Chapter 61
Comparative Study of Regularization Techniques for VGG16, VGG19 and ResNet-50 for Plant Disease Detection

Vaishali Suryawanshi, Sahil Adivarekar, Krish Bajaj, and Reem Badami

1 Introduction

Many complex tasks related to medical diagnosis, plant disease detection, etc. are addressed by complex deep neural networks such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN). Training such models with a huge number of parameters is a challenging task. Due to their complexity the models give good performance on the training set but poor performance on the test set which results in overfitting or poor generalization [1]. The issue of generalization is addressed by regularization techniques. Regularization methods can handle noisy data, scarcity of data and complexity of models [1]. Regularization strategies for deep learning include image data augmentation and explicit regularization techniques such as drop out, weight decay and batch normalization.

Dropout technique proposed by Hinton et al. [1–3], refers to making the activation of the randomly chosen hidden layer neurons zero during the training process. The dropout technique results in a thinner network than the original network and trains in reasonable time [1–3]. The dropout techniques have been widely experimented for classification tasks and have been proven to be effective [1–4]. The variants of dropout such as dropconnect [1, 5], standout [6], curriculum dropout [1, 7], dropmaps [8] have been proposed. Dropconnect [6] is applicable for fully connected networks and instead of activations, weights are dropped. Standout [7] controls the dropout properties of each neuron so that if it does not contribute then it is dropped out. Curriculum dropout is an adaptive regularization method where the number of hidden neurons whose activation is set to zero are increased as the training progresses [7].

V. Suryawanshi (✉) · S. Adivarekar · K. Bajaj · R. Badami
Computer Engineering Department, Thadomal Shahani Engineering College, Mumbai, India
e-mail: vaishali.suryawanshi@thadomal.org

S. Kumar et al. (eds.), *Proceedings of International Conference on Communication and Computational Technologies*, Algorithms for Intelligent Systems,
https://doi.org/10.1007/978-981-99-3485-0_61

This introduces the challenge of utilizing limited amounts of information during the training process. In dropmap [8], for a training batch each feature is kept with the probability of p and omitted with probability of 1-p. It can handle the overfitting issue of large models for small datasets [1, 8].

Batch normalization is the process of standardizing the input to the network architecture to avoid the issue of change of data distribution (internal covariate shift) in the hidden layers [1, 9]. This is done by setting the distribution of each activation to zero mean and unit variance. Batch normalization not only improves generalization of the model but also trains the model faster. Batch normalization weight values are computed for a batch and has a limitation that it depends on the batch size. This limitation is addressed in layer normalization where weight values are computed for each feature of a sample independent of the training examples.

Image data augmentation (IDA) is effective in handling the overfitting issue [2]. IDA artificially enlarges the dataset by including synthetic images generated from existing image distribution. Training the model with a huge dataset results in better generalization [2]. Image data augmentation techniques include traditional techniques, feature space augmentation, meta-learning etc. [1]. Traditional augmentation techniques include rotation, scaling, translation, shear, reflection etc. Traditional methods are easy to implement but not applicable in general to all datasets. Feature space augmentation techniques involve manipulating the feature space for generating synthetic images. Recently the feature space of Generative Adversarial Networks (GANs) network is used for generating synthetic images [8].

Plant disease detection is a challenging task due to each plant having its own unique diseases. Due to the scarcity of the images in each disease category, when complex deep learning models such as VGG16, VGG19, RestNet are trained on such datasets, poor generalization is observed. Regularization techniques are applied while training the model to improve the performance. However, a lot of experimentation and parameter tuning is required to get the appropriate combination of regularization techniques. In this research, we have studied the effect of external regularization methods such as dropout, batch normalization, drop out + batch normalization on plant disease detection. We have also experimented with traditional data augmentation methods where we have not considered any of the external regularization methods. And finally, we have also combined all the regularization methods including drop out, batch normalization and data augmentation.

2 Review of Literature

In this section, we present the review of literature for plant disease detection and Convolutional Neural Networks (CNNs) architectures.

2.1 Plant Disease Detection

Plant disease detection is of great concern in the field of agriculture since it can destroy the crops and affect the yield resulting in losses [10–20]. Precise identification of the disease-causing agents is done using molecular biology-based techniques which are costly, require a lot of resources and domain knowledge. The recent decade has witnessed machine learning and deep learning-based methods to automate plant disease detection for sustainable agriculture.

A review of plant disease detection and classification is presented by Li et al. [17], where the authors discussed the current trends and challenges in plant disease detection. In the recent past, machine learning and deep learning-based methods are applied for plant disease detection [8, 10]. Authors [10] conducted review of various classifiers such as Support Vector Machines (SVM), Artificial Neural Network (ANN), K-Nearest Neighbor and Convolutional Neural Network (CNN) and found that CNN gives better performance CNN's accuracy was based on data from 25 plants with the next best being the SVM classifier trained on a particular disease of Oil Palm only compared to other techniques. Wang et al. proposed an algorithm that combined the deep block attention mechanism and convolution kernel [11]. The authors claim that the propped technique pays more attention to disease detail and is better than the classic baseline method [11]. Delnevo et al. [12] designed a system using deep learning and social IOT for sustainable agriculture. Here the authors used the environmental data collected using the sensors and the plant photos crowdsourced from the users to detect the plant disease. Authors [12] evaluated four different CNN architectures DenseNet121, MobileNet, MobileNetV2, NASNetMobile using accuracy and F1-score and found MobileNetV2 gives better performance, the authors arrived at the conclusion that using compression methods like Filter Pruning and Quantization provides a better tradeoff between the overall size and accuracy of the model. Saleem et al. [13] used various deep learning architectures for horticulture plants and found highest mean average precision with and without application of data augmentation. Authors claim that the proposed modified region-based fully connected network (Modified RFCN) was able to detect multiple diseases in one organ, diseases in different plant organs and identify diseases of different crops [13], the Authors used an RFCN model which gave an acceptable accuracy on default settings but after optimization (k-fold cross validation) it improved considerably. Authors [14] applied EfficeintNetV2 and their variations for Cardamom plant disease detection and found the EfficientNetV2-L model gives better performance compared to CNN, EfficientNet, EfficientNet-S and EfficientNet-M. Zeng et al. [8] applied a DCGAN-based data augmentation method with the InceptionV3 model for training and found significant improvement in detecting severity of citrus disease. Lie et al. [15] used a machine learning-based approach for early prediction of blister blight disease for tea plants to take preventive action. Chen et al. [16] developed a lightweight Inception model for rice plant disease detection and noted improvement in the performance. Authors [17] exploited the disease lifecycle and environmental conditions such as temperature, humidity and rainfall for the prediction of disease. Zhao et al. [18], the Authors worked on solving an industry wide issue of scarce data, thus GAN was employed to fix this with DoubleGAN outperforming by generating clearer and more usable images. Laxmi and Savarimuthu [19] presented a transfer learning-based optimized EfficientNet deep learning framework for plant leaves disease detection and noted high detection accuracy for inter and intra class variations, small infected regions and diseased leaves.

2.2 Convolutional Neural Network (CNN)

In this section, we present the theoretical background of the general architecture of CNN followed by the architectures of ResNet-50, VGG16 and VGG19.

General Architecture of CNN

CNNs are the most successful deep learning model in the domain of computer vision. The design of CNN is motivated by the visual cortex inside the brain. They are developed in order to mimic human behavior. CNN model architectures consist of many convolution blocks followed by a densely connected layer. A convolution block consists of a convolutional layer followed by pooling [20, 21]. When a 2D or 3D convolution filter is applied to an input, it produces a feature map. Pooling layer further reduces the dimensions of the feature maps (which summarizes the features in a certain area) [22] and the models learn fewer parameters resulting in saving processing time.

VGG Net Architecture

Visual Geometry Group (VGG) Net CNN architectures have depth ranging from 11 to 19 and were applied by Simonyan and Zisserman [23], for their Large-Scale Image Recognition ImageNet 2014 Challenge. The VGG model uses small 3×3 filters throughout the network with a stride of 1 pixel. The effect produced by 3×3 filters uniformly distributed makes the VGG network different. The effect produced by two consecutive 3×3 filters is equivalent to that produced by a 5×5 receptive filter. Similarly, three 3×3 filters make an equivalent for a receptive filter of 7×7. This way of combination of multiples of 3×3 filters can be used for a receptive area of larger size. Apart from producing an equivalent effect, the benefit of using multiple 3×3 filters is that, in addition to providing three convolution layers, there are also three non-linear activation layers, instead of one in case of 5×5 or 7×7. This makes the decision function more discriminative and would impart the ability of the network to connect faster. The VGG models have a total number of parameters in the range of 133–144 million [23].

VGG16 Architecture

VGG16 [23, 24] model consists of 13 convolutional layers, 5 max pooling layers, 3 dense layers, with a kernel size of 3X3. First 2 convolutional layers use 64 filters, next 2 with 128 filters, next 3 layers with 256 filters, next 2 blocks with 3 convolutional layers each with 512 filters followed by three dense layers, two layers with 4096 nodes and last dense layer with 1000 nodes**.**

VGG19 Architecture

The VGG19 [23, 25, 26] consists of 16 convolutional layers, 5 max pooling layers, 3 dense layers, with a kernel size of 3 × 3. First 2 convolutional layers with 64 filters, next 2 with 128 filters, next 4 with 256 filters, next 2 blocks with 4 convolutional layers each with 512 filters followed by three dense layers, two layers with 4096 nodes and last dense layer with 1000 nodes.

ResNet-50 Architecture

Researchers discovered that adding additional layers to a neural network actually makes the error rate higher; this phenomenon is known as vanishing/exploding gradient, and it led to the development of Residual Networks [27]. A technique known as skip connections connect layer activations to following layers, skipping over some intermediate levels. This creates a leftover block called the residual block as shown in Fig. 1. To build ResNets, these leftover blocks are stacked. Instead of having layers learn the underlying mapping, ResNet lets the network fit the residual mapping. The skip link in ResNet has the advantage that regularization will skip any layer that has a negative impact on architecture performance. As a result, disappearing or increasing gradients are not a problem while training an exceptionally deep neural network. The layers in the ResNet-50 architecture are described in Table 1.

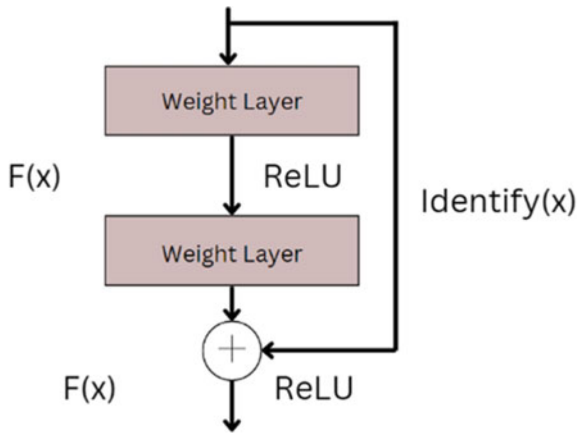

Fig. 1 Residual learning block

Table 1 ResNet-50 architecture

Layer name	ResNet-50
Conv1	7×7, 64, stride = 2 3×3 max pool, stride = 2
Conv2	3 layers each having [(1×1, 64), (3×3, 64), (1×1, 256)]
Conv3	4 layers each having [(1×1, 128), (3×3, 128), (1×1, 512)]
Conv4	6 layers each having [(1×1, 256), (3×3, 256), (1×1, 1024)]
Conv5	3 layers each having [(1×1, 512), (3×3, 512), (1×1, 2048)]
	Average pooling, fully connected dense layer with 1000 nodes, Softmax function at last layer

3 Experimental Setup

In this section, we describe the experimental setup applied in the research. We have carried out the investigation on the Plant Village dataset using VGG16, VGG19 and ResNet-50. The experiments were carried out at Google Colaboratory (Google Colab.) using a GPU processor environment. All the models are trained with Adam optimizer with a learning rate of 0.001 and sparse categorical cross entropy as the loss function. We have used pretraining using ImageNet weights.

3.1 Plant Village Dataset

The PlantVillage dataset consists of 14 kinds of fruits and as described in Table 2 [28] with each plant having their own unique diseases, bringing the total number of classes to 38. This dataset contains 54,309 RGB plant leaf images, which is carefully split into 2 sets, viz., training set and testing set, in the ratio of 80:20. The training set contains 43,429 images and the test set contains 10,876 images.

3.2 External Regularization Techniques

External regularization techniques are applied to address the issue of overfitting that results when a complex deep neural network model is trained with smaller dataset. In this study, we have used dropout, batch normalization and their combinations as explicit regularization techniques. We have chosen dropout rate = 0.4 and for batch normalization momentum rate = 0.8.

Table 2 Plant village dataset description

Plants	Training images
Apple	Healthy [1645 images], Apple Scab [630 images], Black Rot [621 images] and Cedar Apple Rust [275 images]
Blueberry	Healthy [1502 images]
Cherry	Healthy [854 images] and Powdery Mildew [1052 images]
Corn/ Maize	Healthy [1162 images], Cercospora Leaf Spot Gray Leaf Spot [513 images], Common Rust [1192 images] and Northern Leaf Blight [985 images]
Grape	Healthy [423 images], Black Rot [1180 images], Esca (Black Measles) [1383 images] and Leaf Blight (Isariopsis Leaf Spot) [1076 images]
Orange	Haunglongbing (Citrus Greening) [5507 images]
Peach	Healthy [360 images] and Bacterial Spot [2297 images]
Pepper Bell	Healthy [1478 images] and Bacterial Spot [997 images]
Potato	Healthy [152 images], Early Blight [1000 images] and Late Blight [1000 images]
Raspberry	Healthy [371 images]
Soybean	Healthy [5090 images]
Squash	Powdery Mildew [1835 images]
Strawberry	Healthy [456 images] and Leaf Scorch [1109 images]
Tomato	Healthy [1591 images], Bacterial Spot [2127 images], Early Blight [1000 images], Late Blight [1909 images], Leaf Mold [952 images], Septoria Leaf Spot [1771 images], Spider Mites Two Spotted Spider Mite [1676 images], Target Spot [1404 images], Mosaic Virus [373 images], Yellow Leaf Curl Virus [5357 images]

3.3 Data Augmentation

In this experiment we have selected following set of transformations for the data augmentation technique with zoom = 0.2, rotation = 30, width_shift = 0.2, height_shift = 0.2, shear = 0.2. While implementing data augmentation technique alone, we have not applied any other regularization method.

4 Results and Discussion

In this section, we present the results of the RestNet-50, VGG16 and VGG19. To evaluate the performance of the plant disease detection we have applied confusion matrix-based metrics such as balanced accuracy, weighted precision, weighted recall and weighted F1-measure [29, 30].

In Table 3, results of VGG16 model are presented. Results of drop out technique are the best for VGG16 model. Batch normalization gives slightly poor performance than the No regularization method. The performance of batch normalization is however better than data augmentation. With the combination of drop out and batch normalization the performance is better than batch normalization alone but not as good as drop out.

Results of the experiments carried on VGG19 are given in Table 4. Here, similar observations are noted. Drop out gives the best result and data augmentation gives the poorest performance. The combination dropout and batch normalization is better than applying batch normalization alone.

Results of the experiments carried on ResNet-50 are given in Table 5. A combination of drop out and batch normalization gave results comparable to that of VGG16 and VGG19. However, the results of other regularization techniques on ResNet-50 model are average.

The comparison of the VGG16, VGG19 and RestNet-50 are presented in Table 6 using balanced accuracy and normalized MCC scores.

Table 3 Results on VGG16

VGG16	BA	Precision	Recall	F1-score	Norm MCC
No Reg	0.8246	0.8227	0.8246	0.8224	0.9123
Drop out	0.8528	0.8528	0.8528	0.8512	0.9264
Batch norm	0.8213	0.8223	0.8213	0.8197	0.9106
DO + BN	0.8391	0.8402	0.8391	0.8376	0.9195
DA	0.7067	0.7374	0.7067	0.7087	0.8533
DO + BN + DA	0.0116	0.0001	0.0116	0.0003	0.5058

Table 4 Results on VGG19

VGG19	BA	Precision	Recall	F1-score	Norm MCC
No Reg	0.8228	0.8191	0.8228	0.8184	0.9114
Drop out	0.8496	0.8493	0.8496	0.8486	0.9248
Batch norm	0.8103	0.8101	0.8103	0.8082	0.9051
DO + BN	0.8330	0.8363	0.8330	0.8324	0.9165
DA	0.6483	0.6787	0.6483	0.6403	0.8241
DO + BN + DA	0.0116	**0.0116**	0.0116	**0.0116**	0.5058

Table 5 Results on ResNet-50

ResNet-50	BA	Precision	Recall	F1-score	Norm MCC
No Reg	0.6270	0.6156	0.6270	0.6027	0.8134
Drop out	0.5864	0.5716	0.5864	0.5466	0.7932
Batch norm	0.6787	0.6867	0.6787	0.6765	0.8393
DO + BN	0.8330	0.8363	0.8330	0.8324	0.9165
DA	0.2129	0.1041	0.2129	0.1207	0.6064
DO + BN + DA	0.2177	0.2177	0.2177	0.2177	0.6089

Table 6 Comparison of all models

Model	BA	Norm MCC
VGG16	0.8528 (drop out)	0.9264 (drop out)
VGG19	0.8496 (drop out)	0.9248 (drop out)
ResNet-50	0.8330 (drop out + batch norm.)	0.8393 (drop out + batch norm.)

We find that best performance for VGG16 and VGG19 models is obtained with drop out technique. The improved performance could be attributed to the fact that training with a deep learning model is equivalent to obtaining average performance of an ensemble of models with different parameters. For the ResNet-50 model, the performance of dropout technique is average. Since the ResNet-50 model has just 25 M parameters, with dropout the model further gets simplified. Due to this, the ResNet-50 model is not able to learn a sufficient number of features and results in poor test performance. However, when the ResNet-50 model is trained with a combination of drop out and batch normalization, the significant improvement in performance is obtained.

Overall, the performance of the VGG16 model is better compared to other two models. We also observed that the performance of all the models for selected combinations of standard transformations techniques for data augmentation is poor. Similar observation is noted for all the three techniques combined together.

5 Conclusion and Future Work

In this research, we investigated the effectiveness of regularization techniques for VGG16, VGG19 and ResNet-50 models for plant disease detection. Regularization techniques such as drop out, batch normalization, data augmentation and their combinations are experimented on plant disease detection application. Analysis of the results show that the drop out technique is better than all other techniques for both VGG16 and VGG19 models whereas the performance of ResNet-50 improved with the combination of drop out and batch normalization. Further, analysis of the results shows that the VGG16 model with dropout gives best performance compared to other models. The performance of VGG19 is better than ResNet-50. We also noted that the traditional data augmentation techniques gave just average performance. When drop out, batch normalization and data augmentation was applied together, all the three models showed poor performance.

Since the dropout technique has shown improvement in the generalization of VGG16 and VGG19, we would like to carry out similar studies for various dropout techniques and also train different and latest CNN models such as Efficient Net, Wide ResNet.

References

1. Tian Y, Zhang Y (2022) A comprehensive survey on regularization strategies in Machine Learning. Inf Fusion 80:146–166
2. Shorten C, Khoshgoftaar TM (2019) A survey on image data augmentation for deep learning. J Big Data 6:60
3. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv, 1207.0580. http://arxiv.org/abs/1207.0580
4. Deep learning regularization techniques to genomics data
5. Wan L, Zeiler M, Zhang S, Le Cun Y, Fergus R (2013) Regularization of neural networks using drop connect In: International conference on machine learning, PMLR, pp 1058–1066
6. Ba J, Frey B (2013) Adaptive dropout for training deep neural networks. Adv Neural Inf Process Syst 26:3084–3092
7. Morerio P, Cavazza J, Volpi R, Vidal R, Murino V (2017) Curriculum dropout. In: 2017 IEEE International Conference on Computer Vision (ICCV)
8. Zeng Q, Ma X, Cheng B, Zhou E, Pang W (2020) Gans-based data augmentation for citrus disease severity detection using deep learning. IEEE Access 8:172882–172891
9. Annamalai L, Thakur CS (2022) Theoretical insight into batch normalization: data dependant auto-tuning of regularization rate. Preprint: Available: arXiv:2209.07587v1
10. Shruthi U, Nagaveni V, Raghavendra BK (2019) A review on machine learning classification techniques for Plant Disease Detection. In: 2019 5th international conference on advanced computing & communication systems (ICACCS)
11. Wang J, Yu L, Yang J, Dong H (2021) A novel end-to-end object detection algorithm applied to plant disease detection. Inf Basel 12(11):474. https://doi.org/10.3390/info12110474
12. Delnevo G, Girau R, Ceccarini C, Prandi C (2022) A deep learning and social IOT approach for plants disease prediction toward a sustainable agriculture. IEEE Internet Things J 9(10):7243–7250
13. Saleem MH, Potgieter J, Arif KM (2022) A performance-optimized deep learning-based plant disease detection approach for horticultural crops of New Zealand. IEEE Access 10:89798–89822
14. Sunil CK, Jaidhar CD, Patil N (2022) Cardamom plant disease detection approach using EFFICIENTNETV2. IEEE Access 10:789–804
15. Liu Z, Bashir RN, Iqbal S, Shahid MM, Tausif M, Umer Q (2022) Internet of things (IOT) and machine learning model of plant disease prediction–blister blight for tea plant. IEEE Access 10:44934–44944
16. Chen J, Chen W, Zeb A, Yang S, Zhang D (2022) Lightweight Inception Networks for the recognition and detection of rice plant diseases. IEEE Sens J 22(14):14628–14638
17. Li L, Zhang S, Wang B (2021) Plant disease detection and classification by deep learning—a review. IEEE Access 9:56683–56698
18. Zhao Y, Chen Z, Gao X, Song W, Xiong Q, Hu J, Zhang Z (2021) Plant disease detection using generated leaves based on doublegan. IEEE/ACM Trans Comput Biol Bioinf:1–1
19. Lakshmi RK, Savarimuthu N (2022) PLDD—a deep learning-based plant leaf disease detection. IEEE Consum Electron Mag 11(3):44–49
20. Tummala M (2019) Image classification using convolutional neural networks. Int J Sci Res Publ (IJSRP) 9(8)
21. Gua J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang L, Wang G, Cai J, Chen T (2017) Recent advances in convolutional neural networks
22. Ahmad I, Hamid M, Yousaf S, Shah ST, Ahmad MO (2020) Optimizing pretrained convolutional neural networks for tomato leaf disease detection. Complexity 2020:1–6
23. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556
24. Jiang Z-P, Liu Y-Y, Shao Z-E, Huang K-W (2021) An improved VGG16 model for pneumonia image classification. Appl Sci 11(23):11185

25. Xiao J, Wang J, Cao S, Li B (2020) Application of a novel and improved VGG19 network in the detection of workers wearing masks. J Phys Conf Ser 1518(1):012041
26. Sudha V, Ganeshbabu DTR (2020) A convolutional neural network classifier VGG19 architecture for lesion detection and grading in diabetic retinopathy based on deep learning. Comput Mater Continua 66(1):827–842
27. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR)
28. APJ, Gopal G (2019) Data for: identification of plant leaf diseases using a 9-layer deep convolutional neural network. Mendeley Data, 18 Apr 2019. Available: https://data.mendeley.com/datasets/tywbtsjrjv/1
29. Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21
30. Chicco D, Tötsch N, Jurman G (2021) The Matthews Correlation Coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. BioData Mining 14