Parallel Convolutional Neural Networks for Object Detection

Article *in* Journal of Advances in Information Technology · January 2021

DOI: 10.12720/jait.12.4.279-286

3 authors:

Adedeji Olugboja
York College, City University of New York
10 PUBLICATIONS **543** CITATIONS
SEE PROFILE

Zenghui Wang
University of South Africa
236 PUBLICATIONS **8,958** CITATIONS
SEE PROFILE

Yanxia Sun
University of Johannesburg
208 PUBLICATIONS **6,504** CITATIONS
SEE PROFILE

Parallel Convolutional Neural Networks for Object Detection

Adedeji Olugboja and Zenghui Wang
College of Science, Engineering and Technology, University of South Africa, Johannesburg, South Africa
Email: {djimayowa, wangzengh}@gmail.com

Yanxia Sun
Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg, South Africa
Email: ysun@uj.ac.za

***Abstract*—In recent years, Convolutional Neural Network (CNN) has been widely applied in speech/image/video recognition and classification. Although the results achieved are so impressive, CNN architecture is becoming more and more complex since CNN includes more layers to achieve better performance. In this paper, we developed a new CNN structure with several parallel CNNs and a Back-Propagation Neural Network (BPNN). The parallel CNNs can have the same or different numbers of layers. The outputs of the CNNs are the inputs of a fully connected BPNN. The structure of the proposed model can reduce the complexity of CNN by reducing the total number of CNN layers while the performance of feature extraction can be improved. The proposed model was validated based on CIFAR-10, CIFAR-100, and MNIST datasets and the achieved performance of the model is promising.**

***Index Terms*—convolution neural network, back-propagation neural network, feature extraction, visualization, object detection**

I. Introduction

Great accomplishment has been achieved by using Convolutional Neural Networks (CNN) in the areas of image and video recognition [1]-[4]. The success could not be accomplished without the improvement of the computing system Graphical Processing Unit (GPU) and the availability of image datasets [5], [6]. The GPU is becoming a popular tool used in the field of image processing, computer vision, and machine learning, to mention a few because of their rapid manipulation and altered memory, which accelerate the creation of images. GPU highly parallel structure makes them more efficient than the general-purpose Central Processing Unit (CPU).

From the advent of CNNs, a lot of research works have been made to the architecture and these works have improved the performance of CNNs. On studying the architecture, it was found that as the years went by, the architectures are becoming more and more complex and complicated, that is, from LeNet5 in 1998 which is five layers deep to Cuimage in 2016 which is the esembling of six models. Moreover, CNNs have pushed forward the research of deep learning, LeNet5 gave more insight to CNNs, which makes us know that the features of an image are situated across the whole image through convolution, and the learnable features can be extracted from multiple locations with fewer parameters. Most of the recent architectures are built on the inspiration of LeNet5.

Some years after the advent of LeNet5, neural networks lost out of favour, because it was slow to train and there was not much data available then. But after the computing power is becoming increasingly, faster and the data was becoming available everywhere, the neural network started seeing the light of the day. In 2010, Dan Claudiu and Jurgen Schmidhuber developed a 9-layer neural network, they used GPU for their image processing and the training time was reduced drastically than the normal CPU, and this was the first time that the Neural Network (NN) training was implemented using GPU. The network architecture was called DanCiresanNet.

Alex Krizhersky had the biggest breakthrough in using CNN in computer vision. He called his network architecture AlexNet. It was submitted for the imageNet ILSVRC challenge in 2012. AlexNet system was so amazing and it came out top having a top 5 error of 16% and the runner-up had 26% error. AlexNet architecture was similar to Yann LeCun architecture which was called LeNet. The difference was that AlexNet was bigger and deeper. Fig. 1 shows the LeNet and AlexNet architecture.

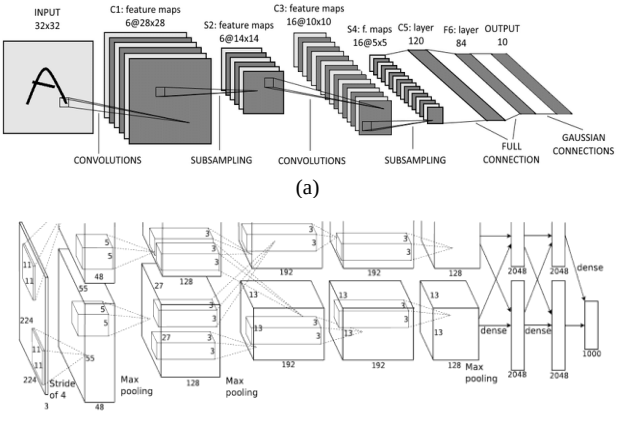


Figure 1. (a) LeNet 5 architecture [2] (b) AlexNet architecture [1].

Manuscript received October 26, 2020; revised June 22, 2021.

doi: 10.12720/jait.12.4.279-286

His main improvements to CNNs architecture are 1) he used the rectified linear units (RELU); 2) he used a method called dropout to carefully avoid some neurons when training, to prevent overfitting, 3) he used the max-pooling instead of the average pooling and 4) for the training time to be reduced, he used GPUs.

Matthew Zeiler and Rob Fergus improved AlexNet architecture by increasing the size of the convolutional layer in the middle, that is, the stride and filter used were made smaller for the first layer and some of the hyper-parameters were twisted and they called their network ZFNet which won the ILSVRC 2013 competition. Fig. 2 shows the ZFNet architecture network

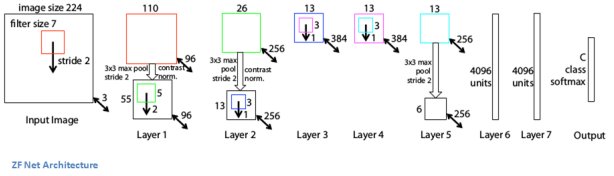


Figure 2. ZFNet architecture network [3].

Szegedy *et al*. from Google developed their architecture and called it GoogLeNet and their contribution was that they reduced the parameters of the network. AlexNet architecture has 60M parameters and in GoogLeNet the parameters were reduced to 4M by developing an inception module. Another development of their architecture was that they didn't use the fully connected layer on top of the convolutional network, instead, they used average pooling. Their architecture is shown in Fig. 3.

Figure 3. GoogLeNet architecture network [8].

In ILSVRC 2014 competition, Karen Simonyan and Andrew Zisserman networks called VGGNet got the second position after the GoogLeNet, and what makes their network different was that they showed that depth was an important element that makes the network do well. Their network has 16 convolutional and fully connected layers (16CONV/FC). Their VGGNet had a short-come, it is very hard to appraise and the memory used is much, that is, 140M parameters.

The winner of ILSVRC 2015 was ResNet which was developed by Kaiming He *et al*. In their network they used batch normalization massively, the fully connected layer was not included in their architecture as shown in Fig. 4.

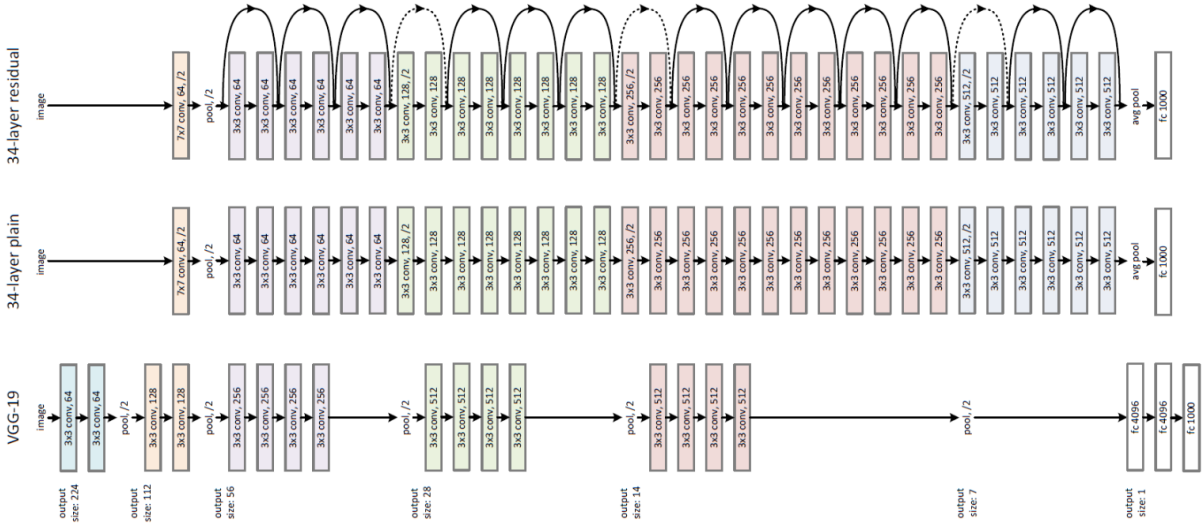


Figure 4. ResNet architecture network [4].

Over the years, there has been a great development in the architecture of convolutional neural networks [1], [7], [8]. This architecture has performed well and has reached the state-of-the-art in image recognition and classification. It has also been used successfully to identify faces, objects, traffic sign and in the medical field [9]-[11], but despite the success of CNNs, one will realise that the architecture of CNNs is becoming more and more complicated [12] proved in their paper titled "Do deep nets really need to be deep"? They concluded that shallow models can also learn, be accurate, and mimic deep mode in the classification task, and they experimented with the TIMIT and CIFAR-10 dataset.

To investigate and find a simple model without reducing the performance, this paper proposes a new method for image recognition and classification task. The rest of this paper is structured as follows. Section II and Section III gives the motivation behind our model and the architecture of our model, respectively. In Section IV the proposed model is validated based on some benchmark dataset. Section V visualizes the inner structure of our model. Conclusions will be given in the final section

II. Motivation

Human being perception is the result of the cooperation of many different brain areas. The outside world is viewed differently by different people, and it can also be said that the brain does not give the same interpretation for the same object, event, or an image. The brain alternates back and

forth to give meaning to things. This process is known as ambiguous or bi-stable stimuli. This knowledge has been in existence for over 150 years since the time of Hermann Von Helmholtz who is known as the prime father of psychology [13].

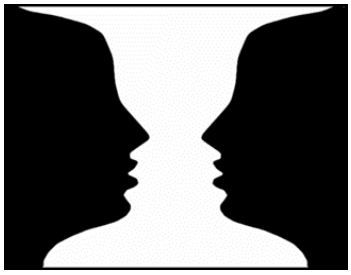


Figure 5. Rubins-vase.

People see things differently and make meanings of what they see, and this is the motivation that we emulated in proposing a new model. when the image in Fig. 5 is placed before people to give information about it, some said it is a flower vase, some said it is an ancient cup used by the kings and some others said it is two faces facing each other. One will have realised that different people looking at the same image gave different information about the image. More information can be deduced from an image if more people are allowed to view it. Based on this above analysis, our model is built to run this way. Several CNNs are working together to process the images and all the information gathered is integrated to have a good visualization and interpretation. We structured our model based on Yann LeCun architecture called LeNet5.

III. Proposed Architecture

Convolutional Neural Network (CNN) is a peculiar kind of neural networks, which uses three key concepts: local receptive fields shared weights and pooling. CNN makes full use of the local correlation by mandating a local connectivity structure between the neurons of adjacent layers. A definitely hidden layer *m*, which is connected to a local subsets units in the $(m-1)^{th}$ layer. The filter h_i is reproduced from one side to the other of the whole visual field. The reproduced unit has the same weight vector and has the same bias, and it is called the feature map layer. Convolving the input with a linear filter produces the feature map h^k, the bias, and a nonlinear function is added which is shown in the equation below:

$$h_{ij}^k = f((W^k * x)_{ij} + b_k) \quad (1)$$

W^k and b_k are weight and bias of the k^{th} feature map, $f(.)$ is the non-linearity [14]. In this study, we used the Rectified Linear Units (ReLU) for the non-linearity. The pooling layer was used and is non-linear downsampling. Introducing the pooling layer into the model reduces the computational complexity and it also produces a translation invariance.

The CNN last layer is a logistic layer, the output follows the class membership probability.

$$P(Y = ix, W, b) = soft\max(W_x + b) = \frac{e^{W_{ix+b_i}}}{\sum_j e^{W_{jx+b_j}}} \quad (2)$$

The network parameters are trained using the backpropagation method [15]. We combined CNN_1 CNN_2 …. CNN_n to abstract more features, and then merged them into fully connected Back-Propagation Neural Networks with three layers including one output layer. The proposed structures are shown in Fig. 6 and Fig. 7. Fig. 6 gives the structure with the same number of layers of CNNs, which look like different persons with similar recognition abilities or views. Fig. 7 gives the structure with a different number of layers of CNNs, which looks like different persons with different recognition abilities or views. In Fig. 6 and Fig. 7, the blocks 'Con', 'ReL', and 'Pooli' are referring to Convolution, Rectified Linear Unit, and Max Pooling, respectively. It should be noted that one convolutional layer of CNN includes three blocks: 'Con', 'ReL' and 'Pooli' in this study and it maybe include different block structures such as 'Con', 'ReL', 'Con', 'ReL' and 'Pooli'.

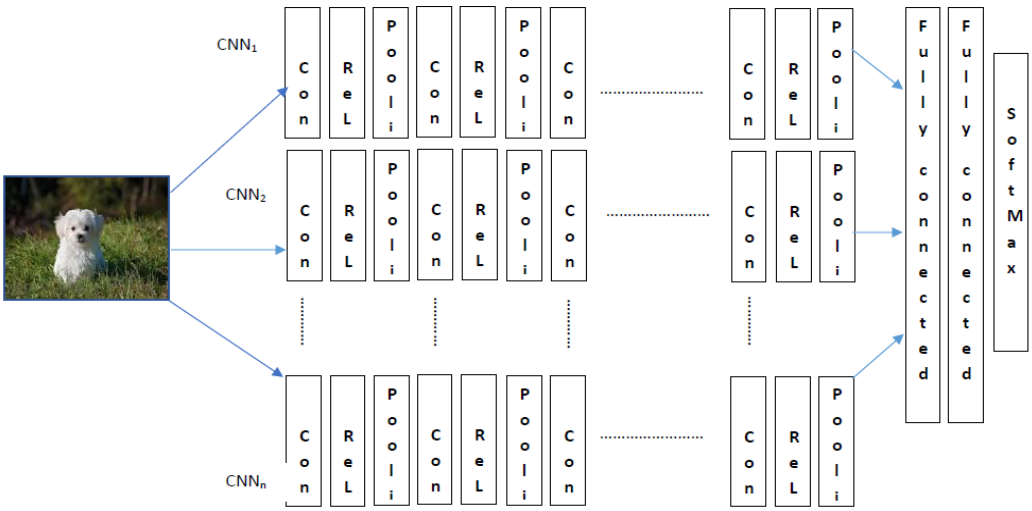


Figure 6. Diagrammatic representation of our model with the same number of layers of CNNs.

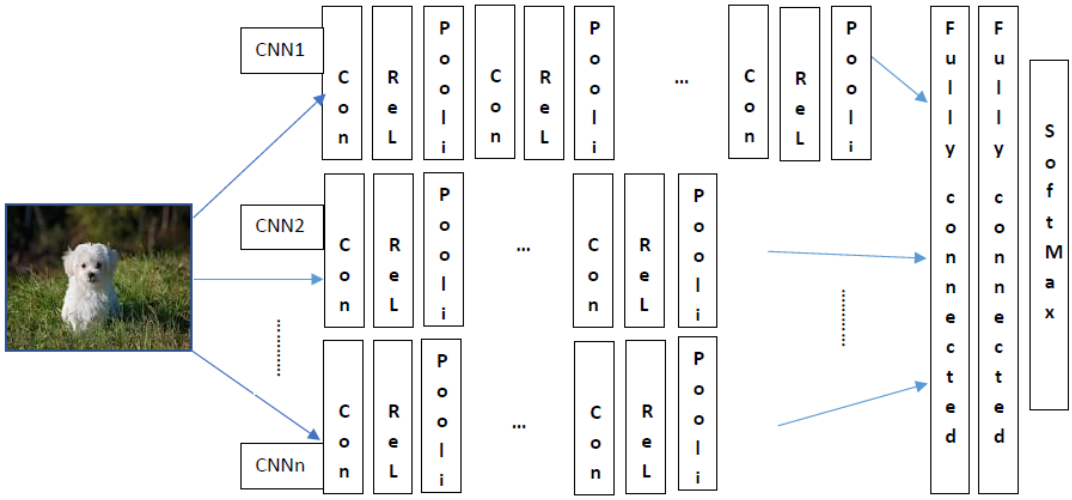


Figure 7. Diagrammatic representation of our model with different number of layers of CNNs.

IV. Experiments

To show the efficiency of the proposed model, it is necessary to do some experiments. We evaluated with different CNN architectures based on 3 benchmark datasets: MNIST, CIFAR-10, and CIFAR-100. The different CNN architectures and cases are:

1) One 5-convolutional-layer CNN, which is the normal CNN;
2) One 4-convolutional-layer CNN, which is the normal CNN;
3) One 3-convolutional-layer CNN, which is the normal CNN;
4) Two parallel 5-convolutional-layer CNNs, which is the proposed model with the same number of CNN layers;
5) Two parallel 4-convolutional-layer CNNs, which is the proposed model with the same number of CNN layers;
6) Two parallel 3-convolutional-layer CNNs, which is the proposed model with the same number of CNN layers;
7) Three parallel 5-convolutional-layer CNNs, which is the proposed model with the same number of CNN layers;
8) Three parallel 4-convolutional-layer CNNs, which is the proposed model with the same number of CNN layers;
9) Three parallel 3-convolutional-layer CNNs, which is the proposed model with the same number of CNN layers;
10) Three-CNN architecture also merged, with 5 convolutional layers, 4 convolutional layers, and 3 convolutional layers, which is the proposed model with different numbers of CNN layers;

It should be noted that the first three architectures are the existing CNN architecture called normal CNN, the architectures 4)-6 are the proposed structure and the parallel CNNs are having the same number of layers, and the architecture 7) is the proposed structure but the number of layers of the different parallel CNNs is different. The CNNs have the same structure although the numbers of convolutional layers are different. For example, in case 4) each of the CNN has five (5) convolutional layers, these CNNs are merged by concatenating their output into a 2 fully-connected Neural network layer and 1 output layer (softmax).

The numbers of filters used in the 5-convolutional-layer CNNs are 32, 32, 64, 64 and 64, respectively; and the numbers of the filters of the 4-convolutional layers 32, 32, 64 and 64, respectively; and the numbers of the filters of the 3-convolutional layers 32, 32 and 64, respectively. The size of weights used in the convolutional layer is 3×3. Each of the convolutional layers is followed by one Max-pooling layer. We also used a Rectified Linear Unit (ReLU) which is an activation function and comes after all the layers and this makes sure that the feature map is positive and all negative are set to zero. The two fully connected layers' dimensions are 516 and 128, respectively. The number of outputs of the soft-max layer is equal to the class of the labels of the task. We also incorporated the dropout technology, which was introduced by [16], [17]. We used it after the third (3) layers and after the second fully connected layer with a probability of 0.25 and 0.5, respectively. The dropout technique limits the complex interaction of the neurons.

We used a mini-batch of 128 sizes, the training process was started using a random initial weight and a learning rate of 0.001. The CNN is refined in a supervised fashion with the stochastic gradient descent until convergence.

In training our model, the label training set:

$$s = \left\{ \left(x_i, y_i \right) \right\}_{i=1}^{m} \tag{3}$$

In the algorithm, we calculated the weights of the convolution and affine layers. The loss function was minimized by choosing the right weight. It needs to determine the weights for:

$$\min \sum_{i=1}^{m} l\left(f\left(W; x_i \right), y_i \right) \tag{4}$$

here, w is the network weights, $f\left(W; x_i\right)$ this is the weight the network predicted for the input x_i and l which is the loss function. We used the gradient-based method to

solve the optimization problem. In this method, an iteration that uses the first-order approximation was used for the minimized function. For every step, the weight was updated in the direction of the loss function descent. Weights at the time $t - W^{(t)}$ at the next time step the weights are:

$$W^{(t+1)} = W^{(t)} - \mu_t \Delta_w \left(\sum_{i=1}^{m} l(f(W^{(t)} x_i), y_i \right) \quad (5)$$

here, μ_t is the learning rate, and a positive scalar Δ_w is a gradient concerning *W*. We used a fixed size mini-batch of 128. Nesterov's momentum is used with Stochastic Gradient Descent (SGD) [17]. Our algorithm was implemented in the Keras python library with TensorFlow backend and tested on window 10 with Intel dual-core i5 CPU.

A. CIFAR-10 Based Experiments

The dataset CIFAR-10 consists of 10 classes of images, which are natural images. The training images are 50000 and the testing images are 10000. The images in **CIFAR**-10 are RGB images with size 32x32. The different CNN architectures are subjected to the same conditions and hyper-parameters; the experimental results are shown in Table I. As can be seen from Table I, the following results can be obtained:

1) Comparing Cases 1, 2 & 3, it can be found that the recognition accuracy can be increased with the increment of the number of CNN convolutional layers.
2) Comparing Cases 1, 4 & 7, and Cases 2, 5 & 8, and Cases 3, 6 & 9, it can be found that the recognition accuracy can be increased with the increment of the number of CNNs.
3) Comparing Case 10 with Cases 1-9, it can be found that using different CNNs in parallel structure can improve the recognition accuracy a lot, which is also shown in Fig. 8. It should be noted that the total number of convolutional layers of Case 10 is equal to or less than Cases 7&8.

TABLE I. TEST ACCURACY RATE FOR CIFAR-10 OF DIFFERENT ARCHITECTURE

Case no.	Total Numbers of convolutional layers	CNNs	Test Accuracy (%)
1	5	One 5-convolutional-layer CNN	75.98
4	10	Two parallel 5-convolutional-layer CNNs	77.94
7	15	Three parallel 5-convolutional-layer CNNs	89.07
2	4	One 4-convolutional-layer CNN	73.58
5	8	Two parallel 4-convolutional-layer CNNs	75.94
8	12	Three parallel 4-convolutional-layer CNNs	84.67
3	3	One 3-convolutional-layer CNN	73.04
6	6	Two parallel 3-convolutional-layer CNNs	75.34
9	9	Three parallel 3-convolutional-layer CNNs	80.07
10	12	**Proposed Model with 1st CNN=5 layers, 2nd CNN= 4 layers, and 3rd CNN=3 layers**	**90.24**

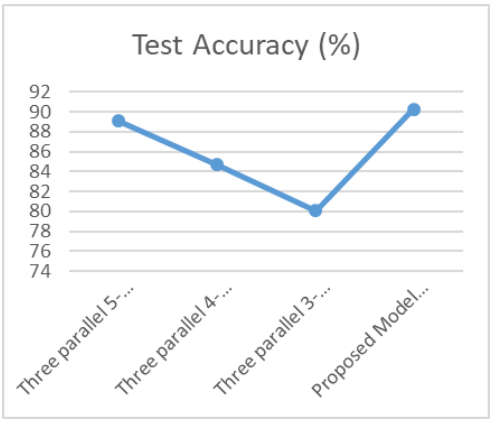


Figure 8. The test accuracy for CIFAR-10.

B. CIFAR-100 Based Experiments

This dataset has the same pattern and size as CIFAR-10 except that it has 100 class of images and has 600 images per class. CIFAR-100 has 500 images for training and 100 images for testing per class. The conditions and hyper-parameters, which are used for CIFAR-10, are used for this dataset. The experimental results are shown in Table II. As can be seen from Table II, the following results can be obtained

1) Comparing Cases 1, 2 & 3, it can be found that the recognition accuracy can be increased with the increment of the number of CNN convolutional layers.
2) Comparing Cases 1, 4 & 7, and Cases 2, 5 & 8, and Cases 3, 6 & 9, it can be found that the recognition accuracy can be increased with the increment of the number of CNNs.
3) Comparing Case 10 with Cases 1-9, it can be found that using different CNNs in parallel structure can improve the recognition accuracy a lot, which is also shown in Fig. 9.

TABLE II. TEST ACCURACY RATE FOR CIFAR-100 OF DIFFERENT ARCHITECTURE (WITH 5, 4, 3 CONVOLUTIONAL LAYERS RESPECTIVELY)

Case no.	Total Numbers of Convolutional layers	CNNs	Test Accuracy (%)
1	5	One 5-convolutional-layer CNN	62.78
4	10	Two parallel 5-convolutional-layer CNNs	65.10
7	15	Three parallel 5-convolutional-layer CNNs	69.98
2	4	One 4-convolutional-layer CNN	61.03

5	8	Two parallel 4-convolutional-layer CNNs	63.58
8	12	Three parallel 4-convolutional-layer CNNs	65.13
3	3	One 3-convolutional-layer CNN	60.58
6	6	Two parallel 3-convolutional-layer CNNs	63.10
9	9	Three parallel 3-convolutional-layer CNNs	62.41
10	12	**Proposed Model with 1st CNN=5 layers, 2nd CNN= 4 layers, and 3rd CNN=3 layers**	**70.54**

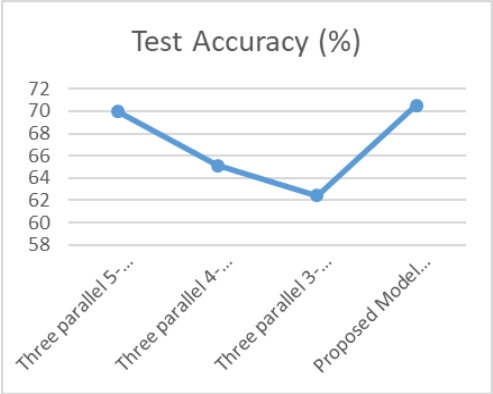


Figure 9. Test accuracy for CIFAR-100.

C. MNIST Based Experiments

This dataset consists of a handwritten digit which ranges from 0 - 9 and has a size of 28 × 28. 60000 images were used to train the model and 10000 images were used to test the model. The experiments were also subjected to the same conditions and hyper-parameters which are used in Sub-sections IV.A and IV.B. The experimental results are shown in Table III. As can be seen from Table III, the following results can be obtained

1) Comparing Cases 1, 2 & 3, it can be found that the recognition accuracy can be increased with the increment of the number of CNN convolutional layers.
2) Comparing Cases 1, 4 & 7, and Cases 2, 5 & 8, and Cases 3, 6 & 9, it can be found that the recognition accuracy can be increased with the increment of the number of CNNs.
3) Comparing Case 10 with Cases 1-9, it can be found that using different CNNs in parallel structure can improve the recognition accuracy a lot, which is also shown in Fig. 10.

TABLE III. TEST ACCURACY RATE FOR MNIST OF DIFFERENT ARCHITECTURE (WITH 5, 4, 3 CONVOLUTIONAL LAYERS RESPECTIVELY)

Case no.	Total Numbers of Convolutional layers	CNNs	Test Accuracy (%)
1	5	One 5-convolutional-layer CNN	98.32
4	10	Two parallel 5-convolutional-layer CNNs	99.10
7	15	Three parallel 5-convolutional-layer CNNs	99.20
2	4	One 4-convolutional-layer CNN	98.22
5	8	Two parallel 4-convolutional-layer CNNs	98.45
8	12	Three parallel 4-convolutional-layer CNNs	99.13
3	3	One 3-convolutional-layer CNN	97.11
6	6	Two parallel 3-convolutional-layer CNNs	98.22
9	9	Three parallel 3-convolutional-layer CNNs	99.02
10	12	**Proposed Model with 1st CNN=5 layers, 2nd CNN= 4 layers, and 3rd CNN=3 layers**	**99.31**

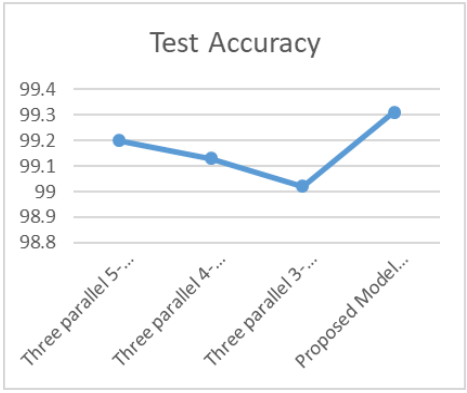


Figure 10. Test accuracy for MNIST.

V. VISUALIZING THE INNER STRUCTURE

To investigate the proposed model, we try to visualize the internal structure of the proposed method. The filters were visualized by their highest activated patches, which show the dimension of each of the representation space. The digit 4 of the MNIST dataset is used in this investigation and is shown in Fig. 11. The internal representation of the first and second layer is shown in Fig. 12 and Fig. 13, it can be seen that the three CNNs observed the digit 4 in different ways, when the features are merged and sent to the fully connected layer for interpretation, and the proposed model can capture more features which means the proposed method can easily recognize the image as 4 in this experiment.

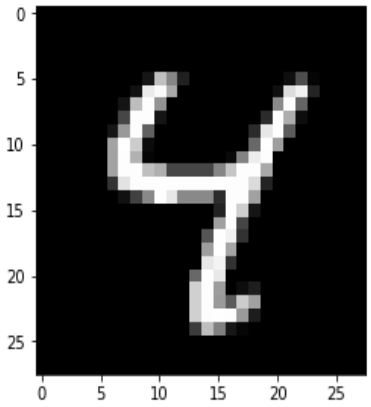
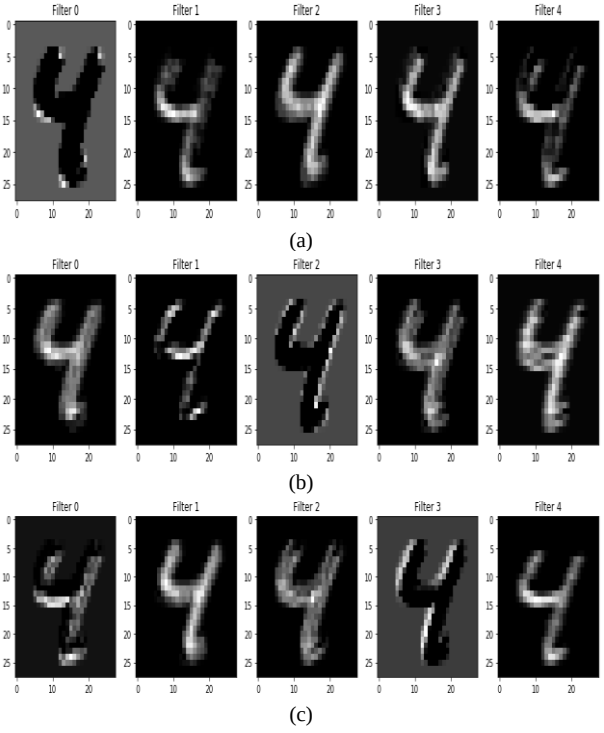
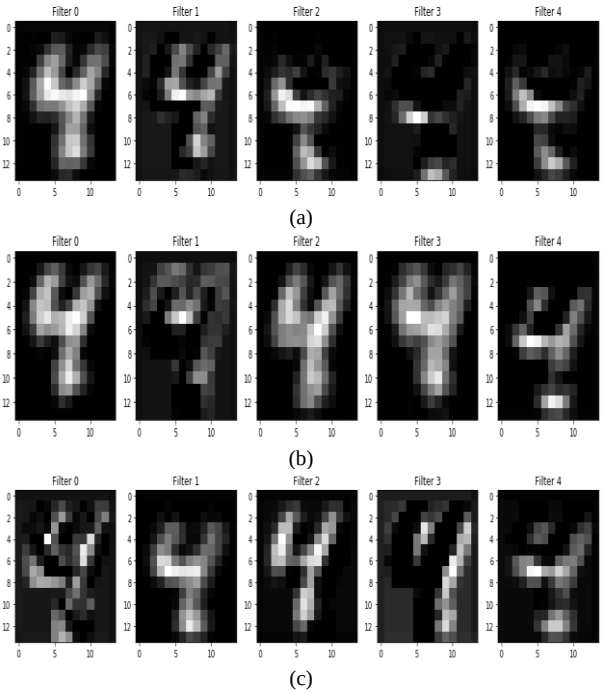


Figure 11. Digit 4 of the original image.



(a) (b) (c)

Figure 12. Visualization of visual information processing through the first layers. (a) CNN 1, (b) CNN 2, (c) CNN 3.



(a) (b) (c)

Figure 13. Visualization of visual information processing through the second layers. (a) CNN 1, (b) CNN 2, (c) CNN 3.

This experiment also explained why the parallel CNNs with different or same convolutional layers can achieve better recognition performance comparing with the normal CNN since different CNNs can be looked as different persons and different features can be obtained by different persons and then the different features are combined and used based on a fully connected BPNN to recognize images.

The performance of Case 10 is better than other cases since the model with a different number of layers of parallel CNNs can obtain more features, which looks like the persons with different recognition abilities or views can get more features from the given images.

VI. Conclusion

A new CNN model was proposed for classification tasks, and this model includes several parallel CNNs with the same or different convolutional layers and a fully connected Back-Propagation Neural Network. The model can fully abstract useful information by combining several CNNs as several people can work together. The model with different architectures was tested on MNIST, CIFAR-10, and CIFAR-100 datasets and the result obtained was impressive especially the model of a different number of layers of parallel CNNs. The reasons for the good performance of the proposed model were also investigated. In the future, we are going to improving the performance of our model by fine turning some of the parameter use such as using Batch normalization, data augmentation, the weight regularization to mention a few.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

A. Olugboja carried out the research under the supervision of Z. Wang and Y. Sun; and all authors approved the final version.

Acknowledgment

This research is supported partially by South African National Research Foundation Grants (No. 112108 and 112142), and South African National Research Foundation Incentive Grant (No. 95687 and 114911), Eskom Tertiary Education Support Programme Grants, Research grant from URC of University of Johannesburg.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1106-1114, 2012.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2323, 1998.

[3] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," in *Proc. the International Conference on Learning Representation*, 2013, vol. 33, pp. 1-9.

[4] M. Lin, Q. Chen, and S. Yan, "Network in network," arXiv Prepr. arXiv:1312.4400, vol. 17, pp. 24-32, 2014.

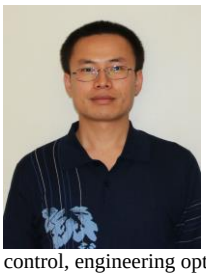
[5] J. Gu, *et al.*, "Recent advances in convolutional neural networks," arXiv:1512.07108, vol. 5. pp. 1-14, 2015.

[6] C. Li, Y. Yang, M. Feng, C. Srimat, and H. Zhou, "Optimizing memory efficiency for deep convolutional neural networks on GPUs," in *Proc. International Conference on High-Performance Computing, Networking, Storage, and Analysis*, Salt Lake City, UT, USA, Nov. 2016, pp. 633-644.

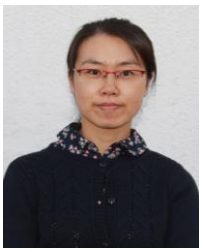
[7] W. Yu, K. Yang, Y. Bai, T. Xiao, H. Yao, and Y. Rui, "Visualizing and comparing AlexNet and VGG using deconvolutional layers," *Int. Conf. Mach. Learn.*, vol. 48, pp. 2-12, 2016.

[8] C. Szegedy, W. Liu, Y. Jia, and P. Sermanet, "Going deeper with convolutions," arXiv Prepr. arXiv 1409.4842, vol. 23, pp. 144-1149 2014.
[9] J. Torresen, J. W. Bakke, and L. Sekanina, "Efficient recognition of speed limit signs," in *Proc. the 7th Int. IEEE Conf. Intell. Transp. Syst.*, Washington, USA, Oct. 2004, vol. 45, pp. 2-6.
[10] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. the IEEE International Conference on Computer Vision*, 2009, pp. 2146-2153.
[11] N. Tajbakhsh, *et al.*, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Trans. on Med. Imaging*, vol. 35, no. 5, pp. 1299-1312, 2016.
[12] L. Ba and R. Caurana, "Do deep nets really need to be deep ?" arXiv Prepr. arXiv1312.6184, vol. 2014, pp. 1-6, 2013.
[13] D. Cahan, *Hermann Von Helmholtz and the Foundations of Nineteenth-Century Science*, University of California Press, 1993.
[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv:1207.0580, vol. 34, pp. 45-50, 2012.
[15] A. Karpathy. (July 2016). CS231n convolutional neural networks for visual recognition, Stanford CS231n. [Online]. Available: http://cs231n.github.io/convolutional-networks/
[16] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. the 30th International Conference on International Conference on Machine Learning*, Atlanta, GA, USA, June 16-21, 2013, pp. 1139-1147.
[17] Y. LeCun, L. Bottou, G. B. Orr, and K. Muller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*, Springer-Verlag Berlin Heidelberg, 1998, vol. 54, pp. 9-50.

Adedeji M. Olugboja is a Ph.D. student at the University of South Africa (UNISA). He received his BSc. and MSc. in Information Technology from the National Open University of Nigeria. Presently, he works in the Department of Computer Science at Eswatini Medical Christian University, Swaziland. His main research areas are machine learning, deep learning, and image processing.



Zenghui Wang received the B.Eng. degree in automation from NavalAviation Engineering Academy, China, in 2002, and the Ph.D. degree in control theory and control engineering from Nankai University, China, in 2007. Currently, he is a Professor with the Department of Electrical and Minning Engineering, University of South Africa, Florida, South Africa. His research interests include model predictive control, nonlinear control, engineering optimization, image/video, and so on.



Yanxiz Sun received her joint qualification: DTech in Electrical Engineering, Tshwane University of Technology, South Africa, and Ph.D. in Computer Science, University Paris-EST, France in 2012. She is an associate professor at University of Johannesburg. Her research interests include Engineering Optimization, Artificial Intelligence, and Control Systems and so on.