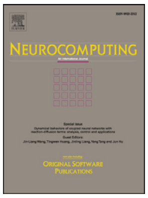
Neurocomputing 378 (2020) 112–119

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Impact of fully connected layers on performance of convolutional neural networks for image classification

S.H. Shabbeer Basha*, Shiv Ram Dubey, Viswanath Pulabaigari, Snehasis Mukherjee

Indian Institute of Information Technology Sri City, Andhra Pradesh 517646, India

ARTICLE INFO

Article history:
Received 23 March 2019
Revised 19 July 2019
Accepted 6 October 2019
Available online 17 October 2019

Communicated by Dr. Ma Jiayi

Keywords:
Convolutional neural networks
Fully connected layers
Image classification
Shallow vs deep CNNs
Wider vs deeper datasets

ABSTRACT

The Convolutional Neural Networks (CNNs), in domains like computer vision, mostly reduced the need for handcrafted features due to its ability to learn the problem-specific features from the raw input data. However, the selection of dataset-specific CNN architecture, which mostly performed by either experience or expertise is a time-consuming and error-prone process. To automate the process of learning a CNN architecture, this paper attempts at finding the relationship between Fully Connected (FC) layers with some of the characteristics of the datasets. The CNN architectures, and recently datasets also, are categorized as deep, shallow, wide, etc. This paper tries to formalize these terms along with answering the following questions. (i) *What is the impact of deeper/shallow architectures on the performance of the CNN w.r.t. FC layers?*, (ii) *How the deeper/wider datasets influence the performance of CNN w.r.t. FC layers?*, and (iii) *Which kind of architecture (deeper/shallower) is better suitable for which kind of (deeper/wider) datasets*. To address these findings, we have performed experiments with four CNN architectures having different depths. The experiments are conducted by varying the number of FC layers. We used four widely used datasets including CIFAR-10, CIFAR-100, Tiny ImageNet, and CRCHistoPhenotypes to justify our findings in the context of image classification problem. The source code of this work is available at https://github.com/shabbeersh/Impact-of-FC-layers.

1. Introduction and related works

The popularity of Convolutional Neural Networks (CNN) is growing significantly for various application domains related to computer vision, which include object detection [1], segmentation [2], localization [3], and many more in recent years. Despite the success of deep learning models, our theoretical understanding about neural networks remains limited. Careful selection of network width (number of neurons in FC layers, number of filters in convolution layers) and network depth (number of trainable layers) plays a vital role in designing deep neural networks in order to obtain better performance. In this paper, we made an attempt to find some of the factors which affect the performance of the CNN w.r.t. Fully Connected (FC) layers in the context of image classification. We have also studied the possible interrelationship between the presence of FC layers in CNN, the depth of the CNN, and the depth of the dataset.

Deep neural networks usually provide better results in the field of machine learning and computer vision compared to the handcrafted feature descriptors [1]. From the available literature, it is apparent that every CNN architecture have one or more FC layers depending on the architecture's depth. To mention a few, AlexNet [4] consists of 5 convolutional (*Conv*) layers and 3 FC layers. The FC layers are placed after all the Conv layers. Zeiler and Fergus [5] made minimal changes to AlexNet with better hyper-parameter settings in order to generalize it over other datasets. This model is called ZFNet which also has 3 FC layers along with 5 convolution layers. In 2014, Simonyan and Zisserman [6] further extended the AlexNet model to VGG-16 with 16 learnable layers including 3 FC layers towards the end of the architecture. Later on, many CNN models have been introduced with an increasing number of learnable layers. Szegedy et al. [7] proposed a 22-layer architecture called GoogLeNet, which has a single FC (output) layer. In 2015, He et al. [8] introduced ResNet with 152 trainable layers where the last layer is fully connected. However, all the above CNN architectures are proposed for large-scale ImageNet dataset [9]. Recently, Basha et al. [10] proposed a CNN based classifier called RCCNet, which is responsible for classifying the routine colon cancer cells of dimension $32 \times 32 \times 3$. This CNN model has 7 learnable layers including 3 FC layers.

Necessity of fully connected layers in CNN: In a shallow CNN model, the features generated by the final convolutional layer

* Corresponding author.
E-mail address: shabbeer.sh@iiits.in (S.H.S. Basha).

https://doi.org/10.1016/j.neucom.2019.10.008
0925-2312/© 2019 Elsevier B.V. All rights reserved.

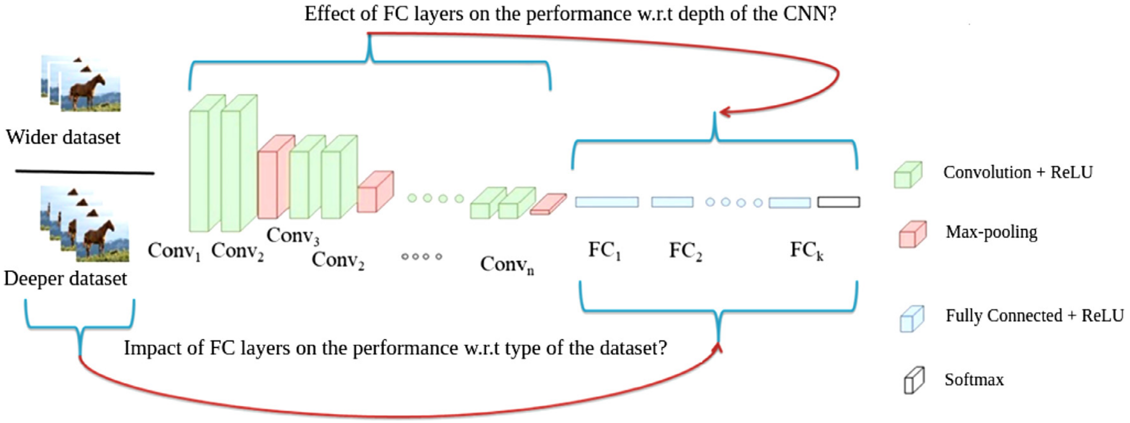


Fig. 1. The illustration of the effect of deeper/wider datasets and depth of CNN (i.e., the number of the Convolutional layers, n) over the number of FC layers (i.e., k). A typical plain CNN architecture has Convolutional (learnable), Max-pooling (non-learnable) and FC (learnable) layers.

correspond to a portion of the input image as its receptive field does not cover the entire spatial dimension of the image. Thus, few FC layers are mandatory in such a scenario. Despite their pervasiveness, the hyperparameters like the number of FC layers and number of neurons required in FC layers for a given CNN architecture to obtain better performance are not explored.

In a typical deep neural network, the FC layers comprise most of the parameters of the network. AlexNet has 60 million parameters, out of which 58 million parameters correspond to the FC layers [4]. Similarly, VGGNet has a total of 138 million parameters, out of which 123 million parameters belong to FC layers [6]. This huge number of trainable parameters in FC layers are required to fit complex nonlinear discriminant functions in the feature space into which the input data elements are mapped. However, this large number of parameters may result in over-fitting the classifier (CNN). To reduce the amount of over-fitting, Xu et al. [11] proposed a CNN architecture called SparseConnect where the connections between *FC* layers are sparsed.

The most common question in deep learning community is “Is Depth needed for neural networks?”. A series of studies have been conducted to address this question over the past decades. In 1989, Cybenko et al. [12] showed that a two-layer feed forward neural network with sigmoid activation can approximate any continuous function with small error. Delalleau et al. [13] proved that the deep neural networks can be used to represent a family of functions efficiently than shallow networks. Recently, Lu et al. [14] proved Universal Approximation Theorem for Width-Bounded-ReLU networks. The width (number of neurons in FC layers, number of filters in convolution layers) and depth (number of trainable layers) are two key elements in neural network architecture design. These two parameters should be carefully tuned to obtain better performance. In this paper, we observe the role of FC layers on the performance of CNN.

The effect of deep or shallow networks on different kind of datasets is well explored in the literature to study the behavioral interrelationship between depth of dataset and the CNN [15,16]. Mhaskar et al. [15] extended a framework for their previous work [16] to investigate when deep networks are better than shallow networks using a Directed Acyclic Graph (DAG). Montufar et al. [17] performed a study to find the complexity of the functions computable by deep neural networks with linear activations.

To the best of our knowledge, no effort has been made in the literature to analyze the role of FC layers in CNN for image classification. In this paper, we investigate the impact of FC layers on the performance of the CNN with a rigorous analysis from various aspects. In brief, the contributions of this paper are summarized as follows.

- We perform a systematic study to observe the effect of deeper/shallower architectures on the performance of CNNs with varying number of FC layers.
- We observe the effect of deeper/wider datasets on the performance of CNN w.r.t. FC layers.
- We generalize one important finding of Bansal et al. [18] to choose deeper or shallow architecture based on the depth of the dataset. In [18], they have reported the same in the context of face recognition, Whereas, we made a rigorous study to generalize this observation over different kinds of datasets.
- To make the empirical justification of our findings, we have conducted the experiments on different modalities (i.e., natural and bio-medical images) of image datasets like CIFAR-10, CIFAR-100 [19], Tiny ImageNet [20], and CRCHistoPhenotypes [21].

Next, we illustrate the developed deep and shallow CNN architectures to conduct the experiments in Section 2. Experimental setup including training details, evaluation criteria, and datasets are discussed in Section 3. Section 4 presents a detailed study of the observations found in this paper. At last, Section 5 concludes the paper.

2. Developed CNN architectures

The main objective of this paper is to analyze the impact of different hyperparameters realted to FC layers (the number of FC layers and the number of neurons) over the performance. Interdependency between the characteristics of both the datasets and the networks are explored w.r.t. FC layers as shown in Fig. 1. In order to conduct a rigorous experimental study, we have implemented four CNN models among which three CNN models are plain architectures. Another model involves skip connections as in ResNet [8]. These models are termed as CNN-1, CNN-2, CNN-3, and CNN-4.

2.1. Pre-requisites

A deep convolutional neural network N has two main blocks C, F which are responsible for extracting the problem-specific features and decision making, respectively. Mathematically N can be represented as $N = \{C, F\}$, where C is combination of convolution (Conv), pooling (Pool), and Batch Normalization (BN) layers $C = \{Conv, Pool, BN\}$ and F is series of Fully Connected(FC) layers $F = \{FC_1, FC_2, \ldots FC_n\}$. The feature extraction block C receives input of dimension $H \times W \times D$ and produces a feature map of dimension $H^{'} \times W^{'} \times D^{'}$. The resultant output feature map is stretched into a single column vector of dimension $1 \times m$ (assuming

$m = H^{'} \times W^{'} \times D^{'}$). The flattened feature vector of dimension $1 \times m$ is given as input to the first FC layer FC_1. In general, the number of neurons in layer $FC_i = n_i$. Since the full connectivity is maintained in FC layers, the connections have weights (parameters) $W \in \mathcal{R}^d$. The output of the nodes in FC_{i+1} layer are computed as follows,

$$\sum_{i=1}^{n_i} \sum_{j=1}^{n_{i+1}} \psi(h_i W_{ji} + b_j) \tag{1}$$

where h_i is the output of the node n_i in layer L_i and ψ is non-linear activation function. The number of operations performed to compute the output of layer L_{i+1} are $n_i.n_{i+1} + n_{i+1}$, where n_i, n_{i+1} are number of neurons in layers L_i, L_{i+1}, respectively. The final (ouput) FC layer has c number of neurons, where c represents the number of classes correspond to a dataset.

Deep and shallow CNNs: As per the published literature [17,22], a neural network is referred to as shallow if it has single fully connected (hidden) layer. Whereas, a deep CNN consists of convolution layers, pooling layers, and FC layers. However, in this paper, we assume a CNN model N_1 as deep/shallow compared to another CNN model N_2, if the number of trainable layers in N_1 is more/less than N_2, respectively.

2.2. CNN-1 architecture

AlexNet [4] is well-known CNN architecture, which won the first ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 [23] with a huge performance gain as compared to the best results of that time using handcrafted features. The AlexNet architecture was proposed for the images of dimension 227 × 227 × 3, we made minimal changes to the model to fit for low-resolution images. We name this model as CNN-1. Initially, the input image dimension is up-sampled from 32 × 32 × 3 to 35 × 35 × 3 in the case of CRCHistoPhenotypes [21], CIFAR-10, CIFAR-100 [19] datasets. Whereas, the images of Tiny ImageNet dataset [20] are down-sampled from 64 × 64 × 3 to 35 × 35 × 3. The 1st convolutional layer *Conv*1 produces 31 × 31 × 96 dimensional feature vector by applying 96 filters of dimension 5 × 5 × 3. The *Conv*1 layer is followed by another Convolution layer (*Conv*2), which produces 27 × 27 × 256 dimensional feature map by convolving 256 filters of size 5 × 5 × 96. The remaining layers of the CNN-1 model are similar to the AlexNet architecture proposed in [4]. The CNN-1 model with a single FC layer (i.e., the output FC layer) consists of following number of trainable parameters, 4,152,906 for CIFAR-10 dataset, 8,046,756 for CIFAR-100 dataset, 12,373,256 for Tiny ImageNet dataset, and 3,893,316 for CRCHistoPhenotypes dataset. Note that, the number of trainable parameters are different for each dataset due to the different number of classes present in the datasets which leads to the varying number of trainable parameters in the output FC layer. The detailed specifications of the CNN-1 model are given in Table 1.

Table 1
The CNN-1 architecture having 5 *Conv* layers. The *S*, *P*, and *BN* denote stride, padding, and batch normalization, respectively. The output (FC) layer has 10, 100, 200, and 4 neurons in the case of CIFAR-10, CIFAR-100, Tiny ImageNet, and CRCHistoPhenotypes datasets, respectively.

Input:	Image dimension (35 × 35 × 3)
[layer 1]	Conv. (5,5,96), S=1, P=0; ReLU; BN;
[layer 2]	Conv. (5,5,256), S=1, P=0; ReLU; BN;
[layer 3]	Pool., S=2, P=0;
[layer 4]	Conv. (3,3,384), S=1, P=1; ReLU;
[layer 5]	Conv. (3,3,384), S=1, P=1; ReLU;
[layer 6]	Conv. (3,3,256), S=1, P=1; ReLU;
[layer 7]	Flatten; 43264;
Output:	(FC layer) Predicted Class Scores

2.3. CNN-2 architecture

Another CNN model is designed based on the CIFAR-VGG [24] model by removing some *Conv* layers from the model. We name this model as CNN-2. The CNN-2 has 6 blocks, where first 5 blocks have two consecutive *Conv* layers followed by a *Pool* layer. Finally, the sixth block has a FC (output) layer which generates the class scores. The input to this model is an image of dimension 32 × 32 × 3. To meet this requirement, images of the Tiny ImageNet dataset are down-sampled from 64 × 64 × 3 to 32 × 32 × 3. The CNN-2 architecture corresponds to $9,416,010$, $9,462,180$, $9,513,480$, and $9,412,932$ trainable parameters in the case of CIFAR-10, CIFAR-100, Tiny ImageNet, and CRCHistoPheno-Types datasets, respectively. The CNN-2 model specifications are given in Table 2.

Table 2
The CNN-2 model having 10-*Conv* layers. The *S, P, BN*, and DP_f denote the stride, padding, batch normalization, and dropout with a factor of *f*. The output layer has 10, 100, 200, and 4 neurons in the case of CIFAR-10, CIFAR-100, Tiny ImageNet, and CRCHistoPhenotypes datasets, respectively.

Input:	Image dimension (32 × 32 × 3)
[layer 1]	Conv. (3,3,64), S=1, P=1; ReLU; BN, $DP_{0.3}$
[layer 2]	Conv. (3,3,64), S=1, P=1; ReLU; BN;
[layer 3]	Pool., S=2, P=0;
[layer 4]	Conv. (3,3,128), S=1, P=1; ReLU; BN, $DP_{0.4}$
[layer 5]	Conv. (3,3,128), S=1, P=1; ReLU; BN;
[layer 6]	Pool., S=2, P=0;
[layer 7]	Conv. (3,3,256), S=1, P=1; ReLU; BN, $DP_{0.4}$
[layer 8]	Conv. (3,3,256), S=1, P=1; ReLU; BN;
[layer 9]	Pool., S=2, P=0;
[layer 10]	Conv. (3,3,512), S=1, P=1; ReLU; BN, $DP_{0.4}$
[layer 11]	Conv. (3,3,512), S=1, P=1; ReLU; BN;
[layer 12]	Pool., S=2, P=0;
[layer 13]	Conv. (3,3,512), S=1, P=1; ReLU; BN, $DP_{0.4}$
[layer 14]	Conv. (3,3,512), S=1, P=1; ReLU; BN;
[layer 15]	Pool., S=2, P=0;
[layer 16]	Flatten; 512;
Output:	(FC layer) Predicted Class Scores

2.4. CNN-3 architecture

Most of the popular CNN models like AlexNet [4], VGG-16 [6], GoogLeNet [7], and many more were proposed for high dimensional image dataset called ImageNet [9]. On the other hand, the low dimensional image datasets such as CIFAR-10/100 have rarely got benefited from the CNNs. Liu et al. [24] introduced CIFAR-VGG architecture, which is basically a 16 layer deep CNN architecture proposed for CIFAR-10. We have utilized CIFAR-VGG model as the third deep neural network to observe the impact of FC layers in CNN and named as CNN-3 in this paper. The input to this model is an image of dimension 32 × 32 × 3. To meet this requirement, images of the Tiny ImageNet dataset are down-sampled from 64 × 64 × 3 to 32 × 32 × 3. The CNN-3 architecture with a single FC (output) layer corresponds to $14,728,266$, $14,774,436$, $14,825,736$, and $14,725,188$ trainable parameters in the case of CIFAR-10 [19], CIFAR-100 [19], Tiny ImageNet [20], and CRCHistoPhenotypes [21] datasets, respectively.

2.5. CNN-4 architecture

In 2016, Zagoruyko and Komodakis [25] introduced Wide Residual Network (WRN) for classification of low-scale images. We considered WRN of depth 16 as fourth CNN model (CNN-4) to observe the imapct of FC layers on the performance of CNN. The CNN-4 model with single FC layer comprises 10,961,370, 11,007,540, 11,058,840 and 10,958,292 trainable parameters in the

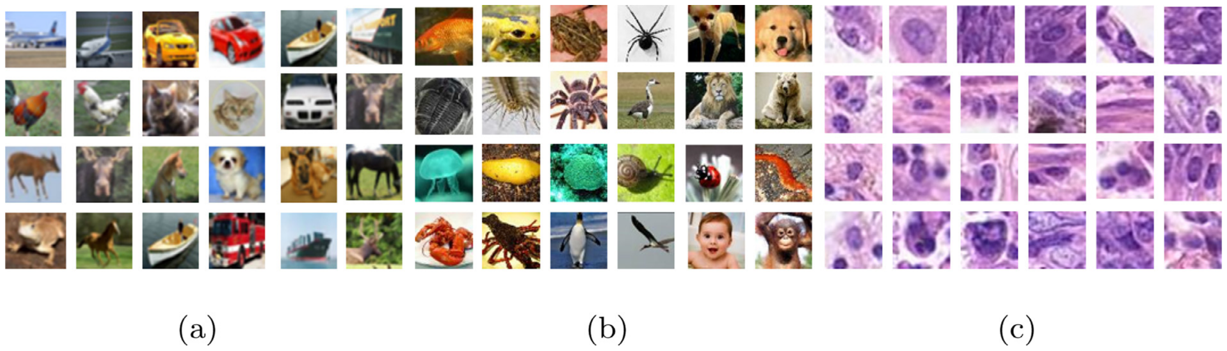


Fig. 2. (a) A few sample images from CIFAR-10/100 dataset [19]. (b) A random sample images from Tiny ImageNet dataset [20]. (c) Example images from CRCHistoPhenotypes dataset [21] with each row represents the images from one category.

case of CIFAR-10, CIFAR-100, iny ImageNet, and CRCHistoPhenoTypes datasets, respectively.

3. Experimental setup

This section describes the experimental setup including the training details, datasets used for the experiments, and the evaluation criteria to judge the performance of the CNN models.

3.1. Training details

The classification experiments are conducted on different modalities of image datasets to provide the empirical justifications of our findings reported in this paper. The initial value of the learning rate is 0.1 and it is decreased by a factor of 2 for every 20 epochs. The Rectified Linear Unit (*ReLU*) based non-linearity [4] is used as the activation function after every *Conv* and FC layer (except the output FC layer) in all the CNN models discussed in Section 2. The Batch Normalization (i.e., *BN*) [26] is employed after *ReLU* of each *Conv* and FC layer, except final FC layer in CNN-2 and CNN-3 architectures. Whereas, in the case of CNN-1, the Batch Normalization is used only with the first two *Conv* layers as mentioned in Table 1. To reduce the amount of over-fitting, we have used a popular regularization method called Dropout (i.e., *DP*) [27] after some Batch-Normalization layers as summarized in Table 2 for CNN-2. For CNN-3, the *DP* layers are used as per the CIFAR-VGG model [24]. The Batch Normalization and dropout are employed for CNN-4 as in [25]. In order to find the impact of Fully Connected (FC) layers on the performance of CNN, any added FC layer has the *ReLU, BN* and *DP* by default. Along with dropout, various data augmentations techniques like rotation, horizontal flip, and vertical flip are also applied to reduce the amount of over-fitting. The implemented CNN architectures are trained for 250 epochs using Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9.

3.2. Evaluation criteria

To evaluate the performance of the developed CNN models (i.e., CNN-1, CNN-2, CNN-3, and CNN-4), we have considered the classification accuracy as the performance evaluation metric.

3.3. Datasets

To find out the empirical observations addressed in this paper, we have conducted the experiments on different modalities of datasets such as CIFAR-10 [19], CIFAR-100 [19], Tiny ImageNet [20] (i.e., the natural image datasets), and CRCHistoPhenotypes [21] (i.e., the medical image dataset).

3.3.1. CIFAR-10

The CIFAR-10 [19] is the most popular tiny image dataset consists of 10 different categories of images, where each class has 6000 images. The dimension of each image is $32 \times 32 \times 3$. To train the deep neural networks, we have used the training set (i.e., $50,000$ images) of the CIFAR-10 dataset, and remaining data (i.e., $10,000$ images) is utilized to validate the performance of the models. A few samples of images from the CIFAR-10 dataset are shown in Fig. 2(a).

3.3.2. CIFAR-100

The CIFAR-100 [19] dataset is similar to CIFAR-10, except that CIFAR-100 has 100 classes. In our experimental setting, the 50,000 images are used to train the CNN models and the remaining 10,000 images are used to validate the performance of the models. Similar to CIFAR-10, the dimension of each image is $32 \times 32 \times 3$. The sample images are shown in Fig. 2(a).

3.3.3. Tiny ImageNet

The Tiny ImageNet dataset [20] consists a subset of ImageNet [9] images. This dataset has a total of 200 classes and each class has 500 training and 50 validation images. In other words, we have used $100,000$ images for training and $10,000$ images for validating the performance of the models. The dimension of each image is $64 \times 64 \times 3$. The example images of the Tiny ImageNet dataset are portrayed in Fig. 2(b).

3.3.4. CRCHistoPhenotypes

In order to generalize the observations reported in this paper, we have used a medical image dataset (consists of routine colon cancer nuclei cells) called “CRCHistoPhenotypes” [21], which is publicly available.[1] This colon cancer dataset consists a total of $22,444$ nuclei patches that belong to the four different classes, namely, ‘Epithelial’, ‘Inflammatory’, ‘Fibroblast’, and ‘Miscellaneous’. In total, 7722 images belong to the ‘Epithelial’ class, 5712 images belong to the ‘Fibroblast’ class, 6971 images belong to the ‘Inflammatory’ class, and the ‘Miscellaneous’ class has remaining 2039. The dimension of each nuclei patch is $32 \times 32 \times 3$. For training the CNN models, 80% of entire data (*i.e.*, $17,955$ images) is utilized and remaining 20% data (i.e., 4489 images) is used to validate the performance of the models. The sample images are displayed in Fig. 2(c).

Deeper vs Wider datasets [18]: For any two datasets with roughly same number of images, one dataset is said to be deeper [18] than another dataset, if it has more number of images per class in the training set. The other dataset which has a lower number of images per class (i.e., more number of classes compared to another

[1] https://warwick.ac.uk/fac/sci/dcs/research/tia/data/crchistolabelednucleihe

Table 3
The effect of depth of the CNN models (i.e., CNN-1, CNN-2, CNN-3, and CNN-4) on FC layers for the CIFAR-10 dataset is shown in this table. The best and 2nd best accuracies are highlighted in bold and italic, respectively. For example, the CNN-2 model produces the best accuracy of 92.29% for three FC layers with 4096, 256, and 10 neurons and the 2nd best accuracy of 92.02% for two FC layers with 256 and 10 neurons.

CNN-1	CNN-2	CNN-3	CNN-4
Output FC layer (44.29)	Output FC layer (91.46)	*Output FC layer (92.05)*	Output FC layer (92.55)
10 × 10 (88.67)	10 × 10 (91.14)	10 × 10 (91.03)	10 × 10 (92.15)
16 × 10 (88.72)	16 × 10 (91.58)	16 × 10 (91.77)	16 × 10 (92.79)
32 × 10 (88.93)	*32 × 10 (91.99)*	32 × 10 (92.02)	32 × 10 (92.73)
64 × 10 (89.72)	64 × 10 (91.82)	64 × 10 (91.8)	64 × 10 (93.32)
128 × 10 (89.2)	128 × 10 (91.86)	128 × 10 (89.2)	128 × 10 (93.13)
256 × 10 (89.23)	*256 × 10 (92.02)*	256 × 10 (89.23)	256 × 10 (92.96)
512 × 10 (88.95)	512 × 10 (90.98)	512 × 10 (91.78)	512 × 10 (92.63)
1024 × 10 (89.56)	1024 × 10 (91.54)	**1024 × 10 (92.22)**	1024 × 10 (92.67)
2048 × 10 (87.4)	2048 × 10 (91.27)	2048 × 10 (91.59)	2048 × 10 (92.79)
4096 × 10 (86.27)	4096 × 10 (87.51)	4096 × 10 (90.68)	4096 × 10 (93.01)
64 × 64 × 10 (89.35)	256 × 256 × 10 (91.97)	1024 × 1024 × 10 (91.27)	64 × 64 × 10 (92.92)
128 × 64 × 10 (89.71)	512 × 256 × 10 (91.92)	2048 × 1024 × 10 (91.43)	128 × 64 × 10 (92.99)
256 × 64 × 10 (89.79)	1024 × 256 × 10 (91.53)	4096 × 1024 × 10 (91.94)	256 × 64 × 10 (92.65)
512 × 64 × 10 (89.88)	2048 × 256 × 10 (91.95)	–	512 × 64 × 10 (92.69)
1024 × 64 × 10 (90)	**4096 × 256 × 10 (92.29)**	–	1024 × 64 × 10 (92.59)
2048 × 64 × 10 (90.28)	4096 × 4096 × 256 × 10 (91.64)	–	2048 × 64 × 10 (92.89)
4096 × 64 × 10 (90.59)	–	–	4096 × 64 × 10 (93.37)
4096 × 4096 × 64 × 10 (90.77)	–	–	4096 × 4096 × 64 × 10 (92.64)
4096 × 4096 × 4096 × 64 × 10 (90.74)	–	-	–

Table 4
The best validation accuracies obtained over CIFAR-10, CIFAR-100, Tiny ImageNet and CRCHistoPhenotypes datasets using four CNN models (i.e., CNN-1, CNN-2, CNN-3, and CNN-4) are depicted in this table. The results are presented in terms of the FC layer structures and validation accuracy.

S.No.	Architecture	Dataset			
		CIFAR-10	CIFAR-100	Tiny ImageNet	CRCHistoPhenoTypes
1	CNN-1	4096 × 4096 × 64 × 10 (90.77)	4096 × 4096 × 2048 × 100 (69.21)	4096 × 4096 × 1024 × 200 (50.1) 4096 × 100 (48.46) (Setting-1)	2048 × 256 × 4 (82.53)
2	CNN-2	4096 × 256 × 10 (92.29)	4096 × 100 (62.28)	512 × 200, 1024 × 200 (41.84) 512 × 100 (36.8) (Setting-1)	512 × 4 (84.89)
3	CNN-3	1024 × 10 (92.22)	Single FC (output) layer (66.98)	Single FC (output) layer (40.27) 256 × 100 (36.38) (Setting-1)	128*x*4 (84.94)
4	CNN-4	4096 × 64 × 4 (93.37)	Single FC (output) layer (68.52)	Single FC (output) layer (48.95) Single FC (output) layer (52.1) (Setting-1)	128 × 4 (83.98)
5	SVM loss	CNN-4 78.89	CNN-1 (58.46)	CNN-1 (47.82)	CNN-4 (76.18)

one) in the training set is called the wider dataset. For example, CIFAR-10 and CIFAR-100 [19], both the datasets have 50,000 images in the training set. The CIFAR-10 is a deeper dataset since it has 5000 images per class in the training set. On the other hand, the CIFAR-100 is wider dataset because it has only 500 images per class.

4. Results and analysis

We have conducted extensive experiments to observe the useful practices in deep learning for the usage of Convolutional Neural Networks (CNNs). The four CNN models discussed in Section 2 are implemented to perform the experiments on publicly available CIFAR-10/100, Tiny ImageNet, and CRCHistoPhenotypes datasets. The results in terms of the classification accuracy are reported in this paper.

4.1. Impact of FC layers on the performance of the CNN w.r.t. to the depth of the CNN

To observe the effect of deeper/shallow architectures on FC layers, initially, the CNN models are trained with a single FC (output) layer. Then another FC layer is added manually before the output (FC) layer to observe the gain/loss in the performance due to the addition of the new FC layer. The number of neurons is chosen (for newly added FC layer) starting from the number of classes to all multiples of 2 (i.e, powers of 2 such as 16, 32, etc.), which is greater than the number of classes and up to 4096. For instance, in the case of CIFAR-10 dataset [19], the experiments are conducted by varying the number of neurons in the newly added FC layer with $10, 16, 32, 64, \ldots, 4096$ number of neurons. In the next step, one more FC layer is added before the recently added FC layer. The number of neurons for newly added FC layer is chosen, ranging from the value for which best performance is obtained in the previous setting to 4096. Suppose we obtained the best performance over CIFAR-10 using CNN-1 having two FC layers with 512, 10 neurons CIFAR-10. Then, we observed the performance of the model by adding another FC layer with 512, 1024, 2048, and 4096 number of neurons. The details like the number of FC layers, number of neurons in each FC layer, best classification accuracies obtained for CIFAR-10 dataset using the four CNN models are shown in Table 3. It is evident from Table 3 that the deeper architectures (i.e., CNN-4, CNN-3,and CNN-2 with more convolution layers (require relatively less number of FC layers and also less number of neurons in FC layers compared to the shallow architecture (i.e., CNN-1 with 5 *Conv* layers) for CIFAR-10 dataset.

To generalize the above-mentioned observation, we have computed the results by varying the number of FC layers over other datasets and reported the best performance in Table 4. From Table 4, similar findings are noticed that the deeper architectures do not require more FC layers. On the other hand, the shallow architectures (such as CNN-1) require more FC layers in order to

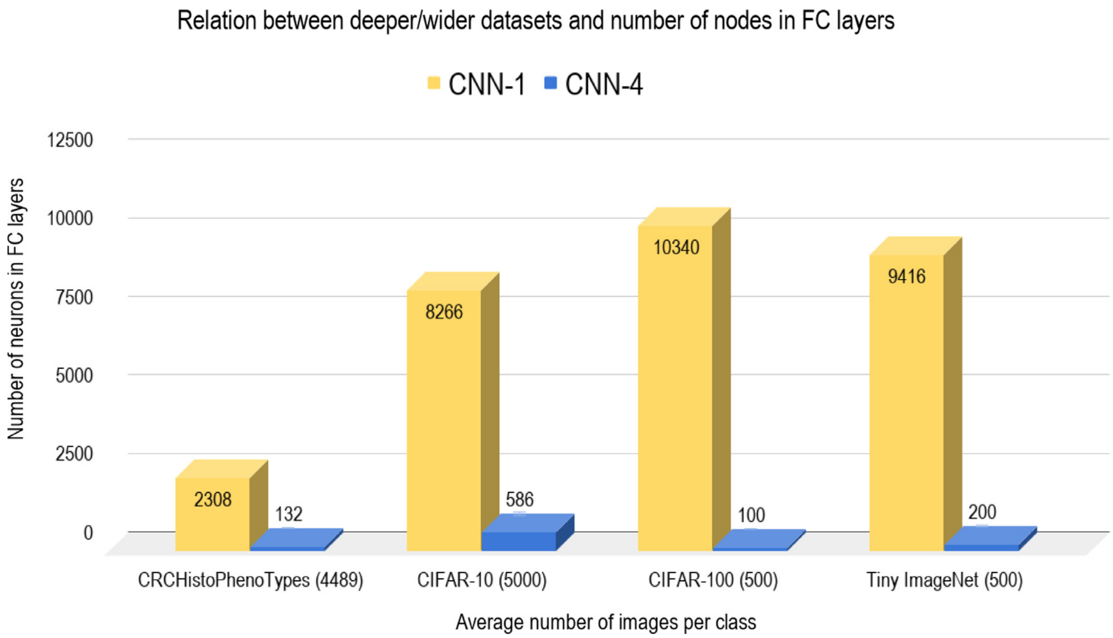


Fig. 3. The effect of deeper/wider datasets on FC layers of CNN. For wider datasets, deeper architecture (CNN-4) requires relatively less number of neurons in FC layers than deeper datasets. On the other hand, for wider datasets, shallow architecture (CNN-1) requires relatively large number of neurons in FC layers compared to deeper datasets.

Table 5
The characteristics of CIFAR-10, CIFAR-100, Tiny ImageNet, and CRCHistoPhenotypes datasets are presented in this table. Here, *N* represents the average number of images per class in the training set, *C* represents the number of classes corresponding to a dataset, *Tr* and *Va* are the number of images in the Training and Validation sets, respectively.

Dataset	N	C	Tr	Va
CIFAR-10	5000	10	50,000	10,000
CIFAR-100	500	100	50,000	10,000
Tiny ImageNet	500	200	80,000	20,000
CRCHistoPhenotypes	4489	4	17,955	4489

obtain better performance for any dataset. The reasoning for such a behavior is related to the type of features being learned by the *Conv* layers. In general, CNN architecture learns the hierarchical features from raw images. Zeiler and Fergus [5] shown that the early layers learn the low-level features, whereas the deeper layers learn the high-level (problem specific) features. It means that the final *Conv* layer of shallow architecture produces less abstract features as compared to the deeper architecture. Thus, the number of FC layers needed for shallow architecture is more as compared to the deeper architectures. To provide powerful evidence to the findings reported in this paper, we have conducted experiments by considering the half of the images (images belong to 100 classes) of Tiny ImageNet dataset. We name this configuration as setting-1 (refer Table 4). We have also considered SVM (hinge) loss to compare the results that we obtained using the popular cross-entropy loss function. The CNN architectures through which the best validation is obtained (using FC layer structure reported in Table 4) are trained using hinge loss. The same results are specified in the last row of Table 4.

4.2. Effect of FC layers on the performance of the CNN model w.r.t. to different types of datasets

We have used two kinds of datasets (deeper and wider) to analyze the effect of FC layers on the performance of CNN. Table 5 presents the characteristics like average number of images per class in the training set (*N*), number of classes (*C*), number of training images (*Tr*), and validation images (*Va*) of four datasets discussed in Section 3.3.

From Fig. 3, we can observe that shallow architecture CNN-1 (less deeper than CNN-2, CNN-3, and CNN-4) requires more neurons in FC layers for wider datasets compared to deeper datasets. On the other hand, deeper architecture CNN-4 (deeper than CNN-1) requires fewer neurons in FC layers for wider datasets compared to deeper datasets. Deeper CNN models such as CNN-4, CNN-3 have more number of trainable parameters in *Conv* layers. Thus, a deeper dataset is required to learn large parameters of the network. In contrast, a shallow architecture like CNN-1 with 5 *Conv* layers has fewer parameters for which a wider dataset is better suited to train the model.

4.3. Deeper vs. shallower architectures, which are better and when?

Bansal et al. [18] have reported that the deeper architectures are preferred over shallow architectures while training the CNN models with deeper datasets. Whereas, for the wider datasets, the shallow architectures perform better compared to the deeper architectures. However, this observation is specific to face recognition problem as reported in [18]. In this paper, we made a rigorous study to generalize this finding by conducting extensive experiments on different modalities of datasets. For example, CIFAR-10, CIFAR-100, and Tiny ImageNet datasets have the natural images and the CRCHistoPhenotypes dataset has the medical images. The results obtained through these experiments clearly indicate that the deeper architectures are always preferred over shallow architectures to train the CNN model using deeper datasets. In contrast, for the wider datasets, the shallow architectures perform better than the deeper CNN models.

From Table 4, we can observe that training deeper architectures CNN-2 and CNN-3 with deeper dataset produce 92.29% and 92.22% validation accuracies for the CIFAR-10 dataset and 84.89% and 84.94% for the CRCHistoPhenotypes dataset. In contrast, we obtained 90.77% and 82.53% validation accuracies,

Table 6
Comparison of the performance of the proposed method results with the state-of-the-art for CIFAR-10, CIFAR-100, Tiny ImageNet, and CRCHistoPhenoTypes datasets.

S.No.	Dataset	Existing Method		Ours	
		Model	Validation Accuracy	Model	Validation Accuracy
1	CIFAR-10	AlexNet CIFAR-VGG WRN	90.59 91.78 92.55	CNN-4 ($4096 \times 64 \times 10$)	93.37
2	CIFAR-100	AlexNet CIFAR-VGG WRN	66.28 64.56 68.52	CNN-1 ($4096 \times 4096 \times 2048 \times 100$)	69.21
3	Tiny ImageNet	AlexNet CIFAR-VGG WRN	45.25 39.91 48.95	CNN-1 ($4096 \times 4096 \times 1024 \times 200$)	50.1
4	CRCHistoPhenotypes	AlexNet CIFAR-VGG WRN	82.15 84.25 81.55	CNN-3 (128×4)	84.94

when the shallow architecture CNN-1 is trained with CIFAR-10 and CRCHistoPhenotypes datasets, respectively. On the other hand, for the wider datasets such as CIFAR-100 and Tiny ImageNet, better performance is obtained using the shallow architecture (CNN-1). From Table 4, we can observe that the CNN-1 gives a validation accuracy of 69.21% for CIFAR-100 and 50.1% for Tiny ImageNet dataset. Whereas, the CNN-1 model performs relatively poor for deeper datasets.

This observation is very much useful while choosing a CNN architecture to train the model for a given dataset. The generalization of this finding intuitively makes sense because the deeper/shallow architectures have a more/less number of trainable parameters, in a typical CNN model which require more/less number of images per subject (class) for the training. We have also compared the obtained results with the state-of-the-art methods in Table 6. From Table 6 we can observe that the FC layers play vital role to obtain better results.

5. Conclusion

In this paper, we have analyzed the effect of certain decisions in terms of the FC layers of CNN for image classification. Careful selection of these decisions not only improves the performance of the CNN models but also reduces the time required to choose among different architectures such as deeper and shallow. This paper is concluding the following guidelines that can be adopted while designing the deep/shallow convolutional neural networks to obtain better performance.

- In order to obtain better performance, the shallow CNNs require more nodes in FC layers. On the other hand, deeper CNNs need less number of neurons in FC layers irrespective of type of the dataset.
- The shallow CNNs require a large number of neurons in FC layers as well as more number of FC layers for *wider datasets* compared to *deeper datasets* and vice-versa.
- Deeper CNNs perform better than shallow models over *deeper datasets*. In contrast, shallow architectures perform better than deeper architectures for *wider datasets*. These observations can help the deep learning community while making a decision about the choice of deep/shallow CNN architectures.

Declaration of Competing Interest

- Indian Institute of Information Technology, Sri City, A.P., India
- Indian Institute of Information Technology, Allahabad, U.P., India

Acknowledgment

This work is supported in part by the Science and Engineering Research Board (SERB), Govt. of India, Grant No. ECR/2017/000082. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GeForce Titan XP GPU.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.
[2] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[3] B. Hariharan, P. Arbelaez, R. Girshick, J. Malik, Object instance segmentation and fine-grained localization using hypercolumns, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 627–639.
[4] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[5] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Proceedings of the European Conference on Computer Vision, Springer, 2014, pp. 818–833.
[6] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[8] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2009, pp. 248–255.
[10] S.H.S. Basha, S. Ghosh, K.K. Babu, S.R. Dubey, V. Pulabaigari, S. Mukherjee, Rccnet: An efficient convolutional neural network for histological routine colon cancer nuclei classification, in: Proceedings of the 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), IEEE, 2018, pp. 1222–1227.
[11] Q. Xu, M. Zhang, Z. Gu, G. Pan, Overfitting remedy by sparsifying regularization on fully-connected layers of CNNs, Neurocomputing 328 (2019) 69–74.
[12] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Syst. 2 (4) (1989) 303–314.
[13] O. Delalleau, Y. Bengio, Shallow vs. deep sum-product networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2011, pp. 666–674.
[14] Z. Lu, H. Pu, F. Wang, Z. Hu, L. Wang, The expressive power of neural networks: A view from the width, in: Proceedings of the Advances in Neural Information Processing Systems, 2017, pp. 6231–6239.
[15] H.N. Mhaskar, T. Poggio, Deep vs. shallow networks: an approximation theory perspective, Anal. Appl. 14 (06) (2016) 829–848.
[16] H. Mhaskar, Q. Liao, T. Poggio. Learning functions: when is deep better than shallow. arXiv preprint arXiv:1603.00988 (2016).
[17] G.F. Montufar, R. Pascanu, K. Cho, Y. Bengio, On the number of linear regions of deep neural networks, in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 2924–2932.
[18] A. Bansal, C. Castillo, R. Ranjan, R. Chellappa, The dos and donts for CNN-based face verification, in: Proceedings of the IEEE International Conference on Computer Vision Workshop (ICCVW), IEEE, 2017, pp. 2545–2554.
[19] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Technical Report, Citeseer, 2009.
[20] J. Wu, Q. Zhang, G. Xu, Tiny Imagenet Challenge, Stanford University, 2017. cs231n
[21] K. Sirinukunwattana, S.E.A. Raza, Y.-W. Tsang, D.R.J. Snead, I.A. Cree, N.M. Rajpoot, Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images, IEEE Trans Med Imaging 35 (5) (2016) 1196–1206.
[22] J. Ba, R. Caruana, Do deep nets really need to be deep? in: Proceedings of the Advances in Neural Information Processing Systems, 2014, pp. 2654–2662.
[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (2015) 211–252.
[24] S. Liu, W. Deng, Very deep convolutional neural network based image classification using small training sample size, in: Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition (ACPR), IEEE, 2015, pp. 730–734.
[25] S. Zagoruyko, N. Komodakis. Wide residual networks. arXiv preprint arXiv:1605.07146 (2016).
[26] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Proceedings of the International Conference on Machine Learning, 2015, pp. 448–456.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

S. H. Shabbeer Basha received the M.Tech. degree from Jawaharlal Nehru Technological Anantapur College of Engineering Pulivendula (JNTUACEP). He is currently pursuing Ph.D. in the area of Machine Learning and Computer Vision from Indian Institute of Information Technology (IIIT) SriCity since January 2017. His research interest includes Machine Learning, Computer Vision, and AutoML.

Shiv Ram Dubey has been with the Indian Institute of Information Technology (IIIT) Sri City since June 2016, where he is currently the Assistant Professor of Computer Science and Engineering. He received the Ph.D. degree from Indian Institute of Information Technology, Allahabad (IIIT Allahabad) in 2016. Before that, from August 2012-Feb 2013, he was a Project Officer in the Computer Science and Engineering Department at Indian Institute of Technology, Madras (IIT Madras). He was a recipient of several awards including the Best PhD Award in PhD Symposium, IEEECICT2017 at IIITM Gwalior, Early Career Research Award from SERB, Govt. of India and NVIDIA GPU Grant Award Twice from NVIDIA. Recently, he received the research grant from DST/GITA for Indo-Taiwan joint research project. His research interest includes Computer Vision, Deep Learning, Image Processing, Image Feature Description, Image Matching, Content Based Image Retrieval, Medical Image Analysis and Biometrics.

Viswanath Pulabaigari has received his Ph.D. from Indian Institute of Science (IISc) Bangalore. He has obtained B.Tech from Sri Venkateswara University, Tirupati and M.Tech. from Indian Institute of Technology (IIT) Madras. Currently, he is working as an Associate Professor in the Computer Vision and Machine Learning group at Indian Institute of Information Technology SriCity (IIIT SriCity). He has written several peer-reviewed research papers (in reputed journals and conferences). His research area includes Machine Learning, Computer Vision, Pattern Recognition, and Data Mining.

Snehasis Mukherjee has obtained his Ph.D in Computer Science from the Indian Statistical Institute in 2012. Before doctoral study, he has completed his Bachelors degree in Mathematics from the University of Calcutta and Masters degree in Computer Applications from the Vidyasagar University. He did his Post Doctoral Research works at the National Institute of Standards and Technology (NIST), Gaithersburg, Maryland, USA. Currently he is working as an Assistant Professor in the Computer Vision Group of the Indian Institute of Information Technology SriCity (IIIT SriCity). He has written several peer-reviewed research papers (in reputed journals and conferences). His research area includes Computer Vision, Machine Learning, Image and Video Processing.