

# Hybrid Algorithm for the Optimization of Training Convolutional Neural Network

Hayder M. Albeahdili

Dep. of Electrical and computer  
Engineering  
University of Missouri, Columbia  
Columbia, Missouri, 65211, USA

Tony Han

Dep. of Electrical and computer  
Engineering  
University of Missouri, Columbia  
Columbia, Missouri, 65211, USA

Naz E. Islam

Dep. of Electrical and computer  
Engineering  
University of Missouri, Columbia  
Columbia, Missouri, 65211, USA

**Abstract**—The training optimization processes and efficient fast classification are vital elements in the development of a convolution neural network (CNN). Although stochastic gradient descend (SGD) is a Prevalence algorithm used by many researchers for the optimization of training CNNs, it has vast limitations. In this paper, it is endeavor to diminish and tackle drawbacks inherited from SGD by proposing an alternate algorithm for CNN training optimization. A hybrid of genetic algorithm (GA) and particle swarm optimization (PSO) is deployed in this work. In addition to SGD, PSO and genetic algorithm (PSO-GA) are also incorporated as a combined and efficient mechanism in achieving non trivial solutions. The proposed unified method achieves state-of-the-art classification results on the different challenge benchmark datasets such as MNIST, CIFAR-10, and SVHN. Experimental results showed that the results outperform and achieve superior results to most contemporary approaches.

**Keywords**—Convolutional Neural Network; Particle Swarm optimization; Image Classification

## I. INTRODUCTION

The Convolutional Neural Network (CNN) algorithm has been widely applied in many applications, including face recognition [1, 2], image classification and recognition [3-6] and object detection [7]. In supervised learning, Back Propagation (BP) algorithm is the prevalence and constituent method used for CNN training and parameters tuning. All researchers used it in CNN training in all their implementations.

However, there are a number of disadvantages of using the back propagation algorithm alone. For example, BP algorithm deterministically occurs in local optima, making it hard to get global optima, especially if a large search space is required for optimal solution. The algorithm is also slow and hardly benefits of using modern machines such as Graphics Processing Unit (GPUs), which runs hundreds to thousands of threads simultaneously. The complex computational equations emerging in the algorithm demand hard and complicated series of steps to find derivative equations for updating weight parameters. Finally, the cardinality of back propagation algorithm recruits intermediate variables to preserve the validity of data. Means, the implication of BP requires keeping forward and backward essential parameters used for updating equations.

To tackle limitations mentioned accompanied with BP algorithm, in this paper an alternate algorithm is proposed for CNN training. In particular, the Particle Swarm Optimization (PSO) algorithm is introduced for training; and it is combined with the Stochastic Gradient Descent (SGD) to achieve better results. The computational algorithm proposed delves to avoid occurring in local optimum, is fully parallel, and induces simple equations for CNN training. It is completely adaptable because it does not require any changes in CNN structure when some network layers are added or eliminated. The PSO equations used for training weights are completely parallelize as described in (1) and (2) and shown in fig. 2. This suggests that the weights of any layer can be updated without the need for backward phase as in SGD, thus GPUs can be completely utilized using this implementation. The proposed method also improves training by overcoming premature saturation and sluggishness inspired by SGD.

The reset of the paper consist of the following: in section II, related works are introduced. In section III a brief introduction of introducing PSO is presented. Then in section IV, the proposed approach is introduced in details. Then in section V the model architecture of CNN is illustrated. In both second VI and VII, challenge benchmark used for model evaluation and conclusion are depicted respectively.

## II. RELATED WORK

Recently there are vast number of research have been proposed for image recognition using different methods and several proposed novel methods are proposed. Generally image recognition can be obtained using different approaches such as Pedro F. Felzenszwalb et al. [8] proposed a method for image recognition using Deformable Part Models (DPM). In addition further works are devoted using different strategies of using DPM as demonstrated in [9, 10, 11]. Variety of other methods are used for image classification such as SVM [12, 13, 14,15], boosting [16], spatial pyramid matching [17]; however, on the other hand the most dominant recent works achieved using Convolutional Neural Network (CNN). The last is used widely variety of applications such as image recognition [31, 18, 22, 19, 20], object detection [20, 21, 23, 24], scene labeling [25], segmentation [26, 27], and variety of other tasks [28, 29, 30]. All the mentioned above works use Stochastic Gradient Descent (SGD). However, in this work, this algorithm is replaced by PSO. In addition, hybrid training algorithm of both PSO and SGD is used.

### III. PARTICLE SWARM OPTIMIZATION (PSO)

PSO is an evolutionary stochastic optimization computational algorithm introduced by Eberhart and Kennedy [32,33,34]. Particles are randomly initialized, and periodically updated to introduce a new sophisticated population with new fitness. Each particle updates its new position contingent on its history and the best particle history. Thus the particle movement exploits on two values. The first value is the local best, which characterizes the best value so far for the particle itself, and the second value is the global best, which denotes the best value achieved so far by any particle within the swarm. At each time step, particles traverses toward its new best position by altering another parameter termed velocity. The following notions are the formulas used to tune CNN parameters.

$$v_k^{t+1} = w^t v_k^t + c_1 r_1 (lbest_k - x_k^t) + c_2 r_2 (gbest - x_k^t) \quad (1)$$

$$x_k^{t+1} = x_k^t + v_k^{t+1} \quad (2)$$

where,  $v_k^t$  and  $x_k^t$  denote the velocity and position of the particle  $k$  at moment  $t$ , respectively;  $c_1$ , and  $c_2$  are accelerating factors,  $r_1$ , and  $r_2$  are random numbers between  $[0,1]$ ,  $lbest$  is the best position for the particle  $k$ ,  $gbest$  is the best particle in the whole swarm. It is obvious that PSO notions are unpretentious and have very rare parameters to be adjusted.

The PSO has remarkable convergence in the initial stages, but it quickly traps to local optimum. In addition, PSO has difficulty incapacitating to avoid local optimum if the search space encompasses only optimal solution [35]. The PSO predominantly experiences premature convergence and searches in region adjacent to global minimum as training progresses chronologically [36, 37] causing PSO permanently trapped in local optimum region. Therefore PSO is amalgamated by Genetic Algorithm (GA), which is an evolutionary algorithm widely used in solving problems in various fields [38-41]. It defines an initial generation that searches in domain space of the problem and generates a new population based mechanisms of reproduction, crossover, and mutation, which is frequently applied to produce new offspring. Usually, new descendants have higher quality and

better fitness than ancestors. The GA induces enhancing PSO by merging particles in a bright approach to produce new generations. Combining GA and PSO crucially leverages the proposed hybrid training method by sharing information among particles, increasing the diversity of search space, countenancing the training vital through computation steps, and finally averting PSO to occur in local optimum. To sustain a smooth transition for the hybrid training along computation steps, Genetic Algorithm is applied to PSO whenever there are one of the following factors; i) premature convergence, ii) no progress in the fitness function, or iii) Error changes remains steady from two to three consecutive steps.

### IV. PROPOSED OPTIMIZATION ALGORITHM

Since SGD has slow convergence and it cannot be fully parallel to take advantages of GPUs, in this paper, a robust hybrid training algorithm is proposed for CNN training. The algorithm is combined both PSO and SGD, and it is called PSO-SGD, which is a highly parallel method. In this approach, it is expected that the unified PSO and SGD algorithm can crucially achieves superior results and surpass previous methods because of still preserving gains of using SGD and the PSO is recruited as revival constituent. For instance, instead of running one particle, which characterizes the whole CNN parameters, plurality of particles is used and scattered over the scope of search space. Also all particles collaborate with each other using delicate method elucidated in next sections. The proposed training algorithm is divided into dual phase. In the first phase, the CNN parameters are initialized and trained using PSO. Then, when PSO progress induction decelerates, the SGD algorithm is applied for few iterations. After few iterations, the process is switched to PSO and so on. In addition, PSO is consolidated by Genetic Algorithm (GA), which is exploited to stimulate particles and overcomes SGD lethargy. Moreover, unlike standard PSO, which requires a long time to reach the potent area, hybrid PSO provides fast and enhanced optimization [33, 34]. In this algorithm, it is endeavored to preserve the training CNN vital for the whole training period. Algorithm 1 shows the CNN training using the proposed hybrid training method. Algorithm 2 describes PSO alone as well.

---

#### *Algorithm 1. CNN training using the proposed hybrid method*

---

- Initialize *parameters* with different means
- $w^n = [w_1^n, w_2^n, \dots, w_p^n]$  and  $b^n = [b_1^n, b_2^n, \dots, b_p^n]$
- Begin
- Repeat  $t$
- For  $k \leftarrow 1$  to  $P$  do
- Model evaluation  $E_k = \frac{1}{N} \sum_{i=1}^N (\text{Ref} - \text{CNN}(\text{output}))^2$
- End
- $pbest = \min(E(p^{p(t)}), E(p^{p(t-1)}), E(p^{p(t-2)}) \dots E(p^{p(t-n)}))$
- $gbest = \min(pbest^1, pbest^2, pbest^3, \dots, pbest^P)$
- Update (1) and (2)
- If (E) saturates for 3-8 iterations,
- go back to repeat  $t$

- else
- Apply GA by choosing random particles from  $P_1$  to  $P_p$
- End
- Choosing 10% of the best particles
- For  $k \leftarrow 1$  to  $n$  do
- Apply SGD with those particles  $SGD(p_i)$
- End
- End
- Until the condition reached

---

**Algorithm 2. PSO Algorithm**

---

- randomly initialize CNN parameters
- Model Evaluation
- $g_{best} = \arg \min_j E(\theta) ; j = 1..p$
- if  $E(p_{best}^{p(t-1)}) > E(p_{best}^{p(t)}) \Rightarrow E(p_{best}^{p(t-1)}) = E(p_{best}^{p(t)})$
- $V_1^{p(t)} = V_1^{p(t-1)} + cr(p_{best}^p - \chi_1^{p(t-1)}) + cr(g_{best} - \chi_1^{p(t-1)})$
- $\chi_1^{p(t)} = \chi_1^{p(t-1)} + V_1^{p(t-1)}$
- until condition satisfied

V. PROPOSED MODEL ARCHITECTURE

CNN generally consists of alternatives two main layers called convolution and max-pooling layer and end up with fully connected layer. All these layers are connected to each other with weights. However, there are many different other CNN architectures. In this study, the same structure proposed by Yann LeCun et al. [42] is used.

There are is of ambiguous steps, which need to be clarified such as how can the CNN parameters be encapsulated into particles? How do they cooperate with each other? How can it justify the best particles with the ensembles of swarm? To answer these questions, how the parameters of CNN are distributed. It is obvious that the weights and biases are constituent parameters of CNN. Therefore, in this work, the weights and bias are dismantled and encapsulated into vectors as shown below:

$$W = \{w^1, w^2 \dots w^p\} \quad (3)$$

$$P = \{b^1, b^2 \dots b^p\} \quad (4)$$

$$w^n = \{w_1^n, w_2^n, \dots, w_L^n\} \quad (5)$$

$$b^n = \{b_1^n, b_2^n, \dots, b_l^n\} \quad l = 1 \dots L, n = 1 \dots P \quad (6)$$

where  $l$  is the layer index,  $L$  is the total number of layers,  $n$  is the particle  $n$ ,  $P$  is the total number of particles,  $w_l^n$  is the weight parameters of layer  $l$ , and  $b_l^n$  is the bias parameters of the layer  $l$ . Finally the final total parameters of bias and weights are given by

$$W^n = \{W, P\} \quad (7)$$

Fig. 2 shows the first convolution and max-pooling layers of CNN, and there are set of filters and each has dimensions  $n \times n$  and can be vectored to be  $1 \times n^2$ . Thus, having  $m$  filters for  $l$ th layer, then the total weight parameters

are  $w_l^n = mn^2$ . In addition, the total bias parameters for the given  $l$ th layer are  $b_l^n = 1 \times m$ .

Since there are  $P$  particles that will be trained, each one of them could be the best one among the swarm and can give an optimal solution. In order to justify which the best particle among swarm, the following notion is used:  $w^* = \arg \min E(\theta)$

where  $w^*$  is the best particle among swarm and  $E$  described below is the measured error between the reference and the model output.

$$E = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^m (Ref_{ji} - d_{ji})^2 \quad (8)$$

Where  $N$  is the number of training samples,  $m$  is the number the output layers,  $Ref_{ji}$  is the reference,  $d_{ji}$  is the output of CNN. For clarification and showing the difference between the updating parameters using BP and PSO, fig. 2 shows the principle of how BP and PSO work. The figure has circles having functions  $g_n$  where  $n = 1 \dots L$  and  $L$  is the number of layers. The last layer has a function  $f$ .

It is noticeable that BP requires both forward and backward phases. In the forward phase, each activation function gives its response with respect to the input. In backward phase, the derivative is required with to respect to network parameters. PSO does not require any backward phase which can save vast expanse of work and time consumption because the forward phase is less problematical than backward phase.

The reason of why the second phase of network is not compulsory because the PSO algorithm depends on positions and velocities of the particles described in fig. 3. For instance, if there are  $P$  particles and the particle  $m$  is the best particle which satisfies (8), then the particle  $n$  can be updated according to (1) and (2).

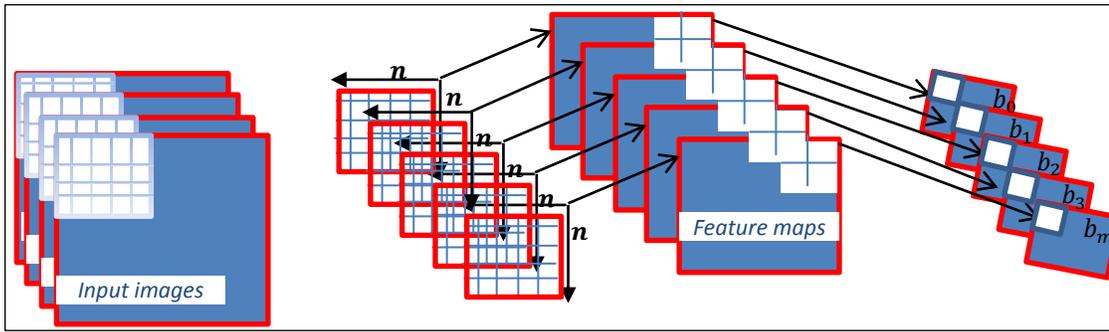


Fig. 1. Convolution and max-pooling layers of CNN network

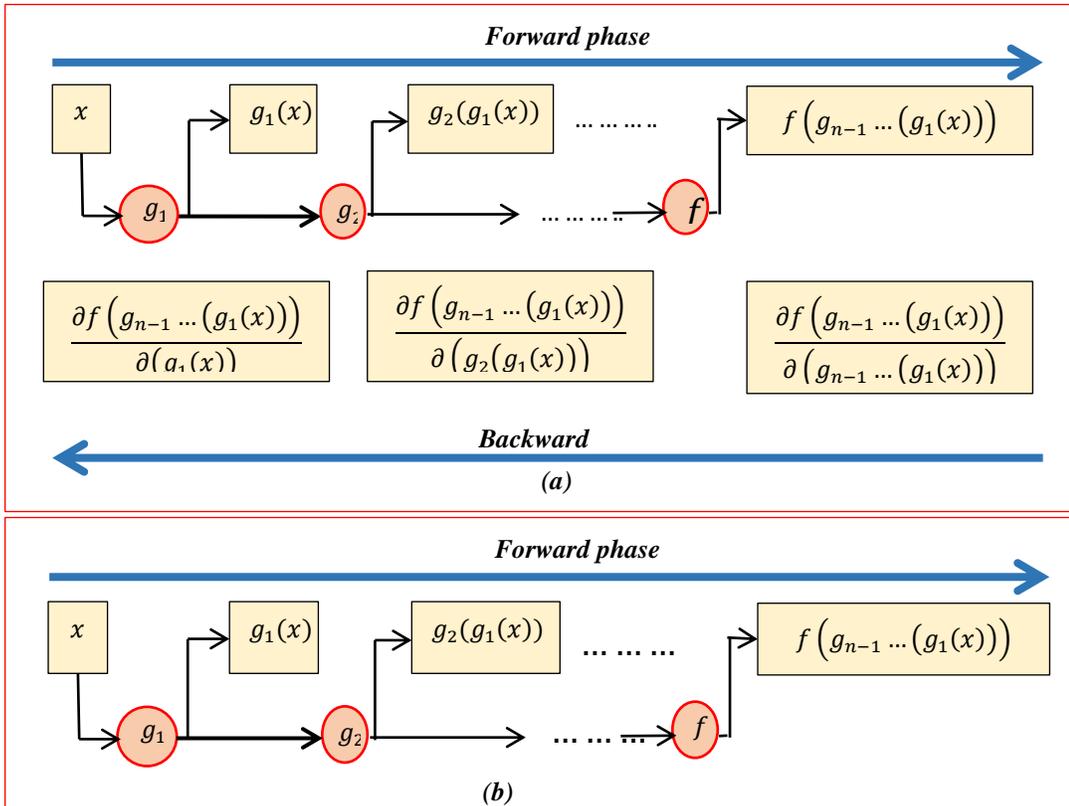


Fig. 2. The principle of working both algorithms (a) BP (b) PSO

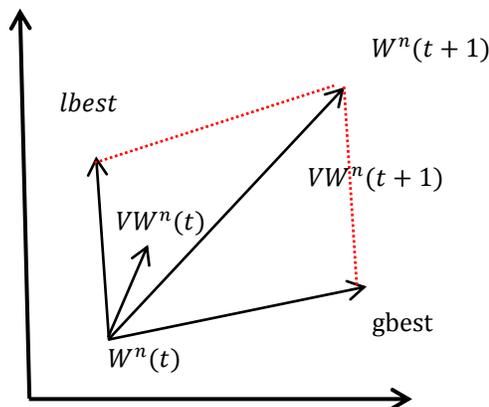


Fig. 3. The principle of parameters updating using PSO

Where  $W^n(t)$  is the parameters mentioned in (7) at time  $t$ ,  $gbest$  and  $lbest$  are mentioned in (1),  $VW^n(t)$  is the velocity of particle  $n$ , and  $W^n(t+1)$  is the next position for the CNN parameters of part  $n$ .

## VI. BENCHMARK EXPERIMENTS

### A. Overview

The algorithm is evaluated on three benchmark datasets: MNIST [18], CIFAR-10 [43], and SVHN [44]. Samples for the datasets are shown in Fig 4. The CNN used in this work consists of alternative convolutional and max pooling layers. Fully connected layer is implemented on the top of the network. The architecture of CNN used for each dataset is dissimilar from each other. The number of particles is 25 and they are randomly initialized with different means and variances.



Fig. 4. Samples of (a) MNIST (b) CIFAR-10 (c) SVHN datasets

### B. MNIST dataset

The MNIST [42] is a hand written digits 0-9. The dataset consists of 60000 samples. 50000 samples are used for training and the rest used for testing. All samples have the same size, which is 28x28 pixels. The pixels are scaled to be in [0, 1] before the training. There is no preprocessing or data augmentation utilized in this work. The CNN structure is 8C-8S-24C-24S-89C-90F-10F, where C stands for Convolution layer, S is for subsampling layer, and F is for full conned layer. In this dataset, the size of mini-batches is 128 images. The proposed hybrid PSO and SGD is exploited for training. At the beginning, the particles are trained using PSO only and Mean Square Error (MSE) mentioned in (3) is used as fitness assessment for the particles. The lowest MSE particle is the highest fitness is. In these experiments, MSE keeps dropping in few iterations and it saturates after that. To circumvent such margins, SGD and GA are launched when there is no further error dropping seen. SGD-GA is usually applied if error saturates between 5-8 iterations. Test accuracy is 0.9957 % for MNIST dataset. To best of the knowledge, this is the best reported result without preprocessing, augmentation, or dropout. A summary of the best published results on MNIST dataset is shown in Table I.

when a large dataset is used such as MNIT, which has 60000 gray images for training and 10000 for testing or CIFAR-10, which has 50000 color images for training and 10000 for testing with a mini-batch 128 sued, it influences PSO performance because it cannot choose the best particle which depends on only 128 images so local minimum occurs.

TABLE I. RESULTS ON MNIST DATASET

Method	Ref. #	Test Accuracy
Unsupervised Learning	[45]	0.64
What is the Best Multi-Stage	[22]	0.53
2-Layer CNN + 2-Layer NN	[46]	0.53
Stochastic Pooling	[46]	0.47
NIN + Dropout	[46]	0.47
Conv. maxout + Dropout	[47]	0.45
<b>Hybrid PSO-SGD</b>	<b>ours</b>	<b>0.43</b>

To tackle this problem, a hybrid training algorithm of PSO and SGD is used. Instead of than using single algorithm, by collaborating two algorithms with each other, a better performance is reached. Table I shows most of the state-of-the-art results on MNIST. A comparison is performed with only results that do not have preprocessing or they have the same architecture of CNN. It is clear that this work surpasses other works that do not use distortions or any preprocessing.

### C. CIFAR-10 Dataset

The CIFAR-10 dataset consists of 10 classes of natural 32x32 RGB images with 50,000 for training and 10,000 for testing [19]. The CNN used for this dataset is described as: 12C-12S-48C-48S-89C-90F-10F, which is denoted to convolutional layer with 12 feature maps, subsampling layer, and a convolutional layer with 48 feature maps, subsampling layer, and a convolutional layer with 89 feature maps, and a fully connected output layer with 10 outputs.

The subsampling layers have filters over non-overlapping region of size 2x2. The same steps are followed as in MNIST for training CNN. However, in this dataset, occurring in local optimum is faster than previous datasets so the number of times applying SGD is higher. It is determined that PSO-GA needs to be united by SGD as complicated dataset used such as CIFAR-10 because the MNIST dataset is easier for classification than CIFAR-10. Nevertheless, the benefit of using hybrid POS-SGD is still obtainable. The test accuracy gotten on this dataset is 82.41%.

From table II, it is evident that the proposed method surpasses the other state-of-the-art works. It is worth mentioning that only comparison with methods that use the same structure of CNN is considered. Any other techniques that can be very valuable for increasing accuracy such as dropout or drop-connect are not used. In this work, the same general structure proposed by Yann LeCun et al. [42] is used and only the training algorithm is replaced but the same configuration of CNN is kept.

TABLE II. TEST SET ACCURACY RATES ON CIFAR-10 DATASET

Method	Reference #	Accuracy
Tiled CNN	[48]	73.10
Improved LCC	[49]	74.50
KDES-A	[50]	76.00
PCANet-2 (combined)	[51]	78.67
PCANet-2	[51]	77.14
K-means (Triangle, 4000 features)	[52]	79.60
Cuda-convnet2	[53]	82.00
<b>Hybrid PSO-SGD</b>	<b>[ours]</b>	<b>82.41</b>

#### D. SVHN Dataset

The last experiment is assessed on the street view house numbers (SVHN). The dataset consists of 604,388 samples (training and extra set) and 26,032 samples as test images. In addition, each the dataset is color images and the size of each sample is 23x32 pixels. Following [1, 2], 400 samples per class from the training set and 200 images per class from extra set are selected to implement validation set. The task in this dataset is to classify the digit in the center of each image. Preprocessing local contrast normalization is used following Goodfellow et al. [6]. In addition, the same CNN assembly and parameters setting are used as CIFAR-10. The test error obtained is 2.48%. The result is shown in Table III.

TABLE III. TEST ERROR RATES ON SVHN DATASET

Method	Reference #	Test Error %
Multi-Stage Conv. Net + 2-layer Classifier	[44]	5.03
Multi-Stage Conv. Net + 2-layer Classifier + padding	[44]	4.90
Maxout Networks	[16]	2.47
<b>Hybrid PSO-SGD</b>	<b>[ours]</b>	<b>2.48</b>

Again Maxout Networks is used in very large CNN implementations because it is implemented over Krizhevsky et al. [31] code. However, a conventional CNN is used instead. In addition, a leveraging PSO algorithm is used in this work which is faster than SGD

#### VII. CONCLUSION

In this work, a new hybrid training process is proposed and demonstrated called Particle Swam Optimization- Stochastic Gradient Decent (PSO-SGD) algorithm, for training Convolution Neural Network (CNN). It is established that the algorithm is well suited for achieving nontrivial results on different datasets and surprisingly achieving state-of-the-art on these datasets. The proposed algorithm is a proficient method for training because it combines both PSO and SGD in an innovative fashion. Analysis also shows that the proposed method is superior on three different benchmark datasets. The hybrid training method avoids occurring in local optimum and premature saturation inspired by using single algorithm. Additionally, it preserves the training vital for the whole training period and restrains the lethargy inherited by a monocular algorithm.

#### VIII. FUTURE WORK

In future more influential parameters will be explored. There are more parameters that can influent model accuracy will be investigated in the future work. Deeper analysis and more challenge datasets such as ImageNet also will be as a

part of the future work. Also reporting time consumption and how fast execution time for training and testing will be consider endeavoring to reach real time execution.

#### REFERENCES

- [1] S. L. Phung and A. Bouzerdoum, "A pyramidal neural network for visual pattern recognition," IEEE Transactions on Neural Networks, vol. 27, no. 1, pp. 329343, 2007
- [2] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf , "DeepFace: Closing the Gap to Human-Level
- [3] Min Lin, Qiang Chen, and Shuicheng Yan "Network In Network" arXiv 1312.4400v3,4 Mar 2014
- [4] Dan Cires,an, Ueli Meier and Jurgen Schmidhuber, "Multi-column Deep Neural Networks for Image Classification" CVPR 2012
- [5] Julien Mairal, Piotr Koniusz, Zaid Harchaoui, and Cordelia Schmid, "Convolutional Kernel Networks" arXiv 14 Nov 2014 .
- [6] Ian J. Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, Yoshua Bengio "Maxout Networks" ICML 2013
- [7] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov, "Scalable Object Detection using Deep Neural Networks" CVPR, 2014.
- [8] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan , " Object Detection with Discriminatively Trained Part Based Models"
- [9] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in IEEE Conference on Computer Vision and Pattern Recognition, 2003
- [10] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 2013. 1, 2.
- [11] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," International Journal of Computer Vision, vol. 77, no. 1, pp. 259–289, 2008.
- [12] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In CVPR, 2006. 1, 2, 5, 6, 7
- [13] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, CalTech, 2007.
- [14] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. arXiv preprint arXiv:1403.6382, 2014. 1, 2.
- [15] Y. Lin, T. Liu, and C. Fuh. Local ensemble kernel learning for object category recognition. In CVPR, 2007
- [16] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In ECCV, 2004.
- [17] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In ICCV, 2005.
- [18] Fabien Lauer, Ching Y. Suen, and Gerard Bloch "A trainable feature extractor for handwritten digit recognition" Journal Pattern Recognition 2007.
- [19] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, Zhuowen Tu, "Deeply-Supervised Nets "NIPS 2014
- [20] M. Fischler and R. Elschlager, "The representation and matching of pictorial structures," IEEE Transactions on Computer, vol. 22, no. 1, 1973.
- [21] Kaiming, He and Xiangyu, Zhang and Shaoqing, Ren and Jian Sun "Spatial pyramid pooling in deep convolutional networks for visual recognition" European Conference on Computer Vision, 2014
- [22] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato and Yann LeCun " What is the Best Multi-Stage Architecture for Object Recognition?" ICCV'09, IEEE, 2009.
- [23] Ross Girshick, "Fast R-CNN "arXiv preprint arXiv:1504.08083, 2015
- [24] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In ICCV, 2013. 8

- [25] Karen Simonyan and Andrew Zisserman "VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION" arXiv:1409.1556v5 [cs.CV] 23 Dec 2014.
- [26] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. International Conference on Learning Representation, 2013. 2.
- [27] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR, abs/1311.2524, 2013. 4
- [28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei. Imagenet: A large-scale hierarchical image database. In CVPR, 2009. 2.
- [29] L. N. Clement Farabet, Camille Couprie and Y. LeCun. Learning hierarchical features for scene labeling. PAMI, 35(8), 2013. 1, 2
- [30] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations".
- [31] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey. ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25 (NIPS'2012). 2012.
- [32] Frans van den Bergh and Andries P. Engelbrecht "A Cooperative Approach to Particle Swarm Optimization" IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, JUNE 2004, VOL. 8, NO. 3.
- [33] Shigenori Naka et al. "A Hybrid Particle Swarm Optimization for Distribution State Estimation", IEEE TRANSACTIONS ON POWER SYSTEMS, FEBRUARY 2003, VOL. 18, NO. 1.
- [34] X. Wang" A Hybrid Particle Swarm Optimization Method", IEEE International Conference on Systems, Man, and Cybernetics, October, pp. 4151-4157, 2006.
- [35] Jiang Du et al. , "Improved PSO Algorithm and Its Application in Optimal Design for Rectifier Transformer", IEEE International conference on Intelligent and Integrated System, pp. 605-608, 2010,
- [36] Yi-Xiong Jin et al. "Local Optimum Embranchment Based Convergence Guarantee Particle Swarm Optimization and Its Application in Transmission Network Planning", IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific Dalian, China, pp. 1-6, 2005.
- [37] Shigenori Naka et al. "A Hybrid Particle Swarm Optimization for Distribution State Estimation", IEEE TRANSACTIONS ON POWER SYSTEMS, FEBRUARY 2003, VOL. 18, NO. 1.
- [38] Zongmei Zhang, Zhifang Sun, and Hiroki TAMURA, "Local Linear Wavelet Neural Network with Weight Perturbation Technique for Time Series Prediction", IEEE international Conference on Computer Science and Software Engineering, pp. 789-801, 2008.
- [39] Zhaohu Huang et al, "A Particle Swarm Optimization Algorithm for Hybrid Wireless Sensor Networks Coverage", IEEE Symposium on Electrical & Electronics Engineering (EEESYM), pp. 630-632, 2012.
- [40] Esmail Hadavandi, Arash Ghanbari, and Salman Abbasian-Naghneh, "Developing a Time Series Model Based On Particle Swarm Optimization for Gold Price Forecasting", IEEE Third International Conference on Business Intelligence and Financial Engineering, pp. 337-340, 2010.
- [41] Mohammad Yunus Ali and Kaamran Raahemifar, "Reactive Power Optimization Based on Hybrid Particle Swarm Optimization Algorithm", IEEE Canadian Conference on Electrical and Computer Engineering, pp. 1-5, 2012.
- [42] Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner "Gradient-Based Learning Applied to Document Recognition" Proc of the IEEE, November 1998.
- [43] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Computer Science Department, University of Toronto, 2009. 1, 6.
- [44] P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In ICPR, 2012
- [45] Marc'Aurelio Ranzato, Fu-Jie Huang, Y-Lan Boureau, Yann LeCun, "Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition" CVPR, 2007
- [46] Matthew D Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. arXiv preprint arXiv:1301.3557, 2013.
- [47] Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. arXiv preprint arXiv:1302.4389, 2013.
- [48] Q. V. Le, J. Ngiam, Z. Chen, D. Chia, P. W. Koh, and A. Y. Ng, "Tiled convolutional neural networks," in NIPS, 2010.
- [49] K. Yu and T. Zhang, "Improved local coordinate coding using local tangents," in ICML, 2010.
- [50] L. Bo, X. Ren, and D. Fox, "Kernel descriptors for visual recognition," in NIPS, 2010
- [51] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi Ma "PCANet: A Simple Deep Learning Baseline for Image Classification?" "arXiv:1404.3606v2 [cs.CV] 28 Aug 2014.
- [52] Adam Coates, Honglak Lee, and Andrew Y. Ng "An Analysis of Single-Layer Networks in Unsupervised Feature Learning" In AISTATS 14, 2011.
- [53] A. Krizhevsky, "cuda-convnet," <http://code.google.com/p/cuda-convnet/>, July 18, 2014.