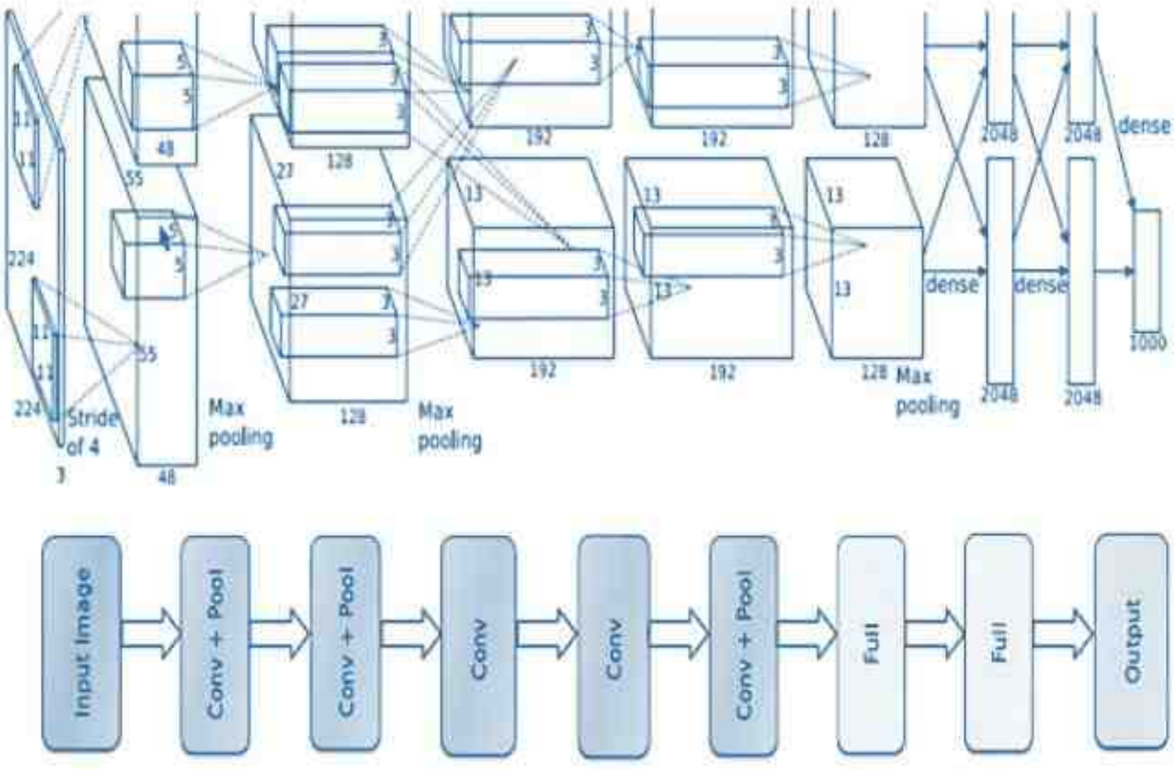

Figure 3 The architecture of AlexNet

The architecture of the AlexNet is shown in Figure 3, the network has greater depth when compared to that of LeNet-5. AlexNet consists of eight layers with trainable parameters, five of these layers has a combing max pooling and the other 3 layers are fully connected. Apart from the output layer, each layer is activated by Relu. It was discovered that relu activated layers speed up the training, making it six times faster. It has about 61 million parameters compared to LeNet-5 that have just 60 thousand parameters.

Mathew D. Zeiler and Rob Fergus developed the ZFNet algorithm which is an enhanced modification of the AlexNet with a greater accuracy level. The greatest disparity of the ZFNet from the AlexNet is that it utilized 7x7 sized filters while 11x11 filters were utilized by AlexNet. The smaller filters were used to reduce pixel details lost as a result of using big filters, since smaller filters are able to retain more pixel information.

ZFNet which is also a 2013 winner of the ILSVRC competition, is implemented using a convolutional neural network algorithm. It was able to achieve a top 5 error of 14.8%. This was mainly done by modifying the hyper parameter of AlexNet and retaining its structure

Figure 4 depicts the architecture of the ZFNet which consists of 8-layer convolutional network model with an input image of 224 x 224 which is convolved using 96 distinct first layers of 7 x 7 dimension with a stride of 2, generating a feature map. The map goes through a linear function to rectify it.

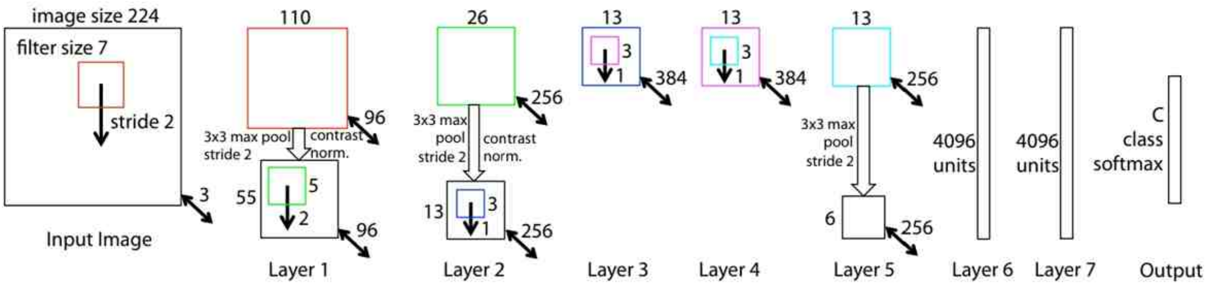

Figure 4 The architecture of ZFNet

Literature Review

Visual Geometry Group

Jangra et al, 2020 in their paper titled ECG arrhythmia classification using modified visual geometry group network, postulated an enhanced convolutional neural network. The network applied automatic arrhythmia classification with Electro-Cardio-Gram (ECG) signal. The paper was aimed at monitoring the heart beat to detect irregularities in the rate of heart beat which are usually caused by cardiovascular diseases, hence, reducing death caused by those deceases. As, a result of the disparity in dimension between ECG and image signal, the author studied how the reduction in the depth and width of the convolutional neural network with respect to cardiac arrythmia classification. Benchmark MIT-BIH database was used to study six configurations with different depth and width. An impressive performance was achieved by a deep network, consisting of 13 convolutional layers, this improved network was called modified VGGNet (mVGGNet). The mVGGNet network attained 98.79% and 99.16% precision when applied in ventricular ectopic beats (VEB) and super-ventricular ectopic beats (SVEB)

Visual Geometry Group-UNet (VGG-UNet) deep learning network was proposed by Mei et al, 2021 in the paper titled Visual Geometry Group-UNet: deep learning ultrasonic image reconstruction of curved parts, aimed at solving the difficulty in discovering the little flaws in curved parts, by improving the image resolution. The study was based on a robot-assisted ultrasonic testing system using the track-scan imaging method. VGG is applied to retrieve advanced information from ultrasonic images and UNet is used for little dataset categorization. It was observed from the reconstructed images on the simulation dataset based on ground truth comparison that peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM) can get up to 39 dB and 0.99, respectively. Also, the trained network is robust enough to resist noise and other environmental factors, based on the results of their experiment. The resolution of ultrasonic images reconstruction based on track-scan imaging is increased by about 10 times.

Xu et al, 2021 studied the continuous change prediction of PM 2.5 concentration in an air contamination research centre in their book titled A Feature Extraction and Classification Method to Forecast the PM2. 5 Variation Trend Using Candlestick and Visual Geometry Group Model. They merged physical methods and deep learning models by segmenting the contamination process of PM 2.5 into effective multiple types as this is essential to obtain a authentic prediction of the PM 2.5 value. A candlestick chart sample generator was designed to enable the generation of candlestick chart from the online PM 2.5 continuous monitoring data of the Guilin monitoring station site. Gaussian diffusion model was applied to the generated candlestick chart to analyse. It was observed that the characteristics of the physical transmission process of PM 2.5 pollutants can be reflected. Applying the time liner convolution method within a three days period, within a three days' time interval, 2188 sets of candlestick chart data were derived using the 2013 – 2018 PM 2.5 concentration data. There are 16 classifications produced by unsupervised classification based on the standard classification judgement. It was established that the accuracy rate of the change trend of the classification got to 99.68% within the next period. Utilizing the candlestick chart data as the dataset for training, an enhanced Visual Geometry Group was applied. The overall accuracy based on the experimental result indicates that the value of the candlestick chart combination classification was 96.19%, and the kappa coefficient was 0.960. In the VGG model, the overall accuracy was enhanced by 1.93% when compared with support vector machines (SVM), LeNet, and AlexNet models. It was also established that the VGG model was able to keep the characteristics of the physical pollution process and give the classification basis required to accurately predict PM 2.5 values.

Raja J., Shanmugam P. and Pitchal R. in 2021, in the paper titled, An automatic early detection of glaucoma using support vector machine based visual geometry group 19 (vgg-19) convolutional neural network, studied the application of deep learning technique in the investigation of medicinal images. The aim is on the early detection of glaucoma which is neurotic condition that results to dynamic neuro degeneration of the optic nerve, which leads to visual impairment. This paper applies novel mechanized glaucoma recognition carried out with computer aided analysis from fundus images. A machine based VGG-19 network architecture was used to perform the simulation. CDR threshold value 0f 0.41 was utilized for glaucoma recognition where greater than 0.41 indicates glaucoma affected and less than 0.41 indicates non-glaucoma fundus images. It was established that a classification precision of 94% was achieved using a set of 175 fundus images.

Google Net

Al-Qizwini et al (2017) in their paper titled, Deep learning algorithm for autonomous driving using Google Net, considered direct perception approach for autonomous driving algorithm and its performance under realistic assumptions. A new more robust and realistic Direct Perception Framework and corresponding algorithm for autonomous driving were introduced. In the feature extraction test that was carried out, Google Net performed better when compared to two other top convolutional neural networks. Hence, a deep learning-based algorithm for autonomous driving was proposed and referred to as Google Net for Autonomous Driving (GLAD). Unlike previous research done on autonomous vehicle, GLAD only makes realistic assumptions about the autonomous vehicle and its environment and utilizes only five affordable parameters in the navigation of the vehicle, compared to the 14 parameters utilized by its predecessors. It was established based on the simulation result that the proposed Glad algorithm outperforms previous Direct Perception algorithm both on empty road and while driving with other vehicles.

Khan R. U., Zhang X. and Kumar R. (2019) used two different models to identify the obscure or new sort of malware in their paper titled, Analysis of ResNet and Google Net models for malware detection. The two different models are ResNet which is of the Microsoft platform and Google Net which is the intelligent property of Google Net. As a result of this disparity, two different datasets were applied for training and validation of the models. Hence, a combination of 10,868 binary records dataset was gotten from Microsoft. These records are further divided in nine different classes. The second dataset is considerate dataset and have 3000 benign files. The datasets were converted from EXE files into opcode and then to images. A testing accuracy of 74.5% was achieved for Google Net and 88.36% precision for ResNet.

Zhong Z., and Jin L. and Xie Z. (2015) proposed a deeper architecture to improve on the handwritten Chinese character recognition (HCCR) convolutional neural network with fewer parameters. They also demonstrated that the conventional feature extraction methods, like Gabor or gradient feature maps, are still be used to improve the performance of convolutional neural network. They designed a streamlined version of Google Net [13], which was originally proposed for image classification in recent years with very deep architecture for HCCR and was denoted as HCCR-Google Net. The HCCR-Google Net was applied in 19 layers deep but consists of just 7.26 million parameters. Experiments were carried out with the ICDAR 2013 offline HCCR competition dataset. It was demonstrated that with proper incorporation with conventional directional feature maps, the postulated single and ensemble HCCR-Google Net models attained an impressive accuracy of 96.35% and 96.74% respectively, greatly outperforming its predecessors.

Tang P., Wang H. and Kwong S. (2017) studied the used of convolutional neural network in scene recognition. As a result of the short coming of poor representation ability of the conventional handcrafted features, convolutional neural network is used to enhance scene recognition. The convolutional neural network contains more semantic and structure information and hence, have more categorizing capabilities through multiple linear and non-linear transformations. In their paper, Google Net model is applied and segmented into three layers from bottom to top. The output from each part was used to scene recognition which resulted in the proposed Google Net based multi-stage feature fusion (G-MS2F). The product rule is then applied to get the last decision for scene recognition from triple outputs corresponding to the three parts of the model. The test results show that the proposed model is better than some of the state-of-the-art CNN models for scene recognition, and achieved the recognition precision of 92.90%, 79.63% and 64.06% on the benchmark scene recognition datasets Scene15, MIT67 and SUN397, respectively.

Methodology

The performance of Google Net is compared to Visual Geometry Group in image detection and recognition. The architecture of Google Net is also compared to that of Visual Geometry Group.

Google Net

Google Net is a deep convolutional neural network that uses inception network architecture which gave an impressive result in classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). Lin et al (2013). The architecture is enhanced by enlarging the depth and width and ensuring that the budget for computation remains the same. This was achieved by applying the Hebbian principle and the intuition of multi-scale processing in the decision making of the network. Szegedy et al (2015)

The most impressive aspect is that most of the enhancement in Google Net is not based on improvement in hardware, bigger datasets and larger models, but majorly on innovations due to new concepts, enhanced algorithmic programs and network design. Google Net in the ILSVRC of 2014 utilises 12 times lesser parameters compared to that of the AlexNet that won the ILSVRC competition in 2012 however, it produced a more impressive and precise result Krizhevsky et al (2012). The greatest profit on object detection did not come from an unenlightened and larger deep network, but the application collaboration of deep architectures and classic computer vision Girshick et al (2014).

Google Net uses the Inception architecture which estimate and deal with optimal local sparse architecture of a convolutional vision network using available dense components. The assumption made is that the network is built from convolutional building block, that is, from translation invariance. The optimal local transaction is then determined and replicated spatially. Arora et al (2014) proposed a layer-by-layer design in which the examined correlation statistics of the previous layer is clustered into groups of units of high correlation which connects and form the next layer. It is assumed that some regions of the input image match with individual units from previous layers, forming groups of filter banks. In the layers next to the input (lower layers) correlated units focuses on local regions. Hence, many clustered are generated in a single region which can be covered by a 1x1 convolutions layer in the proceeding layer Lin et al (2013). Minute number of more spatially dispersed clusters which can be dealt with using convolution over bigger patches which results in a reducing number of patches over greater and greater regions which can generate patch-alignment issues. Patch-alignment issues can be prevented by restricting current incarnations of the Inception design and construction to 5x5, 3x3 and 1x1 filter sizes. Szegedy et al (2015).

Visual Geometry Group

Visual Geometry Group focuses on increasing the depth of the convolutional neural network design by adding more convolutional layers while keeping other parameters constant. This is done in every layer by applying modest (3x3) filters which gave an impressive result in the ILSVRC. A constant size image of 224 x 224 RGB. The difference of the mean RGB value from the training set and the from each pixel goes through a pile of modest pile of convolutional neural network layers consisting of modest 3 x 3 filters. The convolutional stride it kept constant at one pixel maintaining the spatial resolution at 1 pixel for 3 x 3 convolutional layers. A 2 x 2-pixel window is utilized in the application of max-pooling based on 2 strides. Simonyan and Zisserman (2014).

A pile of convolutional layers next to three fully connected layers. Each of the initial two layers consists of 4096 channels and the third consists of 1000 channels utilized for 1000 categorizations. The last layer consists of a soft-max layer. Simonyan and Zisserman (2014).

All hidden layers consist of rectification (ReLU) that accounts for non-linearity is utilized in place of the Local Response Normalisation (LRN) (Krizhevsky et al., 2012)

Comparison of the architecture of Visual Geometry Group to Google Net

The Visual Geometry Group (VGG) architecture applied very little 3 x 3 and 1 x 1 filters with a single stride. It also applied a 2 x 2 size max pooling which also has a stride of 2 x 2 in dimension. Before the application of pooling layer to form a block, convolutional layers were first stacked together with spectacular replication of the convolutional pooling block design which led to the development of a very deep 16- and 19-layers models that achieved a top-5 error of 7.32% in the ILSVRC.

The architecture of the VGG is shown in Figure 5 the of the input is constant at 244 x 244. In the initial processing stage, the mean value of the RGB is removed from the individual pixels in the image. Then image then goes through a stack of convolutional layers in the input channel with very little 3 x 3 receptive field filters, which is then configured by linear transformation to set to a 1 x 1 filter, before the application of non-linearity.

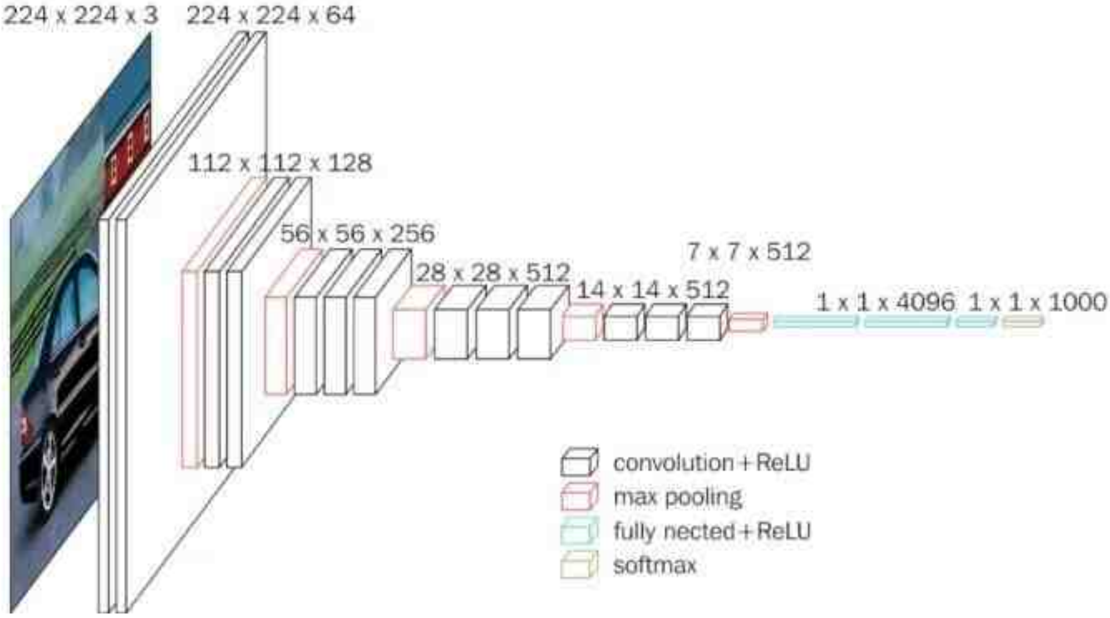

Figure 5 architecture of the VGG

The stride is constant at 1 and spatial pooling is done by 5 max-pooling layers. Max-pooling is carried out using a 2 x 2-pixel window, with a fixed stride size of 2. The first two fully-connected layers have 4096 channels each, the third contains 1000 channels for 1000 distinct classification of images, with a softmax final layer. ReLu activation function is applied to all the hidden layers

VGG network is categorized into five configurations as shown in the table 1 below. It can be observed from the table that the depth of the configuration increased from A to E by the addition of layers, while the configuration follows a universal pattern. The number of channels of convolutional layers also increased from 64 to 512 as we go from A to E. Image is trained in the VGG network by modifying the multinomial logistic regression objective for maximum efficiency by applying mini-batch slope descent with backpropagation. Simonyan K. and Zisserman A (2014).

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 1 VGG network categorized into five configurations

The generalized design of a Visual Geometry Group block is delineated as a single or more convolutional layers with same number of filters with filter dimension of 3 x 3 and a 1 x 1 stride. It also has the same padding which enables the output size to be like that of the input size for individual filters and the application of rectified linear activation function. Next to Visual Geometry Group (VGG) block function is implemented by Keras functional API by specifying a simple model that receives inputs of square colour images and combines one VGG block to the model with two convolutional layers of 64 filters each, as shown in listing 1.

```
# Implementation of a single block VGG

from keras.models import Model

from keras.layers import Input

from keras.layers import Conv2D

from keras.layers import MaxPooling2D

from keras.utils.vis_utils import plot_model
```

```
# function to generate a vgg block
def vgg_block(layer_in, n_filters, n_conv):
    # addition of convolutional layers
    for _ in range(n_conv):
        layer_in = Conv2D(n_filters, (3,3), padding='same', activation='relu')(layer_in)
        # addition of max pooling layer
        layer_in = MaxPooling2D((2,2), strides=(2,2))(layer_in)
        return layer_in


# specify model input
visible = Input(shape=(256, 256, 3))
# addition of vgg module
layer = vgg_block(visible, 64, 2)
# generate model
model = Model(inputs=visible, outputs=layer)
# summarization of model
model.summary()
# plot of the model architecture
plot_model(model, show_shapes=True, to_file='vgg_block.png')
```

Listing 1 A summarized model of a single VGG block

Listing 2 shows the summarised VGG model with the addition of one VGG block to two convolutional layer which has 64 filters each, next to a max pooling layer.

Figure 6 shows a plot of the VGG model architecture which also shows a single VGG block that is added to two convolutional layers consisting of 64 filters which is followed by a max pooling layer.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 256, 256, 3)       0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 256, 256, 64)      1792
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 256, 256, 64)      36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 128, 128, 64) 0
=================================================================
Total params: 38,720
Trainable params: 38,720
Non-trainable params: 0
_________________________________________________________________
```

Listing 2 Output of a summarized model of a single VGG block

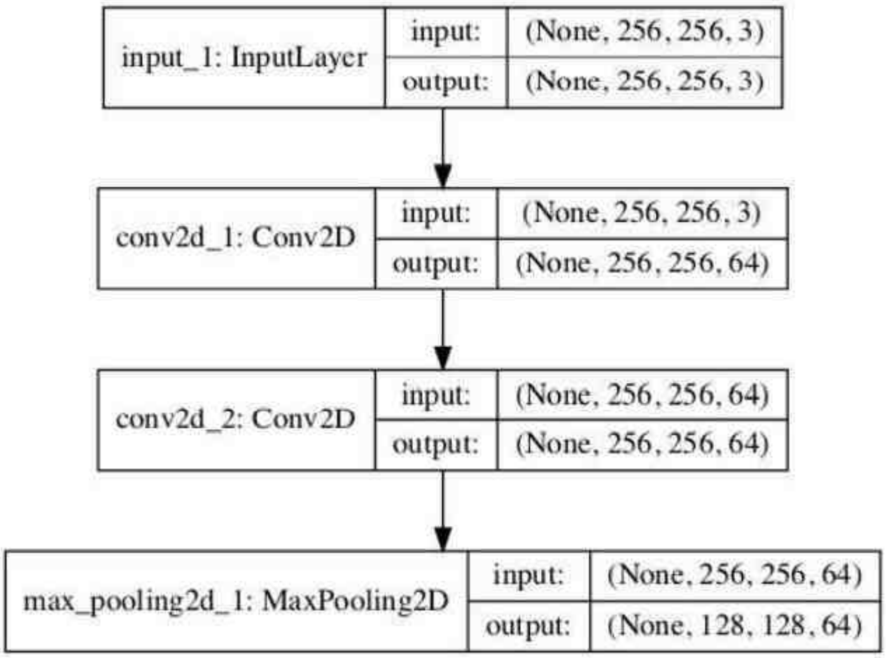

Figure 6 Plot of a single VGG Convolutional Network Block

The demerit of VGG is that it takes longer time to train when compared to Google Net and it has a very big weight based on its network architecture.

While, Google Net was developed from the modification of the inception module. The number of channels were brought down by the application of 1 x 1 convolution and error feedback were also applied in a lot points in the network. 22 layers deep convolutional neural network was applied to develop very deep model. The output of the model consists of global average pooling which generated 7 million parameters and was able to achieve 6.67% top-5 error in the ILSVRC, which is 0.65% better than the 7.32% achieved by VGG in the same competition. Szegedy et al (2015).

Google Net architecture is 22 layers deep and consists of 27 pooling layers shown in Figure 7. It also consists of 9 linearly stacked inception model with global average pooling layers attached at each end.

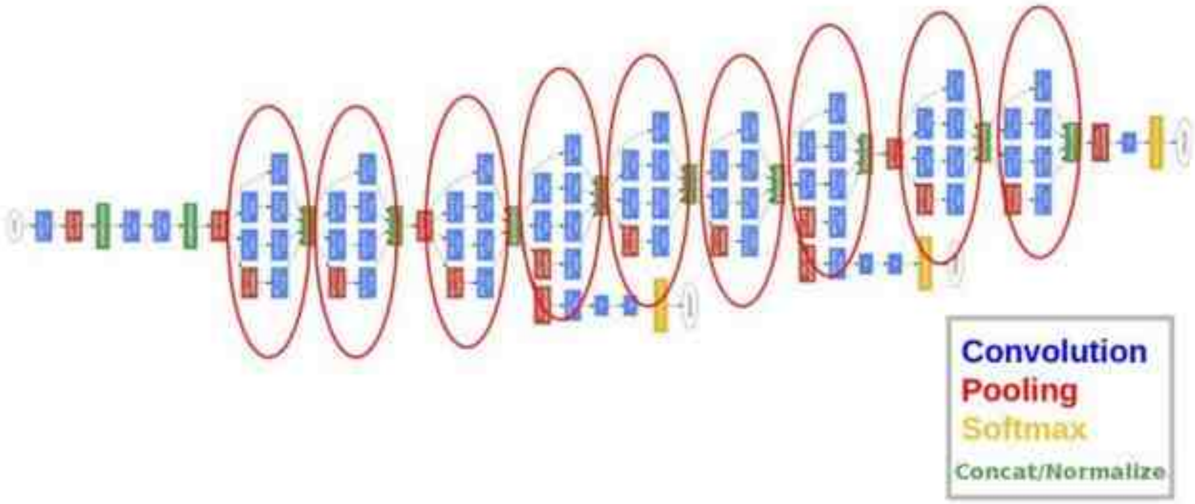

Figure 7 The architecture of Google Net

Table 2 depicts details of the architecture and the parameters involved for inception (3a) to inception (5b). The training of image in Google Net applies asynchronous stochastic slope descent with a constant learning rate schedule and a 0.9 momentum.

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 2 the architecture and the parameters involved for Google Net inception (3a) to inception (5b)

Figure 8 shows two classified distinct class of Google Net in the ILSVRC. Despite the similarities in the dogs, Google Net was able to classify them separately. This distinct class was achieved by the aid of domain knowledge.

(a) Siberian husky (b) Eskimo dog

Figure 8 Two distinct classes from the 1000 classes of the ILSVRC 2014 classification challenge

Google Net is an uncomplicated and robust architecture that enables the model to learn both parallel filters of the same size and different sizes, enabling learning at lots of different scales. Google Net model is designed by applying Keras functional API. Listing 3 function will generate an inception module with a fixed number of filters for each of the parallel convolutional layers.

The application of the function generates the reference to the previous layer as input, the filter number and gives back a reference to the chained filters layer which is eventually linked to more inception modules or submodules to make predictions. Listing 3 shows the implementation of an inception model of inception (3a). Szegedy et al (2015).

```
# Creation an inception module
from keras.models import Model
from keras.layers import Input
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers.merge import concatenate
from keras.utils.vis_utils import plot_model


# function for creation of naive inception block
def naive_inception_module(layer_in, f1, f2, f3):
```

```
    # 1x1 convolution
    conv1 = Conv2D(f1, (1,1), padding='same', activation='relu')(layer_in)
    # 3x3 convolution
    conv3 = Conv2D(f2, (3,3), padding='same', activation='relu')(layer_in)
    # 5x5 convolution
    conv5 = Conv2D(f3, (5,5), padding='same', activation='relu')(layer_in)
    # 3x3 max pooling
    pool = MaxPooling2D((3,3), strides=(1,1), padding='same')(layer_in)
    # concatenation of filters, with assumption of filters/channels last
    layer_out = concatenate([conv1, conv3, conv5, pool], axis=-1)
    return layer_out


# definition of the input to the model
visible = Input(shape=(256, 256, 3))
# addition of inception module
layer = naive_inception_module(visible, 64, 128, 32)
# creation of model
model = Model(inputs=visible, outputs=layer)
# summarization of model
model.summary()
# plot of model architecture
plot_model(model, show_shapes=True, to_file='naive_inception_module.png')
```

Listing 3 A summarized Google Net naïve inception module

Ouput of the summarised Google Net model of a naïve inception module is shown in Figure 9. and Listing 4 shows a plot of the model structure which depicts the parallel structure of the model and also the matching built of the individual element output of the module which gives room for direct concatenation in the third dimension of channels or filters.

```
____________________________________________________________________________________
Layer (type)                   Output Shape         Param #    Connected to
====================================================================================
input_1 (InputLayer)           (None, 256, 256, 3) 0
____________________________________________________________________________________
conv2d_1 (Conv2D)              (None, 256, 256, 64) 256        input_1[0][0]
____________________________________________________________________________________
conv2d_2 (Conv2D)              (None, 256, 256, 128 3584       input_1[0][0]
____________________________________________________________________________________
conv2d_3 (Conv2D)              (None, 256, 256, 32) 2432       input_1[0][0]
____________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D) (None, 256, 256, 3) 0           input_1[0][0]
____________________________________________________________________________________
concatenate_1 (Concatenate)    (None, 256, 256, 227 0          conv2d_1[0][0]
                                                               conv2d_2[0][0]
                                                               conv2d_3[0][0]
                                                               max_pooling2d_1[0][0]
====================================================================================
Total params: 6,272
Trainable params: 6,272
```

Listing 4 Output of a summarized Google Net naïve inception module

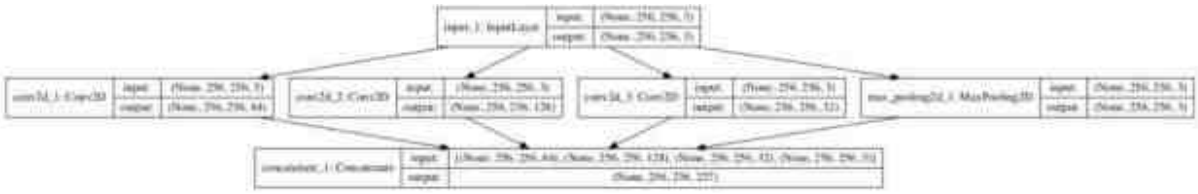

Figure 9 Plot of a summarized Google Net naïve inception module

Table 3 depicts further comparison of the VGG and the Google Net architecture. Xu et al (2018)

	VGG	**Google Net**
Input (RGB image)	224 x 224	224 x 224
Convolution (kernel size/stride)	3 x 3/1 3 x 3/1	7 x 7/2
Max pooling	2 x 2/2	3 x 3/2
Convolution (kernel size/stride)	3 x 3/1 3 x 3/1	3 x 3/1
Max pooling	2 x 2/2	3 x 3/2
Convolution (kernel size/stride)	3 x 3/1 3 x 3/1 1 x 1/1	Inception (3a) Inception (3b)
Max pooling	2 x 2/2	3 x 3/2
Convolution (kernel size/stride)	3 x 3/1 3 x 3/1 1 x 1/1	Inception (4a) Inception (4b) Inception (4c) Inception (4d) Inception (4e)
Max pooling	2 x 2/2	3 x 3/2
Convolution (kernel size/stride)	3 x 3/1 3 x 3/1 1 x 1/1	Inception (5a) Inception (5b)
Pooling	Max Pool 2 x 2/2	Average pool 7 x 7/1
Linear	FC-4096 FC-4096 FC-1000	FC-1000
Output	Softmax	Softmax

Table 3 Architectural comparison of VGG and Google Net

Table 4 below shows the results of the performance of Google Net and VGG in the ILSVRC 2014. It can be seen that Google Net came first with an error rate of 6.67%.

CNN	Year	Place	Error (top-5)
Google Net	2014	1st	6.67%
VGG	2014	2nd	7.32%

Table 4 1st and 2nd position of ILSVRC 2014

Conclusion

Visual Geometry Group established that very deep convolutional neural network architecture (LeCun et al., 1989; Krizhevsky et al., 2012) is greatly effective in the accurate categorization, and gave an impression result in the ImageNet challenge dataset. Significance of the depth of a convolutional neural network in visual representations which enabled it to achieve a top-5 error of 7.32% in the ILSVRC. VGG also takes longer time to train and has a very big weight based on its network architecture when compared to Google Net simple architecture and lesser training time. Google Net demonstrated that estimating the predicted optimal sparse structure by applying usable dense building blocks is an efficient method for enhancing neural networks to enable computer vision. The greatest merit of this inception architecture is the degree of efficiency achieved with small increase of computational needs in comparison to shallower and narrower architectures. Google Net was more efficient than the Visual Geometry Group in the ILSVRC and was able to achieve a top 5 error of 6.67%, which is 0.65% better than that of the Visual Geometry Group.

References

Blakemore C. and Cooper G.F. (1970). Development of brain depends on the visual environment. *Nature 228 (5270), 477 – 478*

Hubel D.H. and Wiesel T.N. (1959). Receptive fields of single neurones in cat's striate cortex. *J. Physiol. 148, 574 – 591*

Hubel D.H. and Wiesel T.N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology 160 (1), 106*

Fukushima K. (1975). Cognitron: A self-organizing multilayered neural network. *Biological cybernetics 20 (3), 121 – 136.*

Fukushima K., Hirota M., Terasaki P.I., Wakisaka A. and Togash H. (1984). Characteristics of Sialosylated Lewis^{x} as a New Tumor-associated Antigen. *Cancer Research 44 (11), 5279 – 5285.*

Fukushima K. (1986). A neural network model for selective attention in visual pattern recognition. *Biological Cybernetics 55 (1), 5 – 15.*

Fukushima K. (1987). Neural network model for selective attention in visual pattern recognition and associative recall. *Applied Optics 26 (23), 4985 – 4992.*

Hubel D.H. and Wiesel T.N. (1965). Receptive fields and functional architecture in two nonstriated visual areas (18 and 19) of the cat. *Journal of neurophysiology 28 (2), 229 – 289*

Hubel D.H. and Wiesel T.N. (1977). Ferrier lecture-Functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society of London. Series B. Biological Sciences 198 (1130), 1 – 59*

LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E. and Wayne (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation 1 (4), 541 – 551 networks*

LeCun Y. and Bengio Y. (1995). Convolutional networks for images, speech and time series. *The handbook of brain theory and neural networks 3361 (10), 1995*

LeCun Y., Bottou L., Bengio Y. and Haffner P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86 (11), 2278 – 2324*

Krizhevsky A., Sutskever I. and Hinton G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems 25*

Zeiler M. D. and Fergus R. (2014). Visualizing and understanding convolutional networks. *European conference on computer vision, 818 – 833*

Lin M., Chen Q., and Yan. S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*

Szegedy C., Liu W., Jia Y., Sermanet P. and Dragomir S. R. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition, 1 – 9*

Girshick R., Donahue J., Darrell T. and Malik J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition, 580 – 587*

Arora S., Bhaskara A., Ge R. and Ma T. (2014). Provable bounds for learning some deep representations. *International conference on machine learning, 584 – 592*

Simonyan K. and Zisserman A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv: 1409.1556*

Xu H., Han Z., Feng S., Zhou H. and Fang Y. (2018). Foreign object debris material recognition based on convolutional neural networks. *EURASIP journal on image and video processing 2018 (1), 1 – 10*

Jangra M., Dhull S. K. and Singh K. K. (2020). ECG arrhythmia classification using modified visual geometry group network (mVGGNet). *Journal of Intelligent & Fuzzy Systems 38 (3), 3151 – 3165*

Mei Y., Jin H., Yu B., Wu E. and Yang K. (2021). Visual geometry group-UNet: deep learning ultrasonic image reconstruction for curved parts. *The journal of the Acoustical Society of America 149 (5), 2997 – 3009*

Xu R., Liu X., Wan H., Pan X. and Li J. (2021). A Feature Extraction and Classification Method to Forecast the PM 2.5 Variation Trend Using Candlestick and Visual Geometry Group Model. *Atmosphere 12 (5), 270*

Raja J., Shanmugam P. and Pitchal R. (2021). An automatic early detection of glaucoma using support vector machine based visual geometry group 19 (vgg-19) convolutional neural network. *Wireless Personal Communications 118 (1), 523 – 534*

Al-Qizwini M., Barjastesh I., Al-Qassab H. and Radha H. (2017). Deep learning algorithm for autonomous driving using Google Net. *IEEE Intelligent Vehicles Symposium (IV), 89 – 96*

Khan R. U., Zhang X. and Kumar R. (2019). Analysis of ResNet and Google Net models for malware detection. *Journal of Computer Virology and Hacking Techniques 15 (1), 29 – 37*

Zhong Z., and Jin L. and Xie Z. (2015). High performance offline handwritten chinese character recognition using Google Net and directional feature maps. *13th International Conference on Document Analysis and Recognition (ICDAR), 846 – 850*

Tang P., Wang H. and Kwong S. (2017). G-MS2F: Google Net based multi-stage feature fusion of deep CNN for scene recognition. *Neurocomputing 225, 188 – 197*

Simonyan K. and Zisserman A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv: 1409. 1556*