Trends in deep convolutional neural Networks architectures: a review

1st Azeddine Elhassouny
ENSIAS, Mohammed V University in Rabat
Rabat, Morocco
azeddine.elhassouny@um5.ac.ma

2nd Florentin Smarandache
Dept. Mathematics,
University of New Mexico
New Mexico, USA
smarand@unm.edu

***Abstract*—Deep convolutional Neural networks(CNN) has recognized much advances in recent years. Many CNN models have been proposed in few years ago which focused by first on improving accuracy, next minimize number of parameters using squeeze architecture, then CNN model adapted for embedded and mobile systems. But face the huge applications of CNN in computer vision, few papers discuss what is the theory behind building CNN models combining some components. In this paper, we present a survey of recent advances in CNN architecture design taking into account the three periods listed above.**

***Index Terms*—Deep convolutional Neural networks(CNN),MobileNet, ResNeXt, ResNet, NiN, Review: SqueezeNet, GoogLeNet, VGGNet, AlexNet, LeNet5, Neocognitron, Survey, artificial intelligence.**

I. Introduction

Even if many application studies of Deep CNN algorithms in many computer vision tasks, and their tremendous advances by getting an accuracy close to or over better than human level perception [1]. However, there are too few studies about its theory and all researchers try to build CNN model by experimentation error and architecture model copying.

There are a large number of papers on CNN, but recording to some famous challenges hosted by ImageNet, COCO, etc, we can conclude that Compressed SqueezeNet [2], SqueezeNet [3], ResNeXt (Aggregated Residual Transformations for Deep Neural Networks) [4], ResNet (Deep Residual Learning for Image Recognition) [5], NiN (Network In Network) [6], GoogLeNet [7], VGGNet [8], ZFNet [9], AlexNet [10] are the standard designed architectures of CNN, and all papers produced in this area are just application of one standard architecture cited above. The reviewed papers here are on supervised deep CNN, for unsupervised deep learning, and in other work [11], we reviewed the main three generative models PixelRNN, DRAW and NADE, thus the building components of recent deep generative models, such as RBM, DBM, DBN, VAE and GAN.

In this paper, we present a survey of recent advances in CNN architectures design. The history of deep CNN from 1980 with Fukushima Neocognitron to present-day can be broken to three eras.

First period, named Limited CNN, that was between Fukushima Neocognitron CNN and 2009, is characterized by the lack of capacity of storage of computers, computation resources, and training data.

Second period, that we called huge CNN, in which researchers look for higher accuracy, regardless number of parameters, layers and neurons. In this period researchers benefit from the powerful capacity of storage and computation resources of new computers. It began and characterized by contribution of Alex and al. called AlexNet [10]. The proposed AlexNet model won the ImageNet challenge, it was the first CNN architecture with many layers(8 layers) and got a remarkable improvement upon shallow CNN.

Since this date several architectures have been proposed to improve performance and minimizing accuracy errors. Among them, a yearly architecture winners of ImageNet challenge have proposed such as ZFNet [9], GoogLeNet [7], VGGNet [8], ResNet [5], that trained on very deep networks (from 152 to up 1200 layers). The ResNet model increases complexity of algorithms to be optimized, grow up exponentially the number of parameters, but get error accuracy (3.57%) better than human rate that is around 5%.

Recently, due to the lack of computing power and energy, memory storage capacity of mobile devices and embedded systems, many squeeze and hardly adapted CNN have been proposed. This kind of CNN leads four main challenges: model size, speed, energy efficiency and memory storage capacity, to achieves the flowing objectives : fast CNN algorithms based on few parameters ($510\times$ smaller than AlexNet) with minimum error accuracy and then able to be embedded in mobile systems.

II. Convolutional Neural Networks Architectures Design

The network architecture design is the general structure of the network, that specify the number of unites and the connections between each groups of unites (layers).

CNN architecture is an organization of a number of convolutional, activation, pooling, and other layers in one skeleton. The overall pattern of the whole CNN reply to the following questions, how many layers, how many units in each layer, and how units are connected each to other.

The values of each layers is calculated by function of its predecessor layers. The first layer is given by

$$h^{(1)} = g^{(1)}(W^{(1)}x + b^{(1)})$$

978-1-7281-0827-8/19/$31.00©2019 IEEE

the following layers are given by

$$h^{(i+1)} = g^{(i+1)}(W^{(i+1)}h^{(i)} + b^{(i+1)})$$

where x is input training data, w weight,

In this section the most popular CNN design models will be listed. Furthermore than list architectures model with their characteristics and pair comparisons respecting chronological order, we will bring out some theoretical rules behind each architecture improvement. However, the draw of some mathematical rules is complicated, because of all famous architectures are established and developed for a specific problems [12] and few papers treat mathematical theory of CNN.

A. More theory of CNN Architectures Design with no practice

CNN procedures have been used since the late 1960, but until 2009, they were mostly limited due to the lack of storage capacity and resources computation. Those constraints affect the use of CNN particularly (Neural networks generally), that conclusion is proved by the number of contributions, number of layers, and accuracy error of such algorithms.

a) Neocognitron: Fukushima et al. [13]–[15] introduced the notion of neural networks and CNN model for mechanism of pattern recognition, based on some earliest predecessors contributions in this field. Nevertheless, this chronological survey is started by $neocognitron$ as the first CNN architecture model.

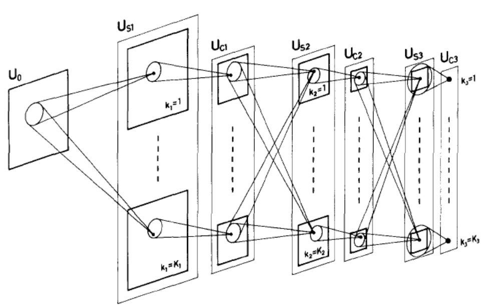

Fig. 1. neocognitron intersection layers

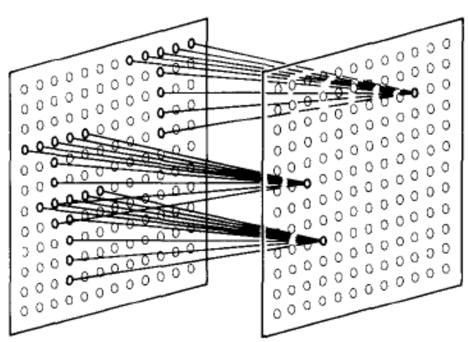

Fig. 2. neocognitron input and output in successive layers [13]

b) From self-organizing to propagation and back-propagating: In 1986, Rumelhart et al. [16], [17] developed, instead of self-organizing, the notions of propagation and back-propagating techniques in learning representations(figure 3). Figure 4 show, for instance, first mapping of these techniques.

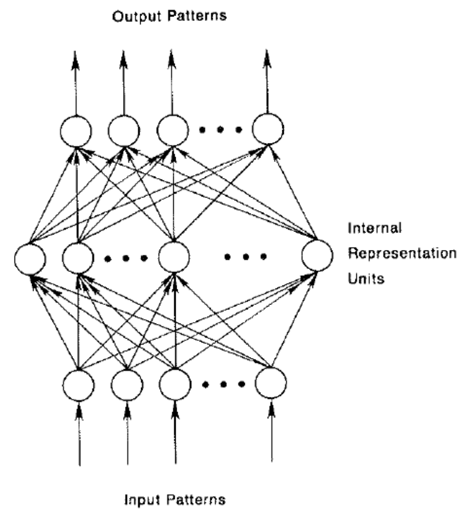

Fig. 3. Learning representations

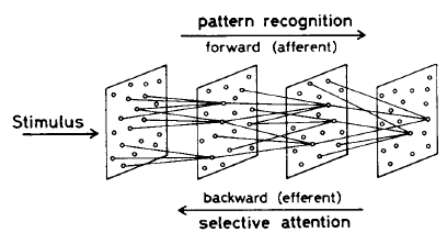

Fig. 4. First mapping of propagation and back-propagating techniques

c) A Neural Network for Visual Pattern Recognition: We imitating the definition of CNN given by Fukushima and al. [18], that have defined, at first, CNN as : "Model of the neural network offers insight into the brain's complex mechanisms as well as design principles for new information processors. Fukushima et al. [18] again have used instead of self-organizing, back-propagation and forward-propagation in 7-layers Neural networks for recognizing digit as shown in the figure 5 and 6.

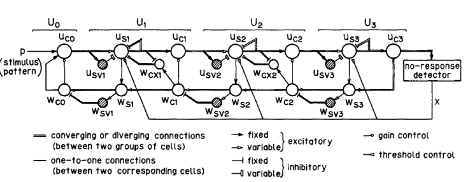

Fig. 5. Neural Network for Visual Pattern Recognition

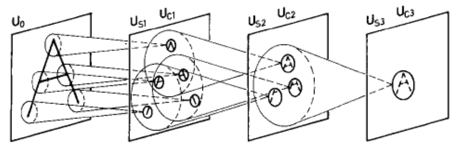

Fig. 6. Example of Neural Network for Visual Pattern Recognition

d) LeNet-5: LeNet−5 [19] is a CNN defined as Gradient−based learning algorithm, designed for handwritten document recognition. LeNet-5 consists of 7 layers : $C1 \rightarrow S2 \rightarrow C3$ The following figure show LeNet-5 CNN model for isolated character recognition

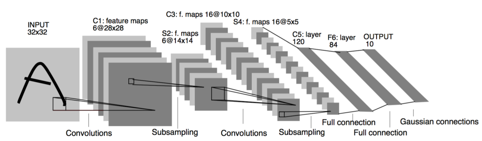

Fig. 7. LeNet-5 CNN model

B. Between 2010 and 2015 : CNN with Higher accuracy regardless number of parameters

in this period, due to exponential progress of powerful capacity of storage and computing of modern computers, the researchers developed many architectures that achieve high accuracy as or even more human.

a) AlexNet: AlexNet model was developed in framework ImageNet Large-Scale Visual Recognition Challenge 2012, which was the first CNN had has $60M$ parameters and $500,000$ neurons. To overcome that problem of big number of parameters, Alex et al. [10] introduced dropout method to reduce Overfitting in CNN connected layers. Authors won the challenge and achieved $top-1$ and $top-5$ error rates of 39.7% and 18.9% respectively, which better than the previous state of the art results.

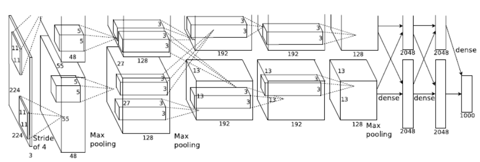

Fig. 8. AlexNet CNN model

AlexNet model consists of 11 simple basic layers(8 composed layers, for more see $makereference$) : 5 convolutional, 3 max-pooling, 2 Normalization, and 3 fully connected layers(FC) (include final $1000 - way$ Softmax).

$CONV1 \rightarrow MAXPOOL1 \rightarrow NORM1 \rightarrow CONV2 \rightarrow MAXPOOL2 \rightarrow NORM2 \rightarrow CONV3 \rightarrow CONV4 \rightarrow CONV5 \rightarrow MaxPOOL3 \rightarrow FC6 \rightarrow FC7 \rightarrow FC8$

b) ZFNet: Matthew and al. [9] introduced a novel diagnostic visualization technique to understand why and how CNN performs well. This technique is applied on AlexNet architecture [10] which allows authors to ameliorate CNN architecture that outperform the existed model. The established ZFNet model (summarized in figure 9) retains the same number of AlexNet's layers but improve the error rates to 11.7%, and therefore Won the ImageNet Large-Scale Visual Recognition Challenge 2013 (ILSVRC 2013).

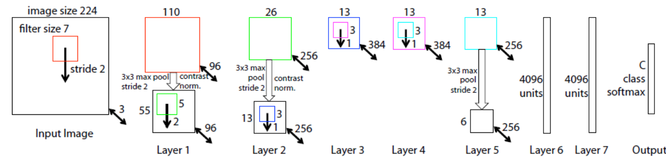

Fig. 9. ZFNet model

c) VGGNet: Simonyan et al. [8] generalized AlexNet model, by depth increasing from 8 layers(AlexNet) to $16-19$ layers (VGG16Net), with few parameters, instead of 7×7 a stack of three 3×3 filters for convolution layers are used. The both findings diminishes the rate error from 11.7% (top 5 error of ZFNet) to 7.3% top 5 error, and allows first place in the localization and second place in classification in ILSVRC14 challenge.

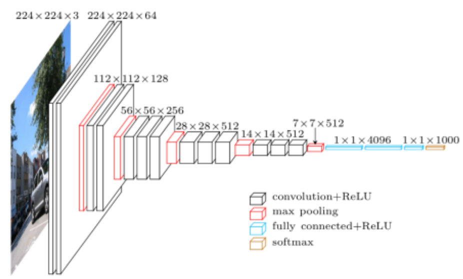

Fig. 10. VGGNet model

$[CONV3 \rightarrow CONV3 \rightarrow POOL2]^2 \rightarrow [CONV3 \rightarrow CONV3 \rightarrow CONV3 \rightarrow POOL2]^3 \rightarrow FC \rightarrow FC \rightarrow FC$

d) GoogLeNet: Going Deeper with Convolutions architecture or nicknamed GoogLeNet [7], that won the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC 2014), is hallmarked by applying parallel filter operations, benefiting of the computing resources. It's a series of multi-scale processing with concatenation of all filter. Szegedy et al. [7] introduced a new level of organization, we can say a new topology of CNN, different than all previous CNN models, moreover increased network depth to 22 layers, compared to all architecture since 2010.

e) NiN: Network In Network: NiN [6] consists on Mlpconv (multilayer perceptron (MLP) convolutional) layer and the global averaging pooling layer. Mlpconv includes a micro network, which built with more complex structures to abstract the data within receptive field.

Furthermore, a study [20] on visual interpretation of network in network has been built to point out the relationships between the processing of NiN and orientation continuity(operation of a mammalian).

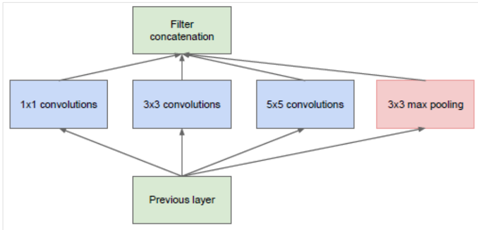

Fig. 11. Inception module, or GoogLeNet

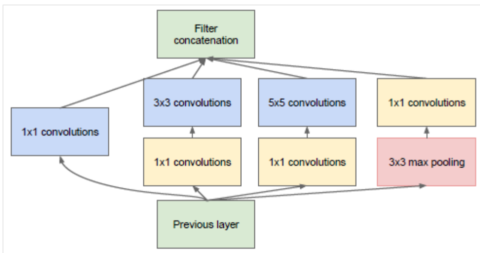

Fig. 12. Inception module with dimensionality reduction

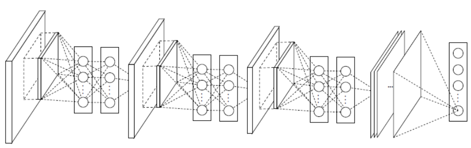

Fig. 13. Network In Network

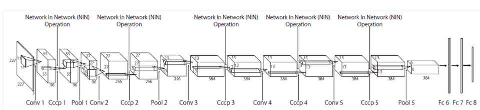

Fig. 14.

As depicted on the figure 15, CCCP operation take output of the convolution layer and apply weighted linear superposition.

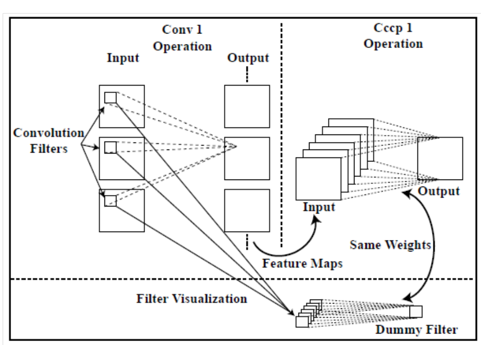

Fig. 15.

f) ResNet : In Deep Residual Learning for Image Recognition codenamed ResNet [5], to overcome the bottleneck of Deep neural networks, which is training phase [5] redefined the layers as learning residual functions. The usefulness of residual nets showed its capacity to train the CNN ResNet with a depth up to 152 layers instead of 22 in GoogLeNet, with grew up of $8\times$ and lower complexity. This architecture secured the first place in ImageNet challenge (ILSVRC 2015) with 3.57% error, which for first time, the CNN achieve error rate more better than human perception.

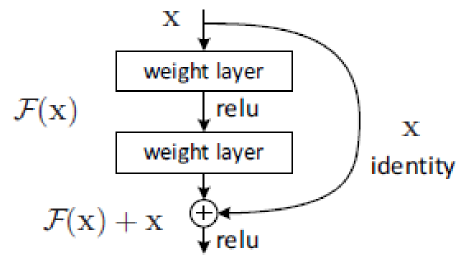

Fig. 16. Residual learning: a building block

The hallmark problem of very deep learning residual networks is its high depth, which slows the training step. To overcame that problem, authors Zagoruyko et al. [21] proposed wide residual networks (WRNs). The Wide Residual Networks characterized by its small depth have used for mitosis detection in breast histology images [22].

Larsson et al. [23] proposed a simple alternative to ResNet, FRACTALNET: Ultra-deep neural networks without residuals, and proved that explicit residual learning is not a requirement for building ultra-deep neural networks. Furthermore, they introduced FRACTALNET's regularization strategy named drop-path.

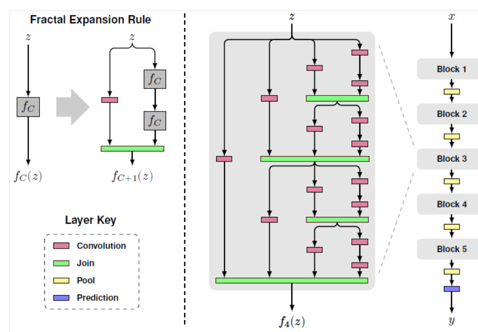

Fig. 17. FRACTALNET model

Let $f_C(\cdot)$ a truncated fractal with C its index, the convolutional layer is defined as $f_1(z) = conv(z)$, and the successive fractals recursively is formulated as

$$f_{C+1}(z) = [(f_C \circ f_C)(z)] \bigoplus [conv(z)]$$

with $\circ$ denotes composition and $\bigoplus$ join operation.

g) ResNeXt : Aggregated Residual Transformations for Deep Neural Networks: in addition to dimensions of depth and width, Xie and al. [4] exposed the cardinal of the set of transformations dimension. Below, figures 18 and 19 present ResNeXt and its equivalent blocks, aggregated residual transformations respectivly.

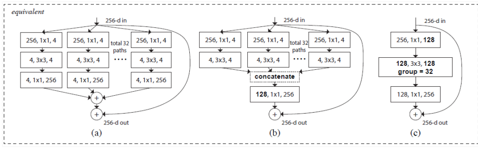

Fig. 18. ResNeXt and its equivalent blocks

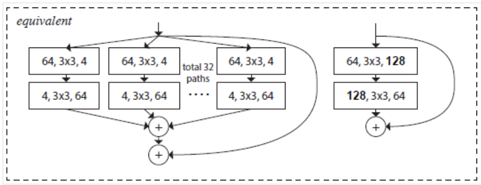

Fig. 19. ResNeXt and its equivalent blocks of depth =2

C. Hybrid CNN Architecture

a) MRF-CNN: To synthesis images, Li and al. [24] studied the combination of Markov Random Fields(MRF) and Convolutional Neural Networks. The produced method transfer both photorealistic and non-photorealistic styles to new images.

b) RIFD-CNN: Cheng et aL. [25] proposed a new method RIFD-CNN (rotation-invariant and Fisher discriminative) to address the problem of recognizing rotate object. The model is built by integrating in basic CNN architecture tow kind of layers rotation-invariant (figure 20) and Fisher discriminative (figure 21).

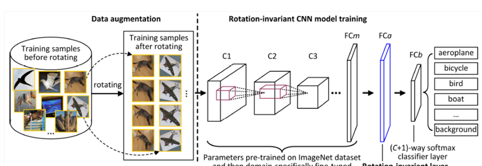

Fig. 20. CNN with rotation-invariant layer

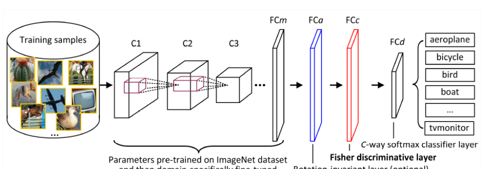

Fig. 21. CNN with Fisher discriminative layer

c) TI-POOLING: The same have made by Laptev and al. [26], in building a method named TI-POOLING (transformation-invariant pooling). The model consists on : first grow up the data, adding a set of transformed images obtained by applying a set of transformations, second : after initializing network, a series of convolutional and subsampling layers are used (figure 22).

d) $CNN + DPCL$: Integration of DPCL and CNN: Wang and al. [27], as depicted in figure 23 included

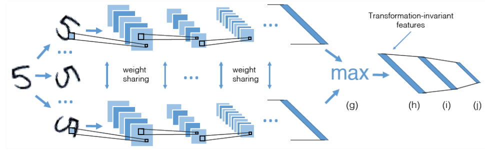

Fig. 22. transformation-invariant pooling

DPL(Dictionary Pair Layer) as a Dictionary Pair Classifier Layer (DPCL) into the CNN deep architecture. Multi-loss CNN+DPCL model based on convolutional layers, fully connected layers, the final score is the combination of objectiveness and categoryness of DPCL.

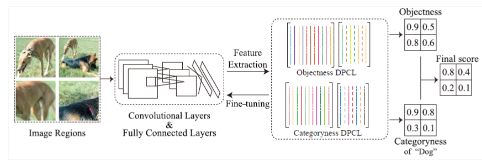

Fig. 23. CNN+DPCL flowchart

e) LF-CNN : Latent Factor Guided Convolutional Neural Networks : The LF-CNN model [28] is composed of two key types of layers: some convolutional layers and Latent Factor-Fully Connected(LF-FC) layer. The architecture of LF-CNNs is shown in Figure bellow 24:

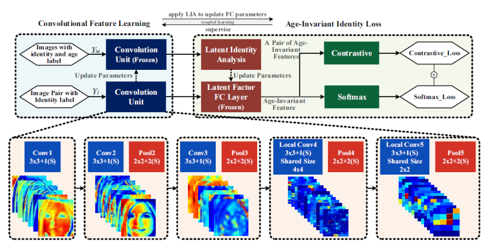

Fig. 24. LF-CNN model

D. Refining architectures of CNN : Squeeze, Embedded and mobile architectures

Table I below, benchmarks [29] for popular convolutional neural network models on CPU and different GPUs, with and without cuDNN. Speed in the table is the total time for a forward and backward pass on a Pascal Titan X with cuDNN 5.1. As showed, most CNN architectures and algorithms focused on increasing accuracy regardless depth and width (number of units and parameters).

To build a fast and squeeze network for mobile platforms and embedded systems, for a given task, all searchers, firstly, try out one of renowned CNN such as AlexNet, GoogleNet, VGG [30], and secondly, attempt to reduce the size of taht model with keeping highly accuracy architecture.

a) how are the accuracy and parameters number are related :: But one should decide for a size reduction of an architecture, how much rate losing of an accuracy. Parameters

TABLE I
BENCHMARKS FOR POPULAR CONVOLUTIONAL NEURAL NETWORK MODELS

Network	Layers	Top-1 error	Top-5 error	Speed (ms)	Citation
AlexNet	8	42.90	19.80	14.56	[1]
Inception-V1	22	-	10.07	39.14	[2]
VGG-16	16	27.00	8.80	128.62	[3]
VGG-19	19	27.30	9.00	147.32	[3]
ResNet-18	18	30.43	10.76	31.54	[4]
ResNet-34	34	26.73	8.74	51.59	[4]
ResNet-50	50	24.01	7.02	103.58	[4]
ResNet-101	101	22.44	6.21	156.44	[4]
ResNet-152	152	22.16	6.16	217.91	[4]
ResNet-200	200	21.66	5.79	296.51	[5]

number and accuracy are indirectly proportional to each other. The parameters number decreases, the accuracy also increases, and as the parameters number increases, the accuracy decreases. As despite on figure ([**?**])

b) Deep compression: To Keep the same accuracy of CNN with reducing their requirement storage, Han and al. [31] made work together three tools : pruning, trained quantization and huffman coding.

Deep compression CNN established achieves : $3\times$ speedup on CPU, $3.5\times$ on GPU and $4.2\times$ on mobile GPU on average (as shown in figure25). $7\times$ less energy on CPU, $3.3\times$ less on GPU and $4.2\times$ less on mobile GP on average (as shown in figure26). $35\times$ to $49\times$ in storage requirements.

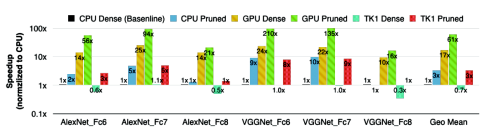

Fig. 25. speedup on CPU, GPU and mobile GPU on average

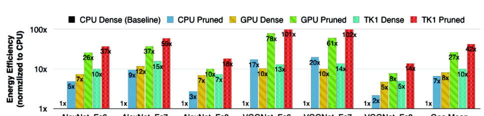

Fig. 26. Energy on CPU, GPU and mobile GPU on average

c) Stretching or Splitting strategy: Shankar et al. [30] Refining Architectures of Deep Convolutional Neural Networks, means looking to reply the following question : How to made up a CNN architecture that have few parameters while maintaining competitive accuracy? To answer that unresolved question, i.e, made up a refined CNN architecture, the authors [30] decide optimally on a strategy of stretching or splitting (figure 27).

The proposed approach, applied on GoogleNet and VGG-11 architectures, with CAMIT and SAD dataset, got a reduction

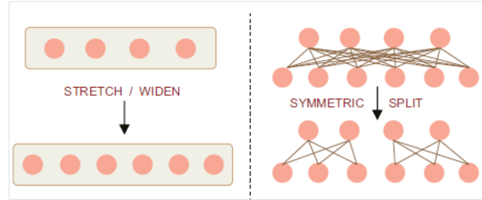

Fig. 27. Stretching or Splitting strategy

in model size and an increase in precision (figure 5, page 7 in [30]).

d) SqueezeNet: Iandola and al. [3] proposed a small CNN architecture called SqueezeNet (figure 28), that achieves AlexNet-level accuracy on ImageNet with $50x$ fewer parameters, and with model compression techniques got less than $0.5MB$ i.e ($510x$ smaller than AlexNet).

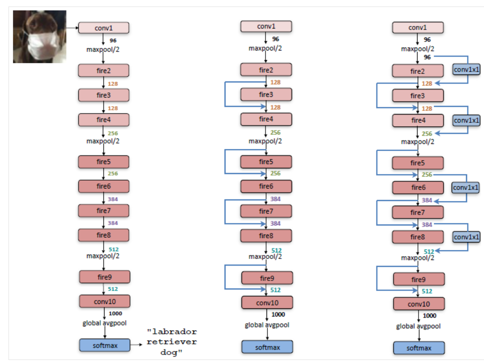

Fig. 28. SqueezeNet model

To made up that CNN architectures, which have few parameters while maintaining competitive accuracy, three main strategies (figure 29) are used :

- Strategy 1. Replace $3x3$ filters with $1x1$ filters, since a $1x1$ filter has $9X$ fewer parameters than a $3x3$ filter
- Strategy 2. Decrease the number of input channels to $3x3$ filters, because of $3x3$ filters. The total quantity of parameters in this layer is $(number of input channels) * (number of filters) * (3*3)$. decrease the number of input channels to $3x3$ filters using squeeze layers
- Strategy 3. Downsample late in the network so that convolution layers have large activation maps. the intuition of [3] is that large activation maps (due to delayed downsampling) can lead to higher classification accuracy confirmed by the research of [32]

The main role of squeeze layer is to limit the number of input channels to the $3x3$ filters by fitting s_{1x1} to be less than (e_{1x1} + e_{3x3}), as shown in figure 29.

Table II [2] present the comparison for squeezed and standard architectures. Iandola and al. [2] proved that compressed approach got the AlexNet-level accuracy with $50x$ fewer parameters and $< 0.5MB$ model size.

Many variants and adapted Squeeze algorithms and methods have been proposed last years, such as Squeeze-SegNet [33]

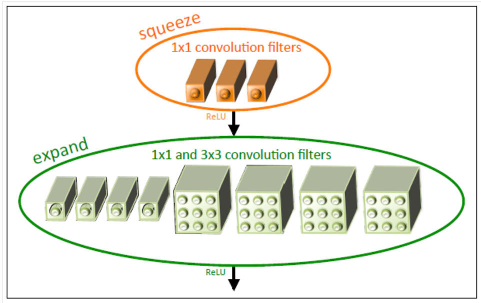

Fig. 29. Squeeze's strategies

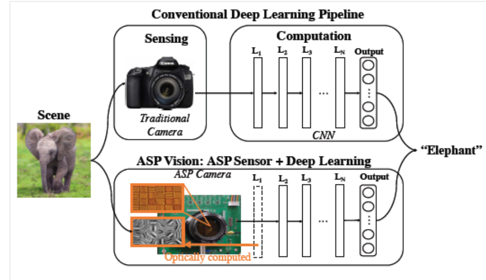

Fig. 30. CNN with backend Angle Sensitive Pixels (CNN-ASP)

TABLE II
BENCHMARKS FOR POPULAR CNN MODELS

CNN model	Compression Approach	Data Type	Original Compressed Model Size	Reduction in Model Size vs. AlexNet	Top-1 ImageNet Accuracy	Top-5 ImageNet Accuracy
AlexNet	None (baseline)	32 bit	240MB	$1x$	57.2%	80.3%
AlexNet	SVD (Denton et al.,2014)	32 bit	$240MB \longrightarrow 48MB$	5	56.0%	79.4%
AlexNet	Network Pruning (Hanet al., 2015b)	32 bit	$240MB \longrightarrow 27MB$	$9x$	57.2%	80.3%
AlexNet	Deep Compression (Hanet al., 2015a)	5-8 bit	$240MB \longrightarrow 6.9MB$	$35x$	57.2%	80.3%
SqueezeNet	None	32 bit	4.8MB	$50x$	57.5%	80.3%
SqueezeNet	Deep Compression	8 bit	$4.8MB \longrightarrow 0.66MB$	$363x$	57.5%	80.3%
SqueezeNet	Deep Compression	6 bit	$4.8MB \longrightarrow 0.47MB$	510x	57.5%	80.3%

a deep fully convolutional neural network for pixel-wise semantic segmentation, in which authors have proposed squeeze-decoder module and used SqueezeNet-like encoder.

E. Advances in CNN Architectures : Embedded and mobile systems CNN

The application of deep convolutional networks in embedded and mobile systems is limited by algorithm's power consumption, bandwidth requirements, tight energy budgets, requirement of very low latency and limited processing resources [34], [35].

To solve these challenges, many algorithms are proposed: Lavin and al. [34] introduced a fast CNN based on Winograd's minimal filtering algorithms (Winograd's algorithm), which a generalization of filtering algorithms of Toom and Cook.

However Chen and al. [35] proposed a combination of ASP sensor with CNN backend Angle Sensitive Pixels (ASP)(figure 30). ASP is a bio-inspired CMOS image sensors, that have Gabor wavelet impulse responses, and perform optical convolution for the CNN first layer. The idea is based on : the hardcoding of first layer leads to significant energy savings.

Wu and al. [36] presented Quantized-CNN for Mobile Devices, which is a framework of convolutional neural network (CNN), and that achieves $3.03\times$ speed-up against the Caffe implementation.

a) MobileNet Model: A team of Google researchers introduced an efficient models named MobileNets [37] for mobile and embedded vision applications. MobileNets is the recent CNN model built able to work in Mobile phone, based on two main operations : depth-wise separable filters and factorization. Tacking advantage of wide use of mobile phone, several applications of MobileNet have been proposed covering many lives activities, such as agriculture [38], factory, business, etc.

III. Conclusion

Until 2017, there are three major periods in CNN history, the period of lack of capacity of storage and computation resources, the period of highest accuracy, regardless number of parameters and the period of embedded CNN model on Mobile phone.

However, there is no rule or theory how to build CNNs models, but researchers proceed by experimentation and model design copying. Experimentally, CNN with large depth gives the better results and high accuracy.

Acknowledgment

The authors would like to thank Prof. Florentin Smarandache, professor of mathematics, University of New Mexico, 705 Gurley Ave., Gallup, New Mexico 87301, USA, E-mail: fsmarandache@gmail.com, site : http://fs.gallup.unm.edu/ for hosted me as postdoctoral researcher at his University of New Mexico. I would like to think the Fulbright program for support me to stay at UNM, US for three months.

References

[1] D. Mishkin, N. Sergievskiy, and J. Matas, "Systematic evaluation of cnn advances on the imagenet," *arXiv preprint arXiv:1606.02228*, 2016.
[2] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, vol. abs/1602.07360, 2016. [Online]. Available: http://arxiv.org/abs/1602.07360
[3] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.
[4] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *arXiv preprint arXiv:1611.05431*, 2016.
[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
[6] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.
[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
[8] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[9] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[11] A. Oussidi and A. Elhassouny, "Deep generative models: Survey," in *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*. IEEE, 2018, pp. 1–8.

[12] M. Ranzato, "Supervised deep learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[13] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 3, pp. 193–202, 1980.

[14] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[15] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, vol. 1, no. 2, pp. 119–130, 1988.

[16] D. E. Rumelhart, G. E. Hinton, R. J. Williams *et al.*, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[17] G. E. H. David E. Rumelhart and R. J. Williams, "Learning representations error propoagation," *Nature*, vol. 323, pp. 533–536, 1986.

[18] K. Fukushima, "A neural network for visual pattern recognition," *Computer*, vol. 21, no. 3, pp. 65–75, 1988.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[20] S. Suzuki and H. Shouno, "A study on visual interpretation of network in network," in *Neural Networks (IJCNN), 2017 International Joint Conference on*. IEEE, 2017, pp. 903–910.

[21] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[22] E. Zerhouni, D. Lányi, M. Viana, and M. Gabrani, "Wide residual networks for mitosis detection," in *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on*. IEEE, 2017, pp. 924–928.

[23] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," *arXiv preprint arXiv:1605.07648*, 2016.

[24] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2479–2486.

[25] G. Cheng, P. Zhou, and J. Han, "Rifd-cnn: Rotation-invariant and fisher discriminative convolutional neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2884–2893.

[26] D. Laptev, N. Savinov, J. M. Buhmann, and M. Pollefeys, "Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 289–297.

[27] K. Wang, L. Lin, W. Zuo, S. Gu, and L. Zhang, "Dictionary pair classifier driven convolutional neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2138–2146.

[28] Y. Wen, Z. Li, and Y. Qiao, "Latent factor guided convolutional neural networks for age-invariant face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4893–4901.

[29] J. Johnson, "Benchmarks for popular cnn models," in *https://github.com/jcjohnson*, 2016.

[30] S. Shankar, D. Robertson, Y. Ioannou, A. Criminisi, and R. Cipolla, "Refining architectures of deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2212–2220.

[31] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[32] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5353–5360.

[33] G. Nanfack, A. Elhassouny, and R. O. H. Thami, "Squeeze-segnet: a new fast deep convolutional neural network for semantic segmentation," in *Tenth International Conference on Machine Vision (ICMV 2017)*, vol. 10696. International Society for Optics and Photonics, 2018, p. 106962O.

[34] A. Lavin and S. Gray, "Fast algorithms for convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4013–4021.

[35] H. G. Chen, S. Jayasuriya, J. Yang, J. Stephen, S. Sivaramakrishnan, A. Veeraraghavan, and A. Molnar, "Asp vision: Optically computing the first layer of convolutional neural networks using angle sensitive pixels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 903–912.

[36] Y. W. Q. H. Jiaxiang Wu, Cong Leng and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[38] E. Azeddine and F. Smarandache, "Smart mobile application to recognize tomato leaf diseases using convolutional neural networks," in *2nd International Conference of Computer Science and Renewable Energies*. IEEE, 2019.