International Journal of Computational Intelligence Systems
Vol. **14(1)**, 2021, *pp.* 693–700
DOI: https://doi.org/10.2991/ijcis.d.201216.003; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

MADL: A Multilevel Architecture of Deep Learning

Samir Brahim Belhaouari[1,*], Hafsa Raissouli[2]

[1]*College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar*
[2]*College of Computer Science and Information Technology, University Putra Malaysia, Serdang, Malaysia*

ARTICLE INFO

Article History
Received 02 May 2020
Accepted 26 Sep 2020

Keywords
Convolutional neural network
Multilevel architecture of deep learning
Advanced activation function
CIFAR-10, MADL

ABSTRACT

Deep neural networks (DNN) are a powerful tool that is used in many real-life applications. Solving complicated real-life problems requires deeper and larger networks, and hence, a larger number of parameters to optimize. This paper proposes a multilevel architecture of deep learning (MADL) that breaks down the optimization to different levels and steps where networks are trained and optimized separately. Two approaches of passing the features from level i to level $i+1$ are discussed. The first approach uses the output layer of level i as input to level $i+1$ and the second approach discusses introducing an additional fully connected layer to pass the features from it directly to the next level. The experimentations showed that the second approach, that is the use of the features in the additional fully connected layer, gives a higher improvement. The paper also discusses an advanced customizable activation function that is comparable in its performance to rectified linear unit (ReLU). MADL is experimented using CIFAR-10 and exhibited an improvement of 0.84% compared to a single network resulting in an accuracy of 98.04%.

1. INTRODUCTION

The use of deep neural network (DNN) has been a growing trend over the years in solving many real-life problems [1]. DNN consists of several layers of neurons that are fully connected to each other. The layers that are located between the input layer and the output layer are called hidden layers. The value of each neuron in the hidden layers is computed as the weighted sum of the previous layers' neurons added to a bias factor and activated using an activation function. The fact that DNN are fully connected makes them heavy and computationally expensive. As stated in Ref. [2], this full connectivity can be wasteful. Convolutional neural networks (CNNs) on the other hand, are not fully connected. CNN uses convolutional layers to pass a filter of size $n \times m$ on the image and perform a convolution operation to extract features from the image. An activation layer is usually placed after the convolutional layer. The activation brings non-linear properties to the network. The activation functions used in the past (Sigmoid and Tanh) suffered from weaknesses such as the vanishing gradient problem [3]. This caused the development of more powerful functions such as rectified linear unit (ReLU), leaky ReLU, and parameterized ReLU. Of these functions, ReLU remains the most used one. ReLU is simply defined as $max(0, x)$ which means that the neuron value is mapped to zero if it is negative and to itself if it is positive. Although this function has a strong gradient, it has a dead activation in negative neurons, which makes the evolution of new activation functions still open to discussion. Following the activation layer, a pooling layer is usually placed to down sample the filtered values to a subset by taking either the max or the average.

CNN has witnessed a tremendous popularity in the last decade making machines and humans rivals [4]. CNN has contributed in many domains and helped solving real-life problems such as object detection [5,6], image classification [7,8] face recognition [9], image denoising [10], and pose estimation [11]. A lot of interest was given to ways to improve the prediction accuracy of the CNN model and the rummage for simple and efficient techniques is still persistent [12–14]. CNN is powerful in extracting high-level features from images [15]. Traditional feature extraction techniques generally consider limited types of features [16,17]. CNN on the other hand, has the ability to learn invariant features [18]. To make use of the feature extraction ability of CNN, many studies have used CNN as a feature extractor [15]. Ensembling methods are also popular techniques where one or several CNNs are used to extract features from the image and then perform class decision using another classifier such as support vector machine (SVM) [19,20]. Some studies have used several CNNs to classify a set of given images and have performed voting to assign the final decision class [21]. Other studies have combined different CNN architectures (e.g. AlexNet and LeNet) in one architecture for richer domain specific feature learning [22].

This paper proposes an ensembling approach of CNN architectures. The approach of this model is to break down the optimization of deeper networks to levels where each network in a given level is trained and optimized separately leading to a better performance. Advanced customizable activation function that is comparable in its performance to ReLU is also proposed. The rest of the paper is

* *Corresponding author. Email: sbelhaouari@hbku.edu.qa*

organized as follows: Section 2 presents a literature review, the proposed approach is in Section 3, Section 4 discussed the experiments and results, and the conclusion is in Section 5.

2. LITERATURE REVIEW

2.1. Ensembling Techniques

The use of CNN for feature extraction and ensembling techniques have been explored in many studies [19–21]. The motivations to seek techniques to assemble different models have been discussed amply [19,23]. The main reasons manifest in the limitations of building one perfect model due to the possibilities for the training phase to land on a local instead of a global minimum [23,24]. In addition, the fact that CNN performs well as a feature extractor supports the concept of assembling several CNNs for more various features [2,21]. In Ref. [19], the authors investigated the use of CNN for feature extraction for face recognition. The method used suggests extracting the features using a CNN of three convolutional layers. Feature selection on the extracted features is then performed using PCA. For the final classification, SVM is employed. The experiments showed that SVM performs better with the reduced features. The authors suggest that the PCA helps describing the data vectors in a better way. Similar to the latter study, in Ref. [20], the authors tackled face detection using two different CNNs for feature extraction (Clarifai and VGG), PCA for feature selection, and SVM for classification. In Ref. [25], the proposed method is a fusion of multiple models that are trained with different feature sets that are extracted using different machine learning algorithms. Then as in Refs. [19,20] feature selection is done using PCA. Other studies have combined popular architectures such as AlexNet and LeNet in one architecture for better feature extraction [22]. A study in Ref. [2] combined AlexNet and LeNet to form three different architectures namely, RFTPM, FTPM, and HTPM. The three proposed architectures are shown to be effective in unconventional image filtering. Out of the three, the HTPM, that is a half trainable model where the first four convolutional layers are initialized by AlexNet, is shown to be the superior and the most scalable compared to the baseline models. Another study in Ref. [26] uses five convolutional layers to extract obstacle features. The features are passed to a network that detects the obstacle area. This developed architecture shows promising results in intricate driving environments. An other study in Ref. [21] introduced an ensembling technique of several CNNs that are optimized and fine-tuned multiple times to form twelve separate models. Two architectures were used in the generation of the models, namely, Resnet and Densenet. The fusion of the models is done by taking the output of the CNNs, that is a probability value, and voting for the correct label.

From the literature, the ensembling techniques use roughly one of the following approaches:

- Using convolutional layers for feature extraction which leads to a large number of features impelling to perform feature selection before passing the features to another classifier.
- Combining different architectures into one architecture forming a single network with a set of convolutional and pooling layers for feature extraction followed by fully connected layers for decision-making.
- Taking the output of n networks as a class probability and making the decision by voting for the correct class.

This paper addresses a generalized ensembling model, namely, multilevel architecture of deep learning (MADL). MADL performs the feature extraction using n networks. Two fusion techniques of the n networks are discussed. The proposed model is experimented on CIFAR-10.

2.2. State-of-the-Art on CIFAR-10

Many studies aimed to improve the performance of CNN through various techniques especially for image classification on the benchmark datasets such as CIFAR-10. Ref. [27] presented Mixup distribution that makes the model act linearly between training examples in order to reduce the oscillations that occur when predicting the labels of the test examples. CIFAR-10 was trained using Mixup and resulted in an accuracy of 97.3%. Ref. [28] proposed the Manifold Mixup regularizer that improves the model generalization. The conducted experiments on CIFAR-10 achieved an accuracy of 97.46%. Ref. [29] argued that standard CNNs are overparameterized and presented a scheme for parameter sharing that results in a hybrid network between CNN and RNN. The experimentations on CIFAR-10 recorded an accuracy of 97.47%. In Ref. [30], the authors proposed Fixup initialization technique that promotes a network training without batch normalization. Fixup initialization was used with mixup and cutout on CIFAR-10 to achieve an accuracy of 97.7%. Ref. [31] proposed the squeeze and excitation (SE) blocks that improve the modeling of the channel-wise features. CIFAR-10 was trained using SE blocks on shake-shake network with cutout to achieve an accuracy of 97.88%. In Ref. [32] the study performed a reduction of the search cost of NAS and employed it using a proxyless strategy, namely, ProxylessNAS. CIFAR-10 was trained with ProxylessNAS gradient using PyramidNet backbone and cutout and resulted in an accuracy of 97.92%. Ref. [33] proposed a semi-supervised learning auto encoding approach the experiments were conducted on 250 labels of CIFAR-10 and resulted in an accuracy of 98.01%. Ref. [34] got an accuracy of 98.3% with 4000 labels of CIFAR-10 that was used to experiment the proposed auto-augment technique that searches for effective augmentation policies. Table 1 describes briefly the state-of-the-art on CIFAR-10.

3. THE PROPOSED APPROACH

3.1. Multilevel architecture of deep learning

Improving the accuracy of deep networks has been a heed to researchers [35–38]. According to the previous work, the concept of grouping several CNNs appeared in many studies [20,21]. In this paper, MADL is proposed as an accuracy improvement technique that makes use of the features extracted by several networks. This architecture builds several networks in levels as illustrated in Figure 1. As Figure 1 shows, there is i levels where each set of networks in a given level are connected to one network in the next level.

The goal of this concept is to break down the parameter optimization of the networks to multi levels and steps, DL_{ij}, where each block is trained and optimized separately in order to improve performance. To pass the features from level $i-1$ to level i, two approaches are discussed:

Table 1 Summary of the state-of-the art on CIFAR-10.

Ref. No.	Network Type	Improvement Technique	Epochs	Augmen-tation	Trans-fer Learning	Acc (%)
[27]	WideResNet and DenseNet-190	Mixup	200	√	×	97.3
[28]	PreAct ResNet34	Manifold Mixup	1200	√	×	97.46
[29]	SWRN 28-14-2	parameter sharing	200	√	×	97.47
[30]	WideResNet40-10	Fixup	Not stated	√	×	97.7
[31]	Shake-Shake 26 2x96d	Squeeze and excitation	Not stated	√	×	97.88
[32]	Proxyless-G	ProxylessNAS	600	√	×	97.92
[34]	BiT-M (4000 labels of CIFAR-10)	Auto-augment	90	×	√	98.4

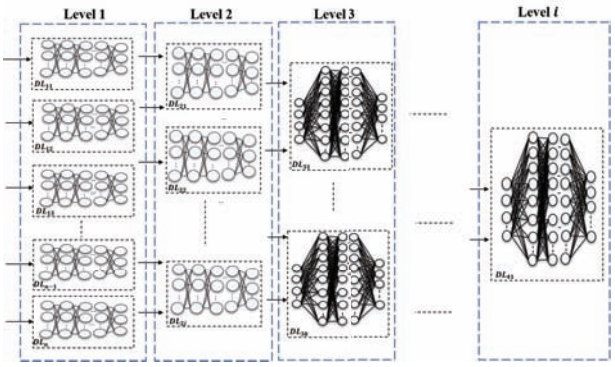

Figure 1 Multilevel architechture of deep learning (MADL) concept.

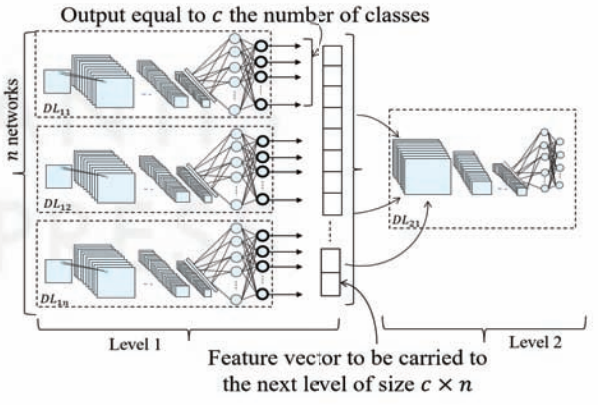

Figure 2 Multilevel architecture of deep learning (MADL) Approach one: A two-level architecture where the input of the second level is taken from the output layer.

- MADL Approach one: taking the values of the output layer in level $i-1$ (that are class probability values), combining them, and passing them to level i as shown in Figure 2
- MADL Approach two: Introducing an additional fully connected layer with x neurons, and passing the features from the additional fully connected layer to level i as shown in Figure 3

3.2. An Advanced Activation Function

The networks across the levels can differ in their architecture and can use different activation functions to help extracting different

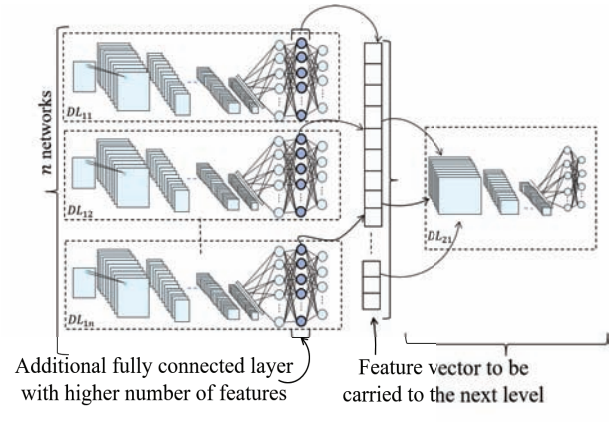

Figure 3 Multilevel architecture of deep learning (MADL) Approach two: A two-level architecture where the input of the second level is taken from the additional fully connected layer.

features. This raises the inducement of finding additional effective activation functions other than ReLU. We propose here a customizable new activation function. Viewing neural networks as a structure that is inspired from the human nervous system, the signal that stimulates the neurons coming from neighboring neurons is an electrochemical stimulus that can be measured, and its strength defines if a neurons will be activated or inhibited [39]. The strength of this signal is usually small [40]. This fact is a motivation to find an activation function that has a small range, yet, overcomes the problem of vanishing gradient that other functions suffered from [3]. Thus, the function in Equation (1) evolved.

$$f(x) = \begin{cases} 0, & x \leqslant a \\ \left(\frac{x-a}{b-a}\right)^n, & a < x < b \\ 1, & x \geqslant b \end{cases} \tag{1}$$

where a, b, and n are parameters to be set. The derivative of this function is expressible as

$$f'(x) = \begin{cases} 0, & x \leqslant a \\ n\left(\frac{x-a}{b-a}\right)^{n-1}, & a < x < b \\ 0, & x \geqslant b \end{cases} \tag{2}$$

When using CNN for feature extraction, the proposed activation function, when set to different parameters, can serve in extracting a variety of features. So as to put this function under experiment, known architectures like Googlenet, Resnet, and Densenet

were used after modifying the ReLU activation layer to the proposed function. The obtained results led to some modifications in the proposed function, conducting to the function in Equation (3)

$$f(x) = \begin{cases} 0, & cx \leqslant a \\ b\left(\frac{cx-a}{b-a}\right)^n, & a < cx < b \\ cx, & cx \geqslant b \end{cases} \quad (3)$$

where the derivative is expressed in Equation (4).

$$f'(x) = \begin{cases} 0, & cx \leqslant a \\ bn\left(\frac{cx-a}{b-a}\right)^{n-1}, & a < cx < b \\ c, & cx \geqslant b \end{cases} \quad (4)$$

The function in Equation (3) still could not attain the starting goal of smaller activation values similar to the human nervous system, and hence, further research can be conducted in this point.

4. RESULTS AND DISCUSSION

4.1. Results of the Advanced Activation Function

So as to scrutinize the use of the proposed activation function and its impact on the performance, the function in Equation (3) was set to different parameters (see Table 2). Note that the proposed function can be viewed as a generalized ReLU as setting $b = 0$ converts the function to ReLU. The experiments used a dataset of 5856 chest X-ray images that has two classes (normal and pneumonia) [41–43]. The network used is a CNN with 6 convolutional layers with batch normalization and maxpooling. Using ADAM optimizer, the initial learning rate is set to 0.01 and the number of epochs is 10. A ten-fold cross-validation is used. Table 2 shows the obtained results.

From Table 2, ReLU and F1 show the highest performance with an accuracy of 95.34% and 95.25% respectively. These two functions also have the lowest variance across the ten-fold cross validations' accuracies. F5 and F6, that have n as a rational number have the lowest average accuracy and the highest variance.

4.2. Results of MADL

The experiments of MADL are carried out on CIFAR-10 dataset [44,45]. This dataset consists of 60,000 colored 32x32 pixel images. The training set is 50,000 images and the test set is 10,000 images. The baseline model comprises an Inception-v3 and a Resnet-50 trained on CIFAR-10. MADL is then built using these two networks. MADL was experimented using approach 1 illustrated in Figure 2 and approach 2 illustrated in Figure 3.

4.2.1. Baseline model results

As a baseline model, Inception-v3 [46], that has a depth of 48 layers and a width of three, and Resnet50 that is 50 layers deep were used. Note that these two architectures (Inception-v3 and Resnet50) have nearly 3 times fewer parameters than Alexnet that is only 8 layers deep. For the training details, Inception-v3 and Resnet-50 were trained with ReLU as an activation function and using SGD for 25 epochs. Transfer learning from pretrained network on ImageNet dataset is performed by conserving the weights of the three first layers to speed up the convergence of the model. The batch size is limited to 10 and the initial learning rate is set to $3e^{-4}$. As Table 3 shows, Inception-v3 and Resnet-50 have resulted in an accuracy of 97.2% and 96.8% respectively. Figures 5 and 6 illustrate the confusion matrices where the classes from left to right are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. With a vertical read of the confusion matrices, we notice that the frog class ranks the highest accuracy of 99.2% and the dog class the lowest that is 93.8% using inception-v3 whereas the deer and cat classes ranks highest and lowest respectively with 99.1% and 92.9% accuracies using Resnet-50.

4.2.2. Proposed model results

The inception-v3 and Resnet-50 in the baseline model are used to build MADL approach 1 by taking the 10 features of the output layer without Softmax. The 10 features from each network are combined to a 20 features vector for each input image that is fed as input to a simple network that contains 5 convolutional layers and one fully connected layer that is trained for 1 epoch. Combining these two networks with 97.2% and 96.8% accuracies resulted in 97.3% that is an improvement of 0.1% over the highest accuracy in the first level. Figure 7 shows the confusion matrix of MADL. With a horizontal read, comparing the MADL results with Inception-v3 and Resnet-50, we notice that only one class (truck class) has improved over the two previous networks.

For MADL approach 2, the first level consists of the inception-v3 and the Resnet-50 with an additional fully connected layer. The same training details of the baseline model are conserved. The second level is a simple CNN with 5 convolutional layers and one fully

Table 2 Results of rectified linear unit (ReLU) and the proposed activation function using different parameters.

	Parameters					
Function	***a***	***b***	***c***	***n***	**Avg. Acc. (%)**	**Variance**
ReLU	–	–	–	–	95.34	1.45
F1	−10	0.01	1	3	95.25	1.68
F2	−10	0.1	1	3	94.15	1.88
F3	−3	0.1	1	2	93.98	2.16
F4	−2	0.1	1	2	93.93	2.64
F5	−2	0.1	1	$\frac{1}{2}$	93.08	3.11
F6	−10	0.1	1	$\frac{1}{3}$	93.16	4.16

Table 3 Summary of results of Inception-v3, Resnet-50, MADL approach 1, and MADL approach 2.

Network	**Level 1**	**Level 2**	**Accuracy (%)**
Inception-v3	–	–	97.2
Resnet-50	–	–	96.8
MADL approach 1	Inception-v3 + Resnet-50	5layers CNN	97.3
MADL approach 2	Inception-v3 + Resnet-50	5layers CNN	**98.04**

connected layer that is trained using SGD for 1 epoch and an initial learning rate of 0.01. Figure 4 illustrates the used architecture.

The figure shows that each input image is trained using the two networks in the first level. Next, the two feature vectors of the image are taken from the additional fully connected layer and concatenated to form the input to the second level with 310 features. The proposed architecture has shown an accuracy improvement from 97.2% using Inception-v3 and 96.8% using Resnet-50 to 98.04% that is a significant improvement. Figure 8 presents the confusion matrix that shows an improvement in most classes compared to the confusion matrices of the first level (see Figures 5 and 6). Table 3 summarizes the obtained results.

As the results suggest, MADL approach 2 outperforms MADL approach 1. Using MADL approach 1, the output of the first level that is 10 neurons, in the case of CIFAR-10 with 10 classes, is combined and passed to the second level. This approach dramatically reduces the number of features from level 1 to level 2. The high number of the useful features in level 1 is summarized in a vector of c values where c is the number of classes, making the next level have way fewer features. This way, we get small use of the trained networks in level 1, and level two shows a small improvement. Using MADL approach 2 overcomes this limitation by preserving a higher number of features from level 1 and hence giving a better improvement. From now on, we consider MADL approach 2 as the proposed model and we call it MADL.

With a view to further experiment MADL on different datasets, skin lesions ISIC dataset and three classes of food-101 dataset were used. The same structure represented in Figure 4 was used with the same network parameters indicated for CIFAR-10 except the number of epochs of the first level that is set to 10 epochs only. Table 4 presents the obtained results. For the skin lesions dataset, the accuracy obtained with Resnet is 77.2% and with Inception-v3 77.3%. The features extracted from the additional fully connected layer are concatenated and used to train the CNN in level two leading to an accuracy of 80.8% that is an improvement of 3.5%. For Food-101 dataset, Resnet and Inception-v3 gave an accuracy of 96.1% and 96.5%, respectively. The combination of the two networks gave an accuracy of 97.1%

As the results in Table 4 manifest, for all the datasets experimented, we notice a significant improvement from level 1 to level 2. The improvement using MADL for CIFAR-10 is 1.24% compared to Resnet-50 and 0.84% compared to inception-v3. For skin lesion dataset, the improvement is 3.5%. For Food-101 the improvement is 0.6%.

5. CONCLUSION AND FUTURE WORK

This paper presents a DNN ensembling approach, namely, MADL. MADL is based on building several networks in levels where each network is trained and optimized separately. This approach breaks down the optimization of the networks to levels instead of optimizing one larger network. The proposed ensembling method also

Table 4 Results of the datasets experimeted with MADL.

Dataset	Level 1 Acc. (%)	Level 2 Acc. (%)	Improvement
CIFAR-10	96.8 97.2	98.04	0.84
Skin Lesion	77.2 77.3	80.8	3.5
Food-101 (3 classes)	96.1 96.5	97.1	0.6

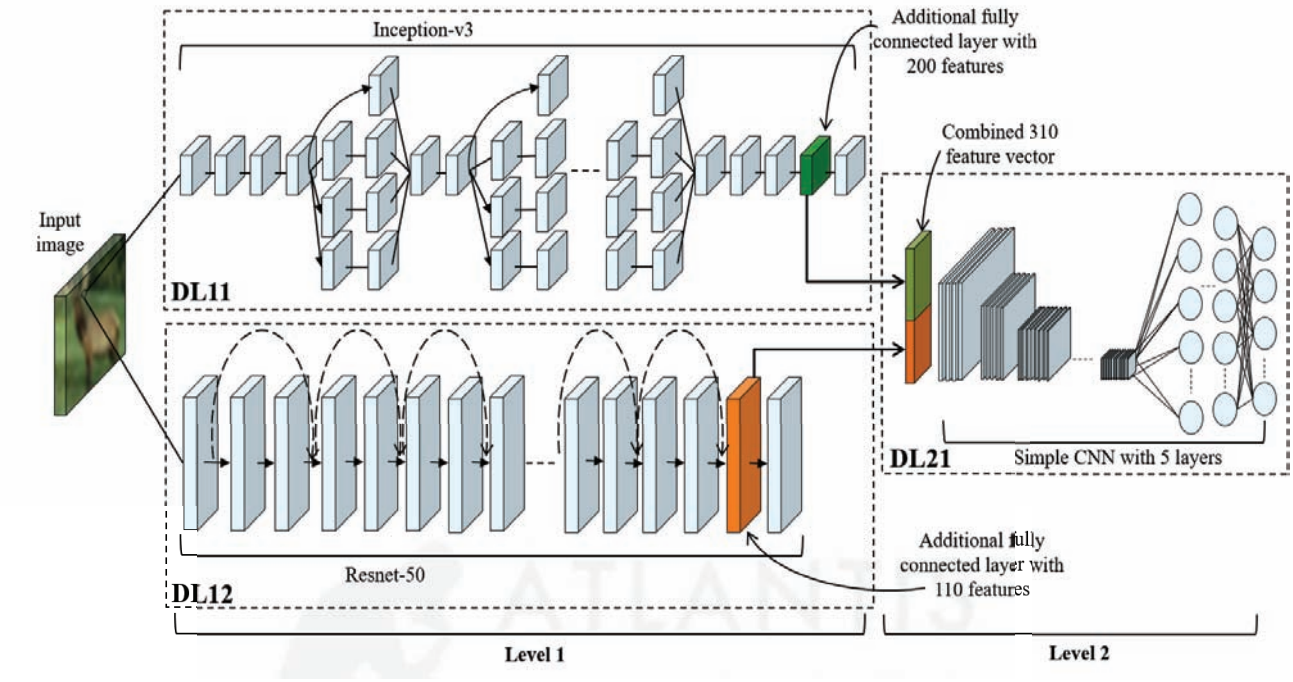

Figure 4 Improved two-level architecture with additional fully connected layer. Inception-v3 as DL11, Resnet-50 as DL12 connected to simple convolutional neural network (CNN) as DL21.

985	1	4	4	0	1	2	2	19	5	96.3 %
2	**989**	1	1	0	0	0	0	5	24	96.8 %
4	1	**967**	6	3	1	4	4	0	0	97.7 %
0	0	8	**940**	3	40	2	1	0	0	94.6 %
0	0	10	13	**989**	13	0	13	0	0	95.3 %
0	0	3	22	1	**938**	0	1	0	0	97.2 %
0	1	5	9	2	3	**992**	1	2	0	97.7 %
0	0	1	2	2	4	0	**978**	0	0	99.1 %
8	0	1	1	0	0	0	0	**972**	3	98.7 %
1	8	0	2	0	0	0	0	2	**968**	98.7 %
98.5 %	98.9 %	96.7 %	94%	98.9 %	93.8 %	99.2 %	97. 8%	97.2 %	96.8 %	**97.2 %**

Figure 5 | Confusion matrix of the Inception-v3 using CIFAR-10 dataset.

958	1	0	1	0	2	1	1	7	1	98.6 %
0	**985**	0	0	0	0	0	0	3	28	96.9 %
9	0	**968**	11	1	6	7	1	2	0	96.3 %
1	0	8	**929**	2	41	3	2	1	1	94%
0	0	9	10	**991**	5	4	9	0	0	96.4 %
0	0	3	35	2	**934**	2	2	0	0	95.5 %
0	0	10	7	2	1	**983**	0	0	0	98%
0	0	0	4	2	11	0	**985**	0	0	98.3 %
28	0	2	1	0	0	0	0	**984**	4	96.6 %
4	14	0	2	0	0	0	0	3	**966**	97.7 %
95.8 %	98.5 %	96.8 %	92.9 %	99.1 %	93.4 %	98.3 %	98.5 %	98.5 %	96. 6%	**96.8 %**

Figure 6 | Confusion matrix of the Resnet-50 using CIFAR-10 dataset.

980	1	4	4	0	2	2	2	9	4	97.2%
1	**987**	1	1	0	0	0	0	4	21	97.2%
4	0	**963**	3	1	1	4	3	0	0	98.4%
0	0	9	**922**	4	19	2	2	0	0	96.2%
0	0	10	9	**989**	11	0	12	0	0	95.9%
0	0	6	47	1	**961**	0	1	0	0	94.6%
0	1	5	9	2	2	**992**	2	2	0	97.7%
0	0	1	2	3	4	0	**978**	0	0	99.0%
12	0	1	1	0	0	0	0	**983**	4	98.2%
3	11	0	2	0	0	0	0	2	**971**	98.2%
98%	98.7 %	96.3 %	92.2 %	98.9 %	96.1 %	99.2 %	97.8 %	98.3 %	97.1 %	**97.3 %**

Figure 7 | Confusion matrix of the multilevel architecture of deep learning (MADL) approach 1 using CIFAR-10 dataset.

983	1	1	1	0	1	1	0	6	1	98.8%
0	**988**	0	0	0	0	0	0	2	16	98.2%
4	0	**979**	4	1	2	2	1	0	0	98.6%
1	0	8	**951**	5	26	2	2	1	1	95.4%
0	0	3	9	**989**	4	0	7	0	0	97.7%
0	0	4	28	3	**963**	0	2	0	0	96.3%
0	0	4	4	0	1	**995**	0	1	0	99%
0	0	0	1	2	3	0	**988**	0	0	99.4%
12	0	1	1	0	0	0	0	**988**	2	98.4%
0	11	0	1	0	0	0	0	2	**980**	98.6%
98.3 %	98.8 %	97.9 %	95.1 %	98.9 %	96.3 %	99.5 %	98.8 %	98.8 %	98.0 %	**98.04 %**

Figure 8 | Confusion matrix of multilevel architecture of deep learning (MADL) approach 2 using CIFAR-10 dataset.

improves the accuracy compared to one single network. The paper also proposes an advanced customizable activation function that is comparable in its results to ReLU. MADL was experimented with CNN using two levels where the first level has Resnet-50 and inception-v3 and the second level has one simple CNN. Two fusion approaches were discussed, the first approach is taking the output layers of the networks in level 1 and feeding them to the network in level 2, while the second approach is introducing an additional fully connected layer to the networks in level 1 and passing the features from the additional fully connected layer to level 2. The experimentations showed that the second approach preserves more features and gives a higher improvement from level 1 to level 2. The experiments conducted on CIFAR-10 showed that the accuracy improved from 96.8% and 97.2% using Resnet-50 and inception-v3 respectively to 98.04%. Additional experiments were conducted using skin lesion dataset and 3 classes of Food-101 dataset. The improvement from level 1 to level 2 was 3.5% for skin lesion dataset and 0.6% for Food-101 dataset.

As a future work, more improvement can be done on the proposed advanced customizable activation function. The later can be experimented with MADL by training each separate network in MADL with a different activation function. This may lead to extracting diverse features.

ACKNOWLEDGMENT

The authors would like to thank Qatar National Library, QNL, for supporting in publishing the paper.

REFERENCES

[1] N. Hordri, S. Yuhaniz, S. Shamsuddin, Deep learning and its applications: a review, in Conference on Postgraduate Annual Research on Informatics Seminar, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia, 2016.

[2] H. Niu, W. Xu, H. Akbarzadeh, H. Parvin, A. Beheshti, H. Alinejad-Rokny, Deep feature learnt by conventional deep neural network, Comput. Electr. Eng. 84 (2020), 106656.

[3] C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, Activation functions: comparison of trends in practice and research for deep learning, CoRR, arXiv:1811.03378, 2018.

[4] A. Bhandare, M. Bhide, P. Gokhale, R. Chandavarkar, Applications of convolutional neural networks, Int. J. Comput. Sci. Inf. Technol. 7 (2016), 2206–2215.

[5] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017), 1137–1149.

[6] C. Cao, B. Wang, W. Zhang, X. Zeng, X. Yan, Z.e.a. Feng, An improved faster r-cnn for small object detection, IEEE Access. 7 (2019), 106838–106846.

[7] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: a comprehensive review, Neural Comput. 29 (2017), 2352–2449.

[8] N. Sharma, V. Jani, A. Mishra, An analysis of convolutional neural networks for image classification, Procedia Comput. Sci. 132 (2018), 377–384.

[9] C. Ding, D. Tao, Trunk-branch ensemble convolutional neural networks for video-based face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (2018), 1002–1014.

[10] C. Tian, Y. Xu, L. Fei, J. Wang, J. Wen, N. Luo, Enhanced CNN for image denoising, CAAI Trans. Intell. Technol. 4 (2019), 17–23.

[11] F. Sultana, A. Sufian, P. Dutta, Advancements in image classification using convolutional neural network, in Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Kolkata, India, 2018.

[12] J. Yim, J. Ju, H. Jung, J. Kim, Image classification using convolutional neural networks with multi-stage feature, in: J.H. Kim, W. Yang, J. Jo, P. Sincak, H. Myung (Eds.), Robot Intelligence Technology and Applications 3, Advances in Intelligent Systems and Computing, Springer, Cham, Switzerland, 2015.

[13] M.A. Nasichuddin, T.B. Adji, W. Widyawan, Performance improvement using CNN for sentiment analysis, Int. J. Inf. Technol. Electr. Eng. 2 (2018), 9–14.

[14] H. Zhou, Y. Wang, X. Lei, Y. Liu, A method of improved cnn traffic classification, in 2017 13th International Conference on Computational Intelligence and Security (CIS), Hong Kong, China, 2017.

[15] P. Szeto, H. Parvin, M. Mahmoudi, B. Tuan, K. Pho, Deep neural network as deep feature learner, J. Intell. Fuzzy Syst. 39 (2020), 355–369.

[16] Z. Horn, L. Auret, J. McCoy, C. Aldrich, B. Herbst, Performance of convolutional neural networks for feature extraction in froth flotation sensing, IFAC-PapersOnLine. 50 (2017), 13–18.

[17] M. Meselhy Eltoukhy, I. Faye, S. Brahim Belhaouari, A statistical based feature extraction method for breast cancer diagnosis in digital mammogram using multiresolution representation, Comput. Biol. Med. 42 (2012), 123–128.

[18] M. Khoshdeli, R. Cong, B. Parvin, Detection of nuclei in H and E stained sections using convolutional neural networks, in Conference on IEEE International Conference on Biomedical Health Informatics, Orlando, FL, USA, 2017.

[19] M.K. Benkaddour, A. Bounoua, Feature extraction and classification using deep convolutional neural networks, pca and svc for face recognition, Traitement Signal. 34 (2017), 77–91.

[20] X. Lu, X. Duan, X. Mao, Y. Li, X. Zhang, Feature extraction and fusion using deep convolutional neural networks for face detection, Math. Probl. Eng. 2017 (2017), 1–9.

[21] R. Minetto, M.P. Segundo, S. Sarkar, Hydra: an ensemble of convolutional neural networks for geospatial land classification, IEEE Trans. Geosci. Remote Sensing. 57 (2019), 6530–6541.

[22] W. Xu, H. Parvin, H. Izadparast, Deep learning neural network for unconventional images classification, Neural Process. Lett. 52 (2020), 169–185.

[23] T.G. Dietterich, Ensemble methods in machine learning, in: Multiple Classifier Systems Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, Germany, 2000, pp. 1–15.

[24] A.J.C. Sharkey, On combining artificial neural nets, Connect. Sci. 8 (1996), 299–314.

[25] H.-H. Zhao, H. Liu, Multiple classifiers fusion and CNN feature extraction for handwritten digits recognition, Granul. Comput. 5 (2020), 411–418.

[26] G. Qi, H. Wang, M. Haner, C. Weng, S. Chen, Z. Zhu, Convolutional neural network based detection and judgement of environmental obstacle in vehicle operation, CAAI Trans. Intell. Technol. 4 (2019), 80–91.

[27] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, Mixup: beyond empirical risk minimization, arXiv e-prints, arXiv:1710.09412arXiv:1710.09412, 2017.

[28] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, A. Courville, D. Lopez-Paz, Y. Ben-gio, Manifold mixup: better representations by interpolating hidden states, arXiv e-prints, arXiv:1806.05236arXiv:1806.05236, 2018.

[29] P. Savarese, M. Maire, Learning implicitly recurrent CNNs through parameter sharing, arXiv e-prints, arXiv:1902.09701arXiv:1902.09701, 2019.

[30] H. Zhang, Y.N. Dauphin, T. Ma, Fixup initialization: residual learning without normalization, arXiv e-prints, arXiv:1901.09321arXiv:1901.09321, 2019.

[31] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, arXiv e-prints, arXiv:1709.01507arXiv:1709.01507, 2017.

[32] H. Cai, L. Zhu, S. Han, ProxylessNAS: direct neural architecture search on target task and hardware, arXiv e-prints, arXiv:1812.00332arXiv:1812.00332, 2018.

[33] X. Wang, D. Kihara, J. Luo, G.-J. Qi, EnAET: self-trained ensemble autoencoding transformations for semi-supervised learning, arXiv e-prints, arXiv:1911.09265arXiv:1911.09265, 2019.

[34] S. Lim, I. Kim, T. Kim, C. Kim, S. Kim, Fast autoaugment, arXiv e-prints, arXiv:1905.00397arXiv:1905.00397, 2019.

[35] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, N. Houlsby, Large scale learning of general visual representations for transfer, arXiv e-prints, arXiv:1912.11370arXiv:1912.11370, 2019.

[36] P. Radiuk, Impact of training set batch size on the performance of convolutional neural networks for diverse datasets, Inf. Technol. Manag. Sci. (2017).

[37] Y. Ioannou, D. Robertson, R. Cipolla, A. Criminisi, Deep roots: improving CNN efficiency with hierarchical filter groups, in IEEE Conference On Computer Vision And Pattern Recognition (CVPR), Honolulu, HI, USA, 2017.

[38] S. Uchida, S. Ide, B.K. Iwana, A. Zhu, A further step to perfect accuracy by training cnn with larger data, in 15th International Conference On Frontiers In Handwriting Recognition (ICFHR), Shenzhen, China, 2016.

[39] Neural activation - an overview — sciencedirect topics, 2019. https://www.sciencedirect.com/topics/psychology/neural-activation

[40] P. Lansky, O. Pokora, J.-F. Rospars, Classification of stimuli based on stimulus response curves and their variability, Brain Res. 1225 (2008), 57–66.

[41] Find open datasets and machine learning projects — kaggle, 2020. https://www.kaggle.com/datasets

[42] J. Chu, M.A. Shaikh, M. Chauhan, L. Meng, S. Srihari, Writer verification using cnn feature extraction, in 16th International Conference On Frontiers In Handwriting Recognition (ICFHR), Niagara Falls, NY, USA, 2018.

[43] S. Dara, P. Tumma, Feature extraction by using deep learning:a survey, in Second International Conference On Electronics Communication And Aerospace Technology (ICECA), Coimbatore, India, 2018.
[44] Cifar-10 and cifar-100 datasets, 2009. https://www.cs.toronto.edu/kriz/cifar.html
[45] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, Technical Report, 2009.
[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in IEEE Conference On Computer Vision And Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016.