

The Feature Parallelism Model of Visual Recognition

Marwa Yousif Hassan¹, Abdi O. Shuriye¹, Aisha-Hassan Abdallah¹, Momoh J. E. Salam² and Othman O. Khalifa¹

¹*Department of Electrical and Computer Engineering*

²*Department of Mechatronic Engineering*

Kulliyyah of Engineering

International Islamic University Malaysia

marwayuosif@gmail.com, aisha@iium.edu.my

Abstract

In this work, the Feature Parallelism Model of visual recognition, which addresses the parallel nature of the human brain compared to the hierarchal (serial) brain model, was studied. First, its accuracy rate and training time were compared to those of DeepFace, a leading industry algorithm for face recognition. Both models were trained using ImageNet object recognition dataset. Accuracy rates were almost the same, around 57% top-1 error rate and 33% top-5 error rate. Training time for feature parallelism model has dropped to 21% less than that of Deep Face. Second, we have investigated feature parallelism model under depth, i.e., when adding more layers along the horizontal axis. We have tested the model with 5, 6, 7, and 8 layers respectively; we found that the best results both in terms of accuracy rates and training time were obtained with the six-layered model. Although the training time enhancement was only a few milliseconds when going from 5 to 6 layers, it has worsened significantly when going from 6 to 7 layers. In fact the training time has tripled, i.e., training time of the 7- layers model is three times of that of the 6- layers model. It continues to worsen by a fewer rate with the 8- layers model. Similarly, accuracy rate was better with the 6- layers model by about 1% of that of the 5- layers model; however, it has worsened by more than 5% whenever we add more layers above six. We consider those results are biologically plausible, as they conform to the biological fact that the cerebral cortex is organized in 6- layers. We've concluded that the organization of parallel processing units into 6- layers, either in our brains or in artificial vision systems, may enhance both processing time and accuracy rates.

Keywords: *Feature Parallelism Model, Deep learning, Neuroscience, Computer vision*

1. Introduction

Nature has continuously inspired inventions throughout the history of mankind. Deep learning is one field that has been inspired by neuroscience and the brain. Deep learning systems have led to great performances of artificial vision [1-2]. However, the well-known neuroscience theories have not yet been utilized by artificial vision, not to mention the undiscovered ones. Current deep learning systems are based on the hierarchical (Serial) model of the brain [3]. That is, multiple layers stacked one after another. To learn a task, information is processed in one layer before is handled to subsequent layers in the hierarchy.

Feature Parallelism Model [4] has addressed the parallel nature of the brain. It conceptualizes underutilized facts about the human visual system; namely, the Feature Integration Theory of visual attention "FIT" [5-8]. In the model, visual features such as shape, color and motion are handled in parallel. The model has been implemented along the shape feature, in which sub- features such as edges, texture and parts are processed in parallel, instead of hierarchical as was the case in previous deep learning models.

In Section 2, we show the theory related to Feature Parallelism Model; Sections 3 shows implementations and results and in Section 4 we discuss those results. Finally, we conclude our paper with a summary in Section 5.

2. The Theory

Since the 80s of the last century, the Feature Integration theory (FIT), which states that different visual features such as shape, color and motion are processed in parallel, has been developed. However, to the best of our knowledge, that principle is yet to be implemented in modern computer vision systems, until recently in Feature Parallelism Model. Here, we review the principle of parallel processing along with the related cortical column concept and the laminar organization of the cortex. Also we show how these concepts are utilized in the design of Feature Parallelism model.

2.1. The Principle of Visual Processing: Parallel versus Serial (Hierarchical)

It has been shown by the Feature Integration Theory (FIT) of visual attention that the processing of each feature dimension, like color, shape, size, motion or orientation is conducted in parallel; while the integration and conjunction of those features together into a whole scene, a face for example, is conducted serially [5-8].

From anatomy we know that many evidences support the FIT theory in our brains. In the retina, there are not less than 17 ganglion cell types; each one conveys different visual information to the upper parts of the visual pathway in parallel. They follow “Retinal Tiling” parallel strategy; that is each ganglion cell type is thought to tile the retina, extract specific kind of visual information, the kind of information it is characterized by, from the entire visual field; process and convey it to the brain independently of the other cell types. Subset of a type does not cover the entire visual field, and different types do not show spatial relationship [9], and [10].

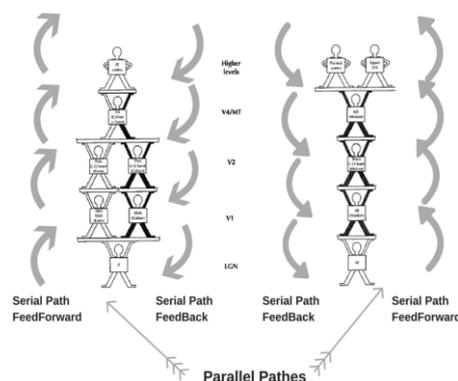


Figure 1. Hybrid Parallel/ Serial Architecture of the Visual System Adapted from [9]

The visual pathway at its upper part is believed to split into two major parallel pathways (Figure 1), one path is to process and recognize objects, this is the “What” pathway; the other path is to process motion, that is the “Where” pathway. The whole scene is comprised of the outputs of both paths integrated together at the uppermost part of the visual hierarchy. Neuroscientists used to believe that within each of these major parallel paths the visual processing is hierarchical and serial (Figure 2). However, this belief has been challenged by more recent neuroscience studies. In spite of that, hierarchical/serial processing within major pathways is widely adopted by modern computer vision algorithms [11, 1-2]. Nowadays, scientists believe that our brains are massively parallel

biological devices, even within the previously thought to be as highly serial areas (Figure 3) [9], [10] and [12].

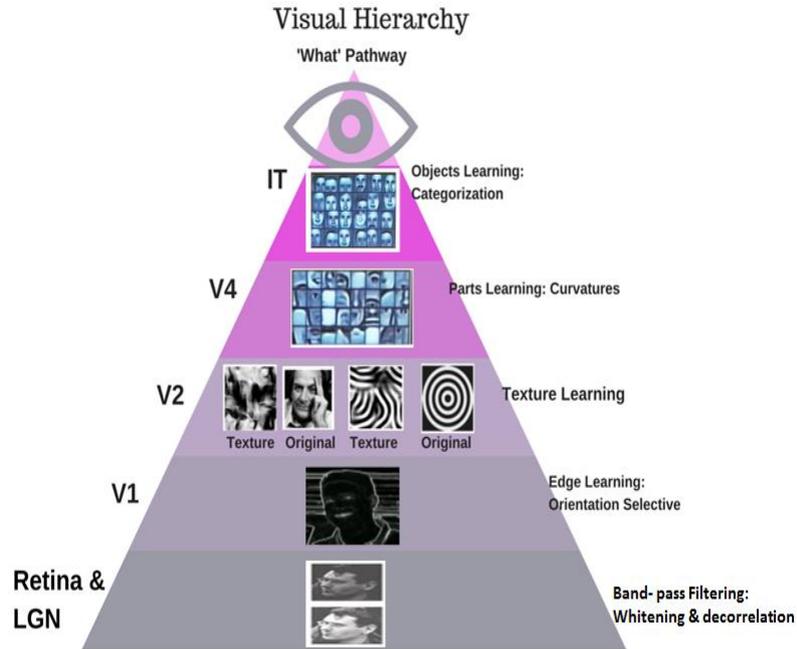


Figure 2. The Hierarchical Model of the Brain

Functions along the (What: Object Recognition) pathway.

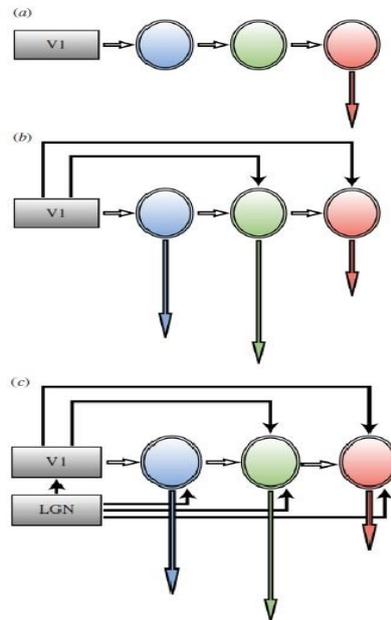


Figure 3. Parallel Processing in the Visual Cortex Adapted from [12]

(a) The classical picture of a single hierarchical chain with two (or more) nodes and a terminal node (in pink) at which activity becomes perceptually explicit (needs no further processing).

(b) The parallel outputs from V1 to two (or more) areas of the upper parts of the visual pathway and the serial, hierarchical, connections between them. Unlike in (a), activity becomes perceptually explicit at each node of the chain.

(c) The more elaborate picture of parallel outputs from V1 and directly from the LGN to three or more areas of the upper parts of the visual pathway, as well as the serial connections between the visual areas.

2.2. Functional Architecture of the Cerebral Cortex

2.2.1. Laminar Organization

Differences in thickness, sizes and the shapes of neurons of the layers of the cerebral cortex have led researchers more than a century ago to identify about fifty distinct areas of the cortex. This classification was the basis to link different functions to these areas. However, the design principle that underlies the six-layered structure of the cerebral cortex remains a mystery. This six-layered structure is uniform across different species and different cortical areas [13-14].

Number of layers in previous deep learning models is considered a hyper-parameter [15-16], it's chosen either experimentally or arbitrary. A hyper-parameter is a variable that need to be adjusted and chosen prior to training. It is defined as a variable to be set prior to the actual application of the learning algorithm to the data, a one that is not directly selected by the algorithm itself [17].

2.2.2. Cortical Columns

A cortical column is the basic information processing unit of the cortex [18]. It refers to cells in any vertical cluster that share the same tuning for a given receptive field's attribute [19]. It is the narrow chain of neurons extending vertically across the six cellular layers, perpendicular to the surface of the cortex [18]. To explain more, consider the orientation selectivity of V1. One cortical column is responsible for the processing of only one orientation (say, the vertical orientation) and not responsible for another (*e.g.* the horizontal or the 45° orientation) [18-20]. While Information inside a column is processed serially across its six layers, we can consider columns as the elementary parallel processing units of the cortex. Some have argued that a cortical column is a structure without a function [19], Here we can argue that dividing the cortex into small parallel units of similar tuning attributes, *i.e.* 'columns', may significantly reduce processing time and make learning faster.

2.3. The Feature Parallelism Model [4]

In Figure 4 below, the general "feature parallelism" model for object recognition is shown. The illustration shows that features such as color, shape and motion, are processed independently in parallel. Within each feature dimension there are parallel paths for sub-dimensions of that feature. For face recognition, the feature "Shape" is subdivided into 3 parallel sub-features, which are, texture, parts, and edges (Figure 5).

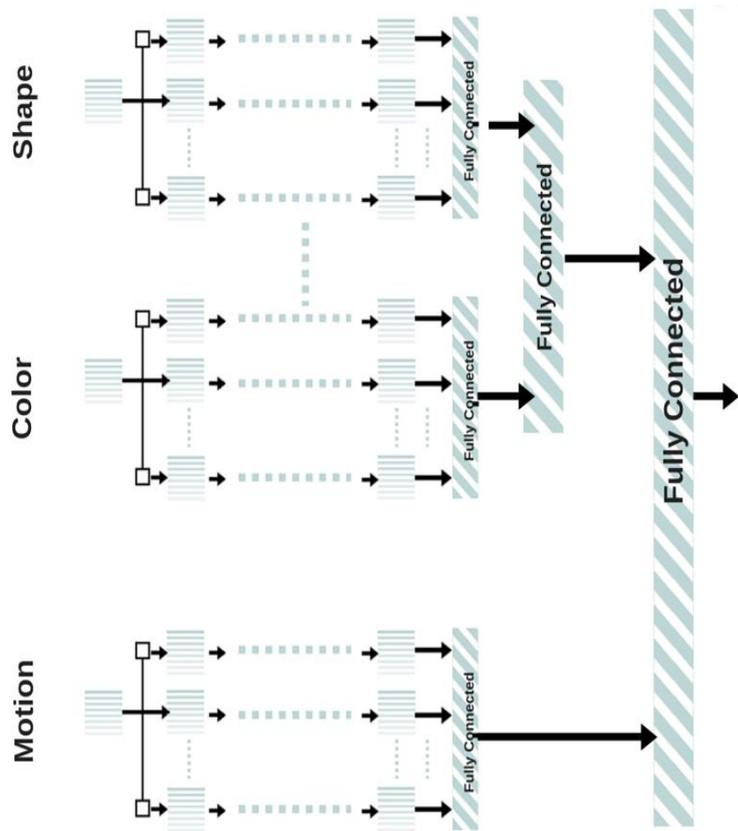


Figure 4. "General Feature Parallelism" Model for Object Recognition

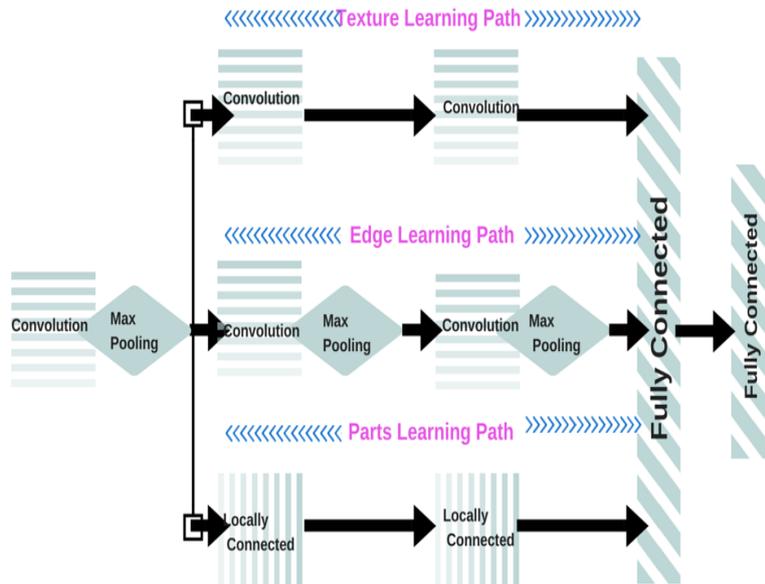


Figure 5. "Feature Parallelism" Model for Face Recognition along the "Shape" Feature Dimension

3. Implementation and Results

3.1. DeepFace versus Feature Parallelism Model

We have implemented the model in Figure 6 using Cuda- convnet2 deep learning framework [27], ImageNet ILSVRC2012 object recognition dataset, and a one Nvidia Tesla K20 GPU card.

In Figure 6 below, 3 parallel paths, the middle consists of 4 convolution layers each followed by a max pooling layer to denote the edge learning path. The upper one consists of 3 convolution layers to denote the texture learning path. The bottom consists of 3 locally connected layers to denote the part learning path. All paths are integrated together at F7, a fully connected layer that generates 4096x3 outputs followed by another fully connected layer, F1000 that generates the 1000 outputs (classes) of ImageNet object recognition dataset.

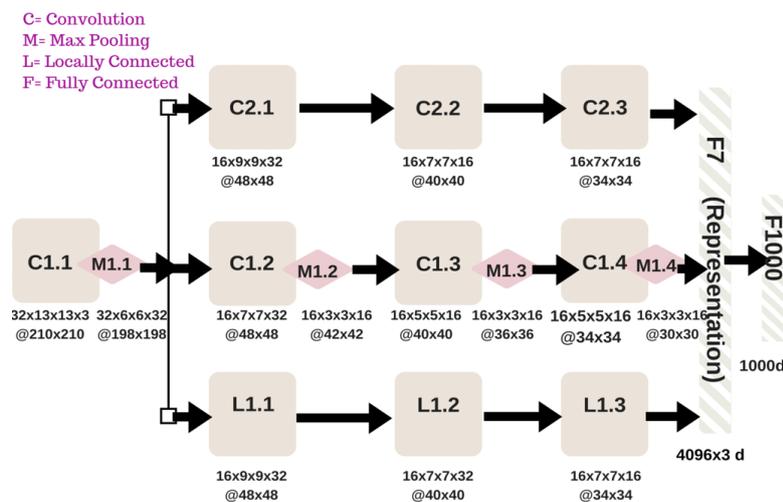


Figure 6. An Implementation of Feature Parallelism Model along the “Shape” Feature

We have tested the network implemented in Figure 6 above and examined error and time performances. The network test data has saturated around 57% for DeepFace architecture top-1 error rate (Figure 7), and around 33% top- 5 error rate (Figure 9). For Feature Parallelism architecture, test data has saturated around 57% top-1 error rate (Figure 8), and around 33% top-5 error rate (Figure 10), showing almost the same accuracy rates for both models

The screen shot on Figure 11 shows that training patches require for DeepFace architecture around 31 sec. and 200 ml sec. The feature parallelism architecture requires around 24.630 sec. per a training patch (Figure 12), a round 21% drop in training time, a significant improvement of time performance.

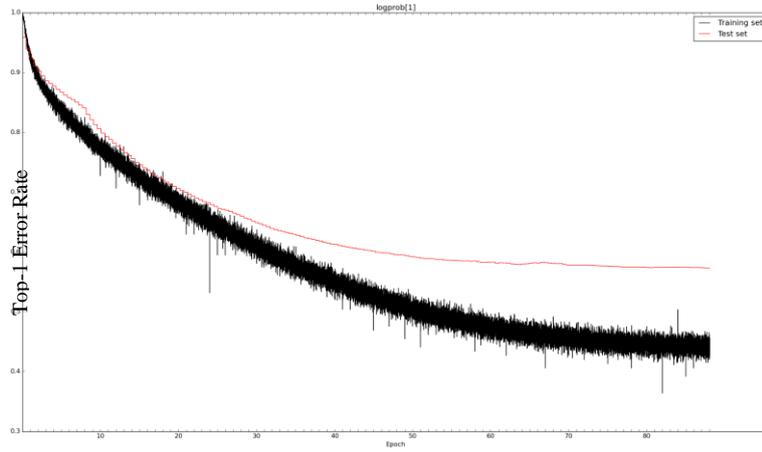


Figure 7. Top-1 Error Rate-DeepFace Architecture

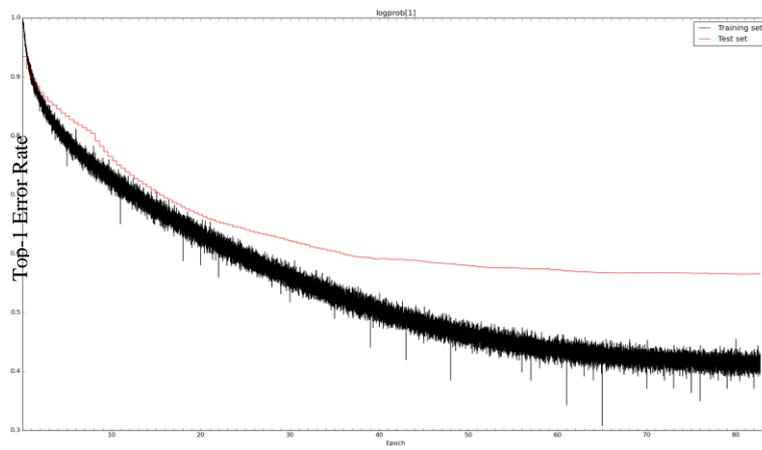


Figure 8. Top-1 Error Rate-Feature Parallelism Architecture

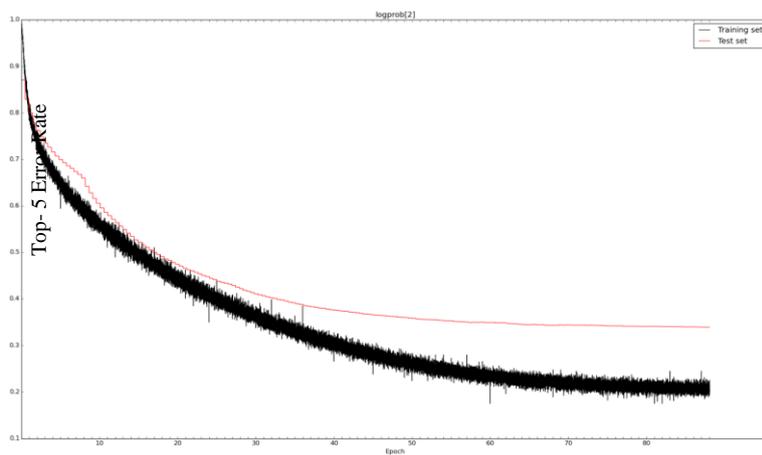


Figure 9. Top-5 Error Rate-DeepFace Architecture

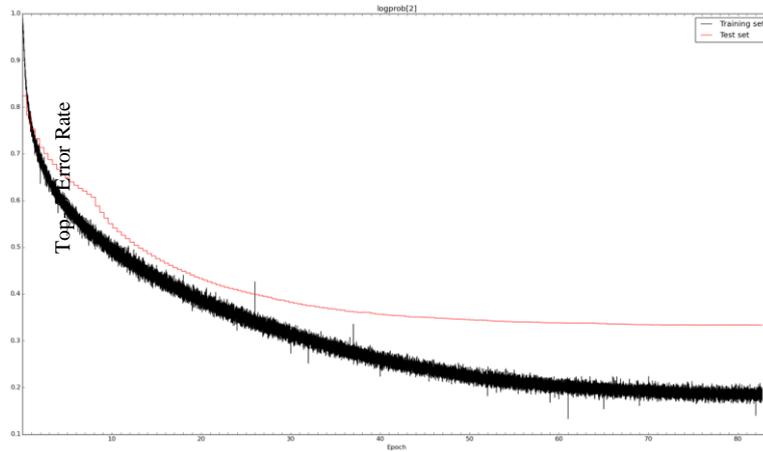


Figure 10. Top-5 Error Rate-Feature Parallelism Architecture

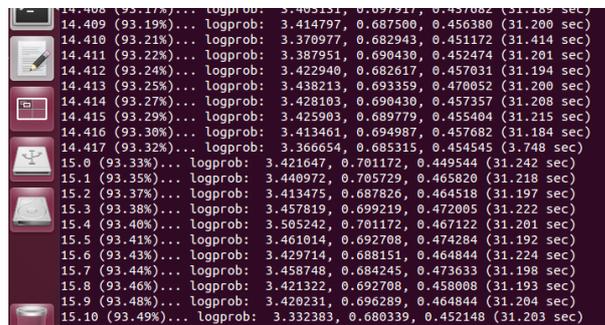


Figure 11. A Screen Shot of DeepFace Architecture while in Training; Notice the Time per a Training Patch is Around 31.200 Sec.

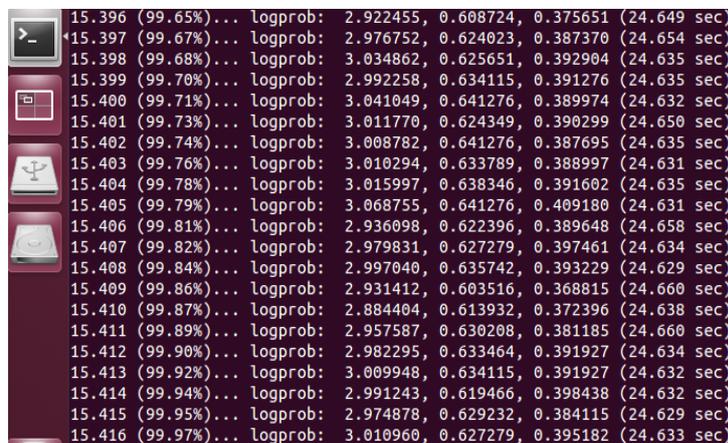


Figure 12. A Screen Shot of a One Layer Deeper “Feature Parallelism” Architecture while in Training; Notice the Time per a Training Patch is Around 24.630 Sec.

3.2. Investigating the Effect of Depth on Feature Parallelism Model

We have used smaller image size for this particular experiment; 136x136 images instead of 210x210 images used for the previous experiment. The same dataset, ImageNet ILSVRC object recognition dataset and the same GPU device, Nvidia K20 GPU card and

Cuda-convnets deep learning framework were used. Our reference model is the 6- layers model.

We have trained all models for 45 epochs. We found that the optimum number of layers is six; with 68% top-1 error rate (Figure 14) and 45% top-5 error rate (Figure 15). Training time for the six- layered model was around 8.080 seconds (Figure 16). Training time for the 5- layered model is about 8.450 seconds (Figure 19), with 69% top-1 error rate (Figure 17) and 46% top-5 error rate (Figure 18). Results for the 7- layers model were 73% top-1 error rate (Figure 20), 52% top-5 error rate (Figure 21), and training time of about 24.060 seconds (Figure 22). The model with 8 layers has results as follows: 78% top-1 error rate (Figure 23), 58% top-5 error rate (Figure 24), and around 27.350 seconds training time (Figure 25).

6- Layers End- to- End (Reference Model)

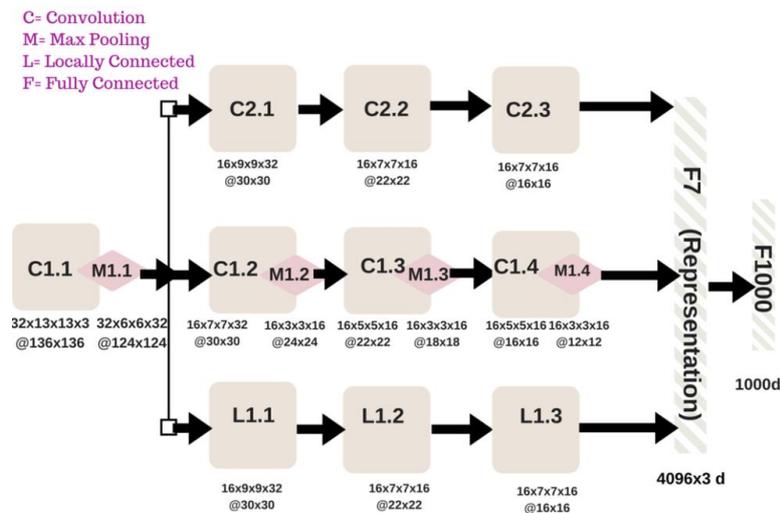


Figure 13. Feature Parallelism Model-6 Layers

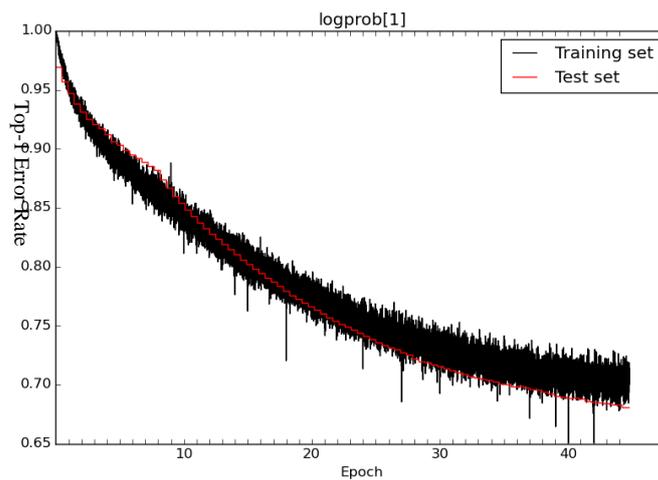


Figure 14. Top-1 Error Rate: 6- Layer Around 68%

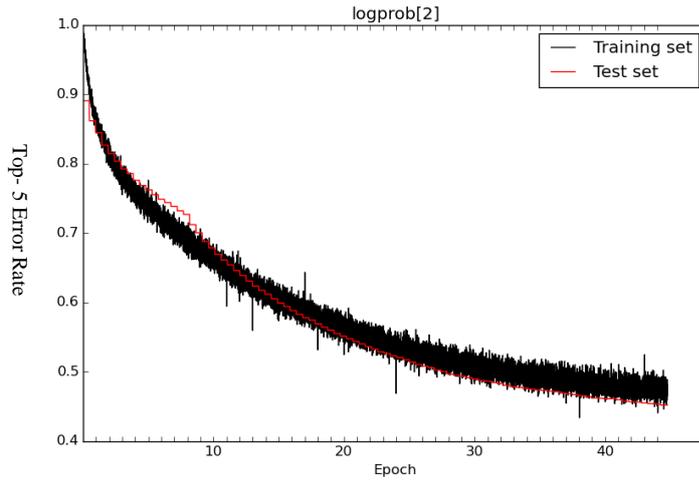


Figure 15. Top-5 Error Rate: 6-Layers Around 45%

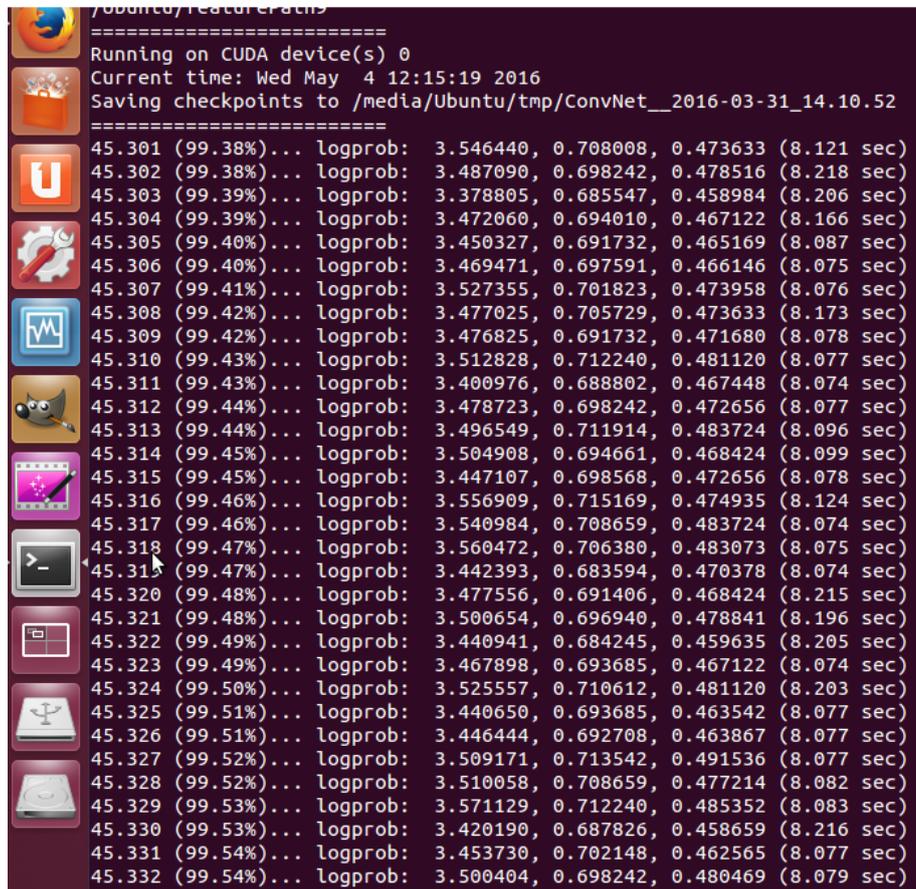


Figure 16. 6- Layers: Training Time Around 8.080 Seconds

5- Layers End- to- End

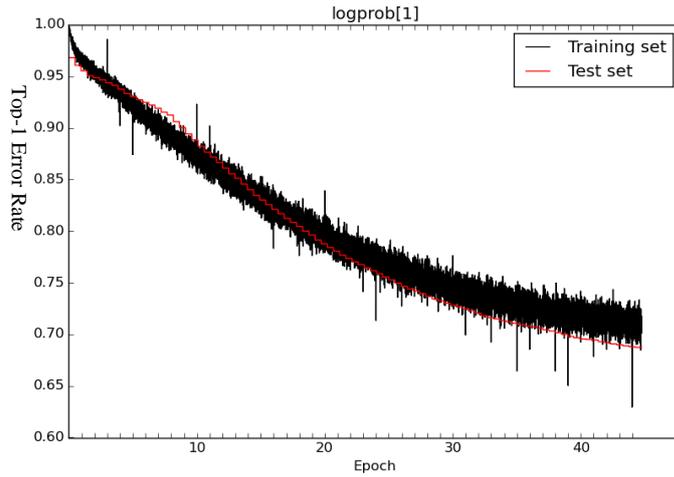


Figure 17. 5- Layers: Top-1 Error Rate Around 69%, 1% more Error Rate

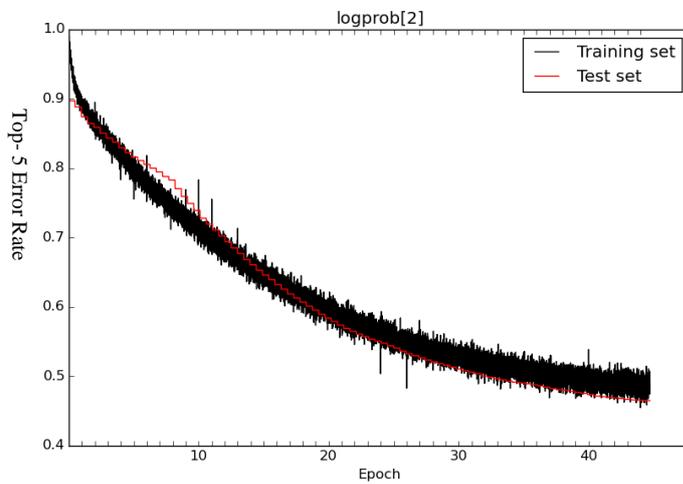


Figure 18. 5- layers: Top- 5 Error Rate Around 46%, 1% more Error Rate

```

/Ubuntu/featurePath11
=====
Running on CUDA device(s) 0
Current time: Wed May 4 11:48:17 2016
Saving checkpoints to /media/Ubuntu/tmp/ConvNet_2016-04-06_16.39.35
=====
45.301 (99.38%)... logprob: 3.591503, 0.723633, 0.487630 (8.477 sec)
45.302 (99.38%)... logprob: 3.562866, 0.708333, 0.483073 (8.436 sec)
45.303 (99.39%)... logprob: 3.535786, 0.706055, 0.479818 (8.440 sec)
45.304 (99.39%)... logprob: 3.556321, 0.701823, 0.476888 (8.438 sec)
45.305 (99.40%)... logprob: 3.552375, 0.708659, 0.485352 (8.444 sec)
45.306 (99.40%)... logprob: 3.591288, 0.715169, 0.490234 (8.437 sec)
45.307 (99.41%)... logprob: 3.565993, 0.716797, 0.490885 (8.553 sec)
45.308 (99.42%)... logprob: 3.557942, 0.709835, 0.485026 (8.436 sec)
45.309 (99.42%)... logprob: 3.589031, 0.722805, 0.497396 (8.523 sec)
45.310 (99.43%)... logprob: 3.588841, 0.714844, 0.496745 (8.438 sec)
45.311 (99.43%)... logprob: 3.482977, 0.700521, 0.472005 (8.454 sec)
45.312 (99.44%)... logprob: 3.582669, 0.710286, 0.491211 (8.438 sec)
45.313 (99.44%)... logprob: 3.550574, 0.718099, 0.479818 (8.445 sec)
45.314 (99.45%)... logprob: 3.545908, 0.711263, 0.485677 (8.442 sec)
45.315 (99.45%)... logprob: 3.592960, 0.708659, 0.499674 (8.473 sec)
45.316 (99.46%)... logprob: 3.576854, 0.715820, 0.489258 (8.457 sec)
45.317 (99.46%)... logprob: 3.654563, 0.715169, 0.499349 (8.435 sec)
45.318 (99.47%)... logprob: 3.585565, 0.708659, 0.488932 (8.449 sec)
45.319 (99.47%)... logprob: 3.530350, 0.713216, 0.477539 (8.459 sec)
45.320 (99.48%)... logprob: 3.600506, 0.713216, 0.493815 (8.446 sec)
45.321 (99.48%)... logprob: 3.569272, 0.711263, 0.493164 (8.475 sec)
45.322 (99.49%)... logprob: 3.491048, 0.695312, 0.476888 (8.539 sec)
45.323 (99.49%)... logprob: 3.517747, 0.701823, 0.471354 (8.440 sec)
45.324 (99.50%)... logprob: 3.601497, 0.717122, 0.495443 (8.433 sec)
45.325 (99.51%)... logprob: 3.546704, 0.715495, 0.490234 (8.441 sec)
    
```

Figure 19. 5- Layers: Training Time Around 8.450 Seconds, a few Milliseconds more Training Time

7- Layers End- to- End

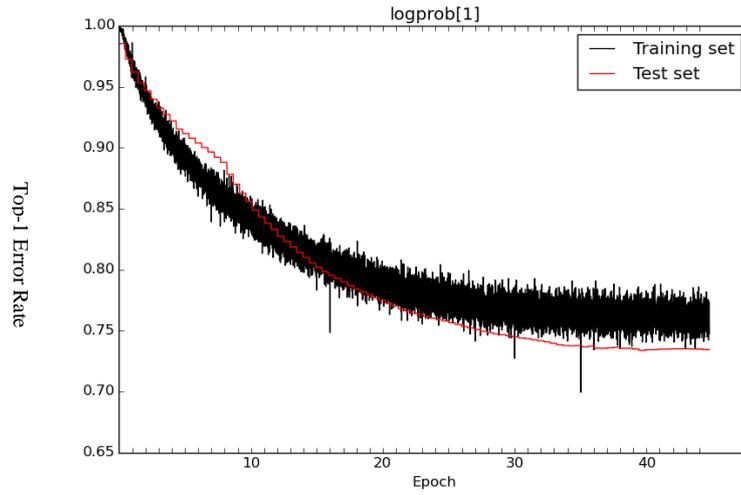


Figure 20. 7-Layers: Top-1 Error Rate Around 73%, 5% more Error Rate

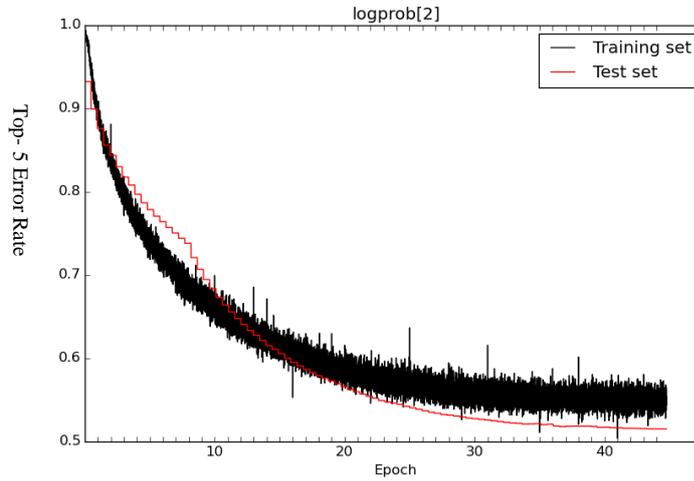


Figure 21. 7-Layers: Top-5 Error Rate Around 52%, 7% more Error Rate

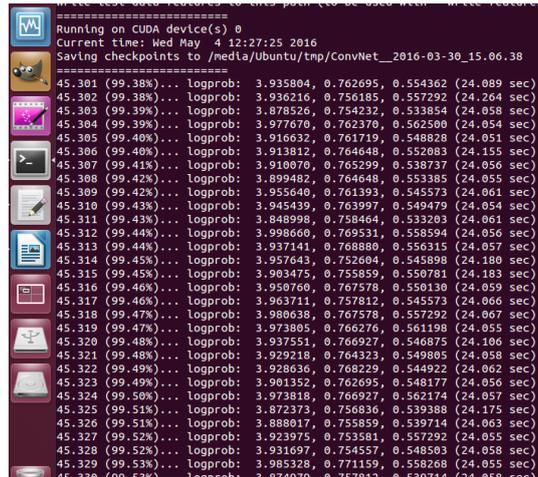


Figure 22. 7-Layers: Training Time Around 24.060 Seconds, almost 3 Times the Training Time of the 6-Layered Model

8- Layers End- to- End

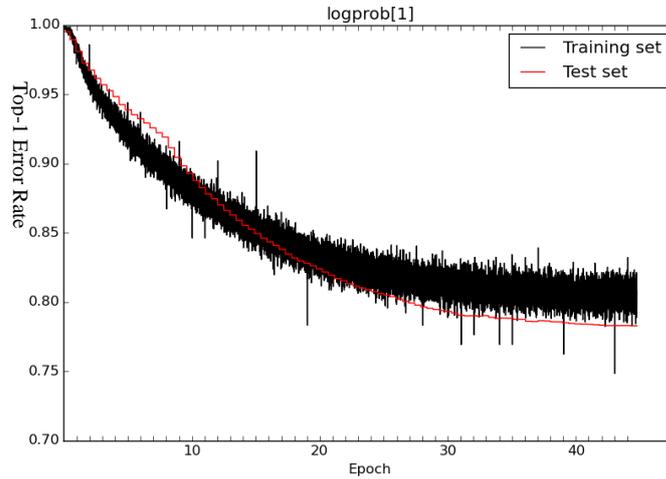


Figure 23. 8-Layers: Top-1 Error Rate Around 78%, 10% more Error Rate Compared to 6-Layered Model

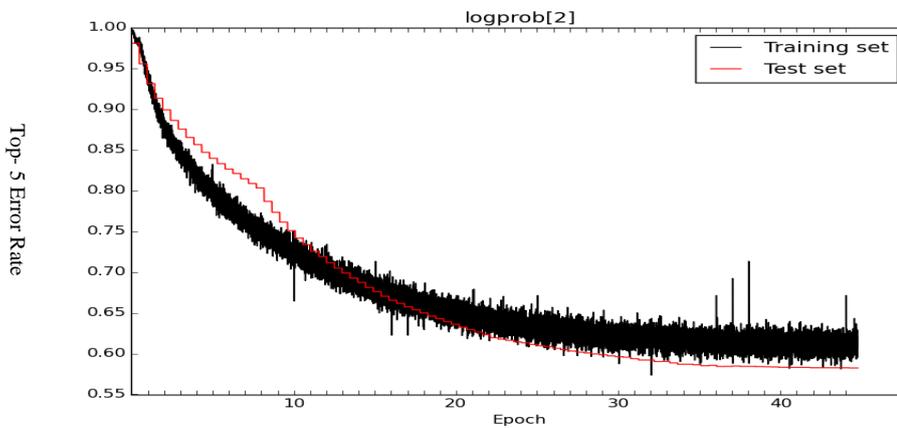


Figure 24. 8-Layers: Top-5 Error Rate Around 58%, 13% more Error Rate, Compared to 6-Layered Model

```
Write test data features to this path (to be used with --write-features
=====
Running on CUDA device(s) 0
Current time: Wed May 4 12:43:49 2016
Saving checkpoints to /media/Ubuntu/tmp/ConvNet_2016-03-28_14.46.46
=====
45.301 (99.38%)... logprob: 4.245807, 0.814453, 0.600911 (28.122 sec)
45.302 (99.38%)... logprob: 4.220267, 0.807292, 0.609375 (27.314 sec)
45.303 (99.39%)... logprob: 4.196749, 0.799805, 0.602539 (27.314 sec)
45.304 (99.39%)... logprob: 4.324251, 0.819661, 0.618164 (27.391 sec)
45.305 (99.40%)... logprob: 4.252449, 0.810221, 0.613281 (27.439 sec)
45.306 (99.40%)... logprob: 4.255644, 0.802083, 0.597005 (27.370 sec)
45.307 (99.41%)... logprob: 4.242230, 0.804362, 0.608724 (27.315 sec)
45.308 (99.42%)... logprob: 4.244366, 0.798828, 0.604818 (27.359 sec)
45.309 (99.42%)... logprob: 4.302440, 0.805013, 0.624674 (27.348 sec)
45.310 (99.43%)... logprob: 4.279879, 0.819010, 0.612956 (27.454 sec)
45.311 (99.43%)... logprob: 4.206632, 0.794596, 0.596680 (27.313 sec)
45.312 (99.44%)... logprob: 4.340428, 0.809245, 0.627930 (27.314 sec)
45.313 (99.44%)... logprob: 4.237346, 0.805664, 0.611328 (27.451 sec)
45.314 (99.45%)... logprob: 4.255441, 0.792318, 0.603190 (27.312 sec)
45.315 (99.45%)... logprob: 4.268523, 0.803385, 0.620443 (27.316 sec)
45.316 (99.46%)... logprob: 4.303213, 0.824544, 0.628255 (27.316 sec)
45.317 (99.46%)... logprob: 4.330134, 0.812174, 0.625651 (27.314 sec)
45.318 (99.47%)... logprob: 4.296220, 0.807617, 0.608724 (27.298 sec)
45.319 (99.47%)... logprob: 4.265041, 0.798503, 0.600260 (27.456 sec)
45.320 (99.48%)... logprob: 4.286528, 0.804036, 0.618164 (27.313 sec)
```

Figure 25. 8-Layers: Training Time Around 27.350 Seconds, 19 Seconds more Training Time, Compared to that of 6-Layered Model

When compared to DeepFace algorithm, Feature Parallelism model has substantially improved training time up to 21%. This implies that parallel processing is of a central role, especially for acceleration, in both the human visual system and the biologically inspired computer vision systems. When Feature Parallelism Model was investigated regarding depth, training time has dropped by few milliseconds while going from 5 to 6 layers; and error rate slightly drops by 1%. However, when going from 6 to 7 layers, training time increases significantly from 8.080 seconds to 24.060 seconds; almost three times that of the 6- layers model. It continues to increase by a smaller rate when going from 7 to 8 layers to around 27.350 seconds. Similarly, top-1 error rate increases from 68% to 73%, a 5% increase when going from 6 to 7 layers; and top- 5 error rate increases from 45% to 52%, a 7% increase. Also, when going from 7 to 8 layers we have noticed a 5% increase in top-1 error rate from 73% to 78%, and a 6% increase in top-5 error rate from 52% to 58%. The optimum number of layers has been found to be six layers. Many studies have been conducted to decipher the wisdom behind cortical columns and the six-layered architecture of the cortex. Many theories have been developed. In this study we suggest a one possible wisdom, that is, the organization of parallel processing units into six- layers architecture appears to significantly enhance processing time and optimizes accuracy rates.

4. Conclusion

The significance of this study is that it has suggested an optimum number of layers for deep architectures. Under Feature Parallelism model, we don't need to choose and adjust the number of layers as a hyper- parameter; we've found out that six layers is optimum, and this is biologically plausible.

Acknowledgment

We'd like to thank MIMOS Berhad, Malaysia for supporting this work and allowing us to use their equipment. Special thanks to Azhar Bin Abu Talib, the director of Network Security lab, who made that possible.

References

- [1] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", Nips. doi:http://dx.doi.org/10.1016/j.protcy.2014.09.007, (2012), pp. 1-9.
- [2] Y. Taigman, M. A. Ranzato, T. Aviv and M. Park, "DeepFace : Closing the Gap to Human-L evel Per for mance in Face Verification Abstract", in: 2014 IEEE Conf. Comput. Vis. Pattern Recognit., IEEE,

- Columbus, OH, doi:10.1109/CVPR.2014.220, (2014), pp. 1701-1708.
- [3] M. Y. Hassan, O. O. Khalifa, A. Abu and A. H. Abdulla, "Unconstrained Facial Recognition Systems: A Review Unconstrained Facial Recognition Systems", *A Review, Asian J. Appl. Sci.*, vol. 3, (2015), pp. 346-354.
- [4] Y. H. Marwa, O. K. Othman, B. A. T. Azhar, F. O. Rashidah and H. A. Aisha, "A novel neuroscience-inspired architecture: For Computer Vision Applications", in: 2016 Conf. Basic Sci. Eng. Stud., IEEE, Khartoum, doi:10.1109/SGCAC.2016.7458013, (2016), pp. 101-109.
- [5] K. Nakayama and G. H. Silverman, "Serial and parallel processing of visual feature conjunctions", *Nature*, doi:10.1038/320264a0, vol. 320, (1986), pp. 264-265.
- [6] A. M. Treisman and A. Treisman, "A Feature-Integration Theory of Attention", *Cogn. Psychol.*, doi:10.1016/0010-0285(80)90005-5, vol. 12, (1980), pp. 97-136.
- [7] A. Treisman, "Perceptual grouping and attention in visual search for features and for objects", *Journal Exp. Psychol. Hum. Percept. Perform.* doi:10.1037/0096-1523.8.2.194, vol. 8, (1982), pp. 194-214.
- [8] P. Quinlan, "Formats: Visual Feature Integration Theory: Past, Present, and Future", *Contents By: Philip T. Quinlan, Psychol. Bull.*, vol. 129, (2003), pp. 1-50.
- [9] J. Bullier and L. G. Nowak, "Parallel versus serial processing: new vistas on the distributed organization of the visual system", *Curr. Opin. Neurobiol.* doi:10.1016/0959-4388(95)80011-5, vol. 5, (1995), pp. 497-503.
- [10] J. J. Nassi and E. M. Callaway, "Parallel Processing Strategies of the Primate Visual System", doi:10.1038/nrn2619. *Parallel*, vol. 10, (2009), pp. 360-372.
- [11] Y. LeCun, K. Kavukcuoglu and C. Farabet, "Convolutional networks and applications in vision", in: *Proceeding 2010 IEEE Int. Symp. Circuits Syst.*, IEEE, Paris, doi:10.1109/ISCAS.2010.5537907, (2010), pp. 253-256.
- [12] S. Zeki, "A massively asynchronous, parallel brain", *Philos. Trans. R. Soc. B Biol. Sci.*, doi:10.1098/rstb.2014.0174, vol. 370, (2015), pp. 20140174-20140174.
- [13] S. Grossberg, "Laminar cortical architecture In Visul Perception", in: M.A. Arbib (Ed.), *Handb. Brain Theory Neural Networks*, Second Edi, THE MIT PRESS Cambridge, Massachusetts, London, (2003), pp. 594-600.
- [14] R. D. S. Raizada, "Towards a Theory of the Laminar Architecture of Cerebral Cortex: Computational Clues from the Visual System", *Cereb. Cortex*, doi:10.1093/cercor/13.1.100, vol. 13, (2003), pp. 100-113.
- [15] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *ImageNet Chall*, doi:10.1016/j.infsof.2008.09.005, (2014), pp. 1-10.
- [16] S. M. Plis, D. R. Hjelm, R. Slakhutdinov, E. A. Allen, H. J. Bockholt and J. D. Long, "Deep learning for neuroimaging", A validation study, *Front. Neurosci.* doi:10.3389/fnins.2014.00229, vol. 8, (2014), pp. 1-11.
- [17] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures", *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*. 7700 LECTURE NO, doi:10.1007/978-3-642-35289-8-26, (2012), pp. 437-478.
- [18] V. B. Mountcastle, "The columnar organization of the cerebral cortex", *Brain*, vol. 120, (1997), pp. 701-722.
- [19] J. C. Horton and D. L. Adams, "The cortical column: a structure without a function", *Philos. Trans. R. Soc. B Biol. Sci.*, doi:10.1098/rstb.2005.1623, vol. 360, (2005), pp. 837-862.
- [20] N. Maçarico, K. A. C. Martin and K. S. Rockland, "Whose cortical column would that be", doi:10.3389/fnana.2010.00016, vol. 4, (2010), pp. 1-10.

