International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 7 ǁ July 2022 ǁ PP. 784-793

A Comparative Study on Regularization Techniques in Convolutional Neural Networks

A. Agnes Lydia[*1], Sheela Chandrasekar[2]
[*1] *School of Computer Science and IT, Jain (Deemed-to-be) University, Bengaluru, India*
[2]*School of Computer Science and IT, Jain (Deemed-to-be) University, Bengaluru, India*
Corresponding Author:A. Agnes Lydia

Abstract
In today's scenario, information sharing is majorly based on images. Information in the form of tweets, messages or social media updates, are all in the form of images. With such voluminous amount of data being piled up across the cloud, there arises a need to develop a model to accuratelyanalyse this data. Convolutional Neural Networks are the state-of-art network architectures to perform several Computer Vision tasks involving raw images as input data. As the domain grows over the ages, the number of variants in Convolutional Neural Networks have outgrown in number, with each variant satisfying only a specific need. On the bigger picture, the area where each Convolutional Neural Network can be improvised is still not widely explored. The phenomenon in which a model with good performance on training data and fails to performwell on a test data is termed as Over-fitting. This article focuses on detailed analysis of various techniques that reduces overfitting in a Convolutional Neural Network. These techniques are implemented on a benchmarking dataset Caltech-256, which is an imbalanced datasets. For experimental purpose, these regularization techniques are applied to ResNet-50, one of theprominent variants of Convolutional Neural Networks.

Keywords:*Image Classification, Transformations, Convolutional Neural Network, ResNet, Overfitting, RegularizationTechniques, Dropout, Augmentation, Caltech-256.*

Date of Submission: 10-07-2022 Date of acceptance: 26-07-2022

I. INTRODUCTION

Neural networks are one of the prominent models in Machine Learning that has the ability to learn from the data fed into it, by analysing the distribution of every data point in a vector space [1]. An efficient neural network is expected to make accurate predictions not only on the data that it has already seen, but also predict the category of data which it has never seen before. This nature of a neural networkis referred to as *Generalization* [2] [3]. A Network thatlacks Generalizability, suffers from *Over-fitting* [4] [5]. Theoccurrence of *Over-fitting* can be controlled by exploring the number of parameters involved in the model's architecturewhile training with different kinds of data. The term *Generalization* is directly proportional to the number of parameters in the network, which directly depends on the number ofneurons and its weights in each layer.

$$G \propto P \qquad (1)$$

where, *G* represents *Generalization* and *P* represents *number of Parameters*. A complex network architecture consistingof more number of parameters is also known as a *Non-Linear model* [6]. A complex dataset requires a relativelycomplex mathematical function to understand the data and make predictions. This is where *Artificial Intelligence* (AI) comes into picture, in which the network decides a suitable function to learn the features in the data without any human intervention [7]. The objective is to increase or decrease the parameters in the model in relevance to the complexity of the training data, such that the performance of the model isbetter in both the training as well as the testing data.

II. RELATED WORKS

Smirnov, Evgeny A. et al., in 2014, trained a Convolutional Neural Network with ImageNet Large Scale Visual Recognition Challenge 2013 dataset and compared the results of two regularizers, namely DropConnect and Dropout [8] [9] [10]. The results were obtained in such a manner that Dropout technique outperformed the DropConnect regularization technique. Xie, Saining, et al., in 2015, built a neural network to identify objects using data that was highly grained. This is a complex task for the model to learn the patterns

from a grained image, as every patch of image might look similar [11]. To increase the efficiency of learning, the technique of Hyper-class Data Augmentation was adapted, which increases the number of samples in every class by applying the traditional augmentation techniques namely, rotation, scaling, shifting and cropping.

Though Dropout technique had been widely used to reduce overfitting over many years, an in-depth analysis was not made on it until Park, Sungheon, et al., in 2016, revealed the fact that, Dropout actually adds more noise to the feature map in each layer and increases the robustness to variation of images [12]. And also experimented with a Stochastic Dropout technique, in which the drop ratio varies for every iteration. Ghiasi, Golnaz, et at., in 2018, worked on a technique of structured dropout, that drops an entire block that is contiguous in the convolutional layer, called as DropBlock. DropBlock improves the accuracy of ResNet trained on ImageNet by 1.6% [13] [14]. Labach, Alex, et al., in 2019, made a comprehensive study on several variants of Dropout that have been in existence from 2012 to 2019. Even though several variants have been developed, the Standard Dropout still provides better results for majority of CNNs [15] [16].

The information gathered from these articles are taken into consideration for the experimental evaluation of the Dropout regularization technique trained on an imbalanced image dataset, Caltech-256 [17] [18]. A sample of the Caltech-256 dataset is exhibited in Figure 1.

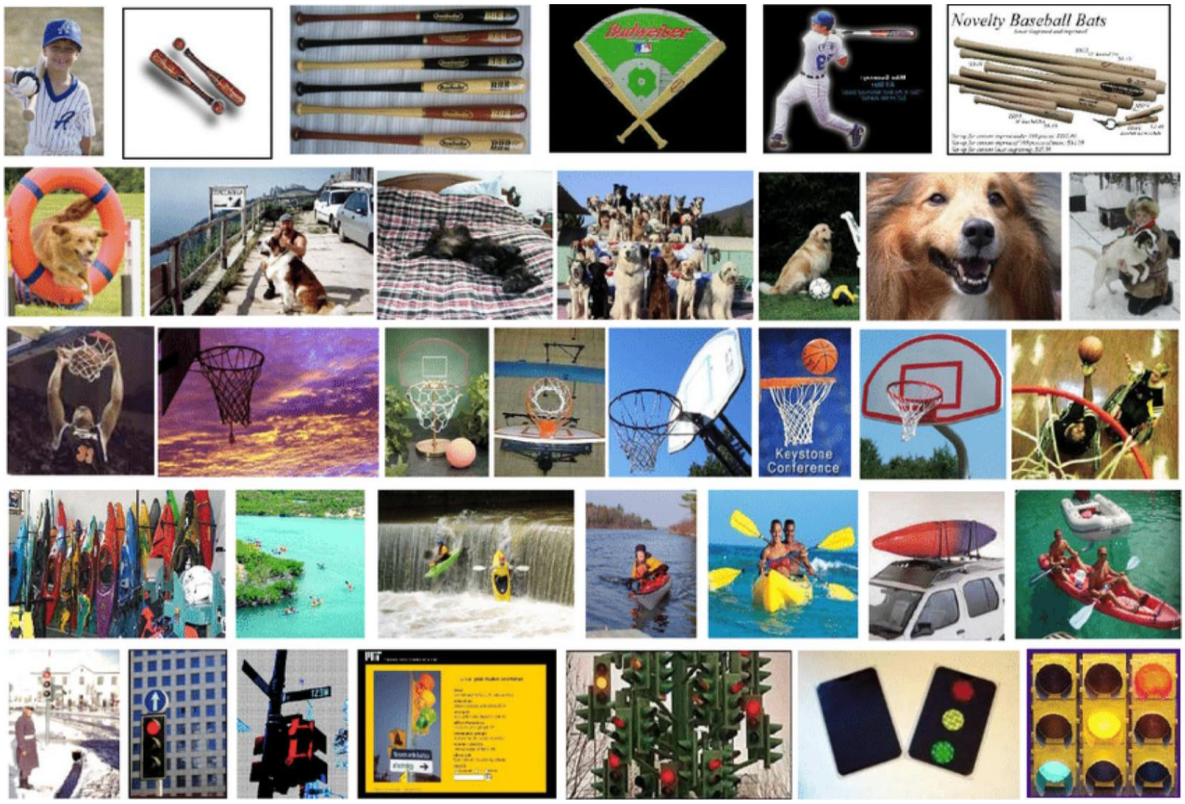

Figure 1: Sample Data in Caltech - 256 Dataset

III. CONVOLUTIONAL NEURAL NETWORK VARIANTS

Convolutional Neural Networks have the ability to handle image data in its original form, namely, raw pixels. These pixels are provided as sequence of numbers to the input layer of the Convolutional Neural Network (CNN). Every CNN layer performs several operations on these pixel values at different levels of abstraction (Figure 2). Convolution Layer, the first operational layer concentrates on one portion of the image at every instance, with high density of pixel values and the prominent patterns in the image are extracted (Feature Extraction) [19]. Feature Extraction is initiated with the help of Kernels or Filters, that are cross multiplied with the pixel values in a sequence, resulting with an output of images with reduced dimension. These images are fed into the next operational layer, Pooling. Similar operation of Feature Extraction is performed in the layer, and the size of the images are further reduced to one half of its original size Dimensionality Reduction [20]. By extracting features with Pooling operation, the model becomes much more efficient to identify the patterns

irrespective of its spatial location and the pattern's orientation in the images. The next operational layer, Fully Connected Layer (FC) converts the reduced image dimension into a single vector which is then fed as input to the Classification layer, the final layer of the network [21]. Based on the nature of the Classifier used to build the network, it performs Binary or Multi-Class Classification.

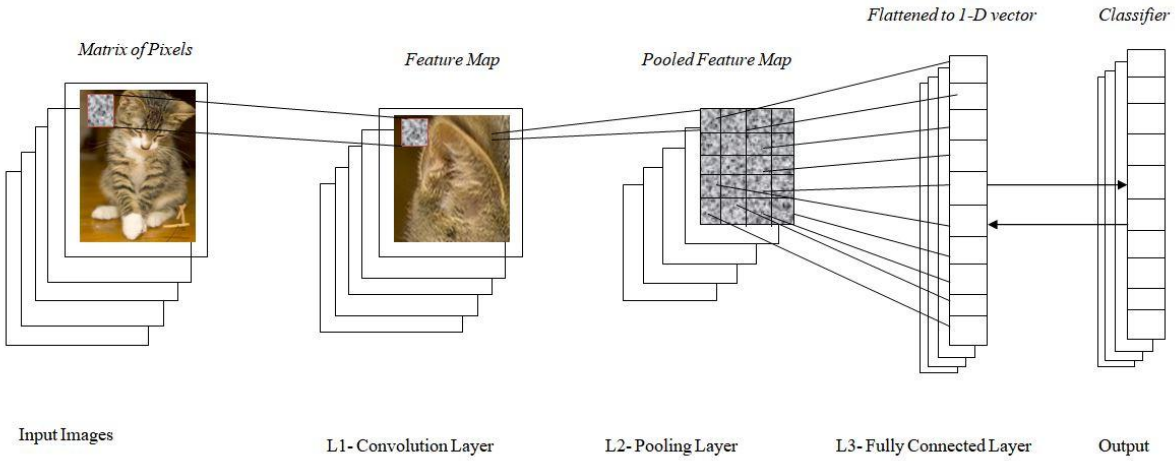

Figure 2 - Architecture of Convolutional Neural Network

This being the general architecture of a Convolutional Neural Network, the operational layers can be increased or decreased in numbers and ordered in any manner as the task in concern requires [22]. Several variants of CNN have been developed, by overcoming the drawbacks in its predecessor models. Analysing all the successful variants of CNN, one of the unique kind of CNN, the ResNet is considered for experimental evaluation in this article.

3.1. ResNet

A general idea that prevails in the field of Deep Learning is that, "The deeper the network, the better the performance". This is was true until it started to fail in VGG networks [23]. Increasing the depth of the network beyond its capability reduced the generalizability of the network. And another obstacle that was faced while building a deeper network is, Vanishing Gradient Descent problem [24]. When the loss is calculated from the gradients at a specific level in the network, while backpropagating and updating the weights of the neuron, the weights of the weaker neurons tend to be decreased to less than 0 and consequently get ignored during multiple iterations.

The architecture of the ResNet is designed in such a manner that it skips a few intermediate neurons while backpropagating which leads to reducing the ratio of degrade in the weights of the neurons[25]. This is represented in Figure 3. By default, the incoming weights 'x' should be passed on to the first layer with function F(x), followed by the next layer with the function F(x)+x. But in ResNet, with the facility to skip layers which does not have any change in the neuron weights in consecutive layers, the weights are passed on to the proceeding layers with a function that causes an upgradation in the weights.

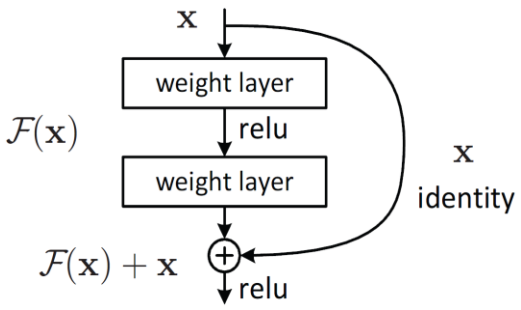

Figure 3 - Residual Network – Structure of a skipped Block

From the literature survey on related existing works, we get an insight that, building Convolution layers with smaller filters makes the network to learn even the minute details in the patterns from the images and takes

less time to train. With this ideology, the regular ResNet model has been tweaked to have filters of size $3 * 3$. And the depth of the ResNet model have been reduced to have only 34 operational layers in comparison to its original model with 50 layers. With this architecture as the base model, some of the important regularization techniques are used to compare the performances of different regularizers in the following sections.

IV. REGULARIZATION TECHNIQUES

Neural Network consists of smaller units called nodes. The number of layers in a network and the number of nodes in every layer constitutes the complexity of the model. The network learns the underlying patterns in the input data by adjusting the weights of the concerned nodes in every layer leading towards the expected outcome.

One of the major hurdles in training a neural network is that, it tends to perform well on training data but performs poorly on test data. To overcome this, Regularization Techniques provide support by penalizing the weights of the nodes in such a manner that it improvises the model's performance [26].

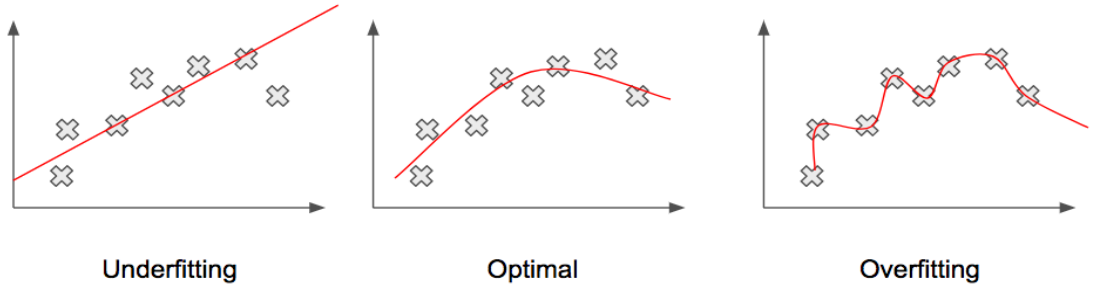

Figure 4 -Finding the Best Fit.

Considering an uneven distribution of data as in Figure 4, finding the best fit without overfitting is the objective of building a generalized model. A diagrammatic representation of the overall process flow is depicted in the Figure 5. A model with over-fitting can be improvised in several ways by using different regularization techniques, and these techniques are analysed in detail as follows.

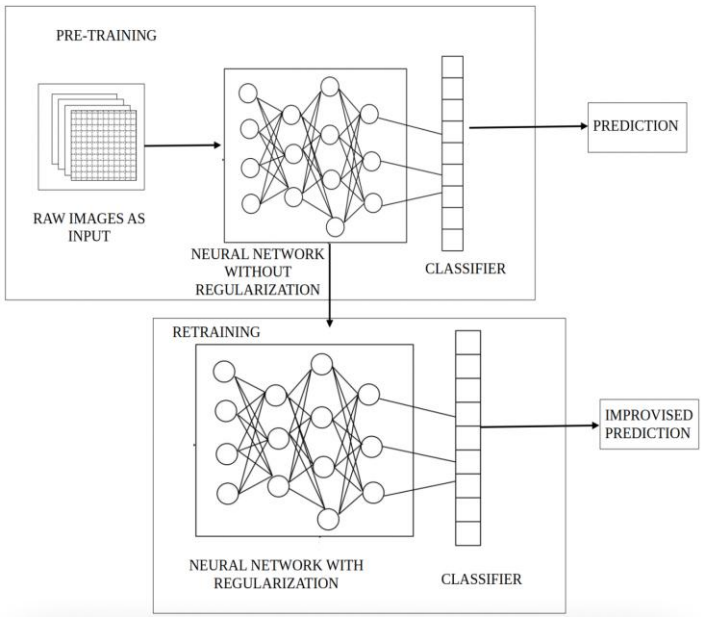

Figure 5 – Process Flow Diagram

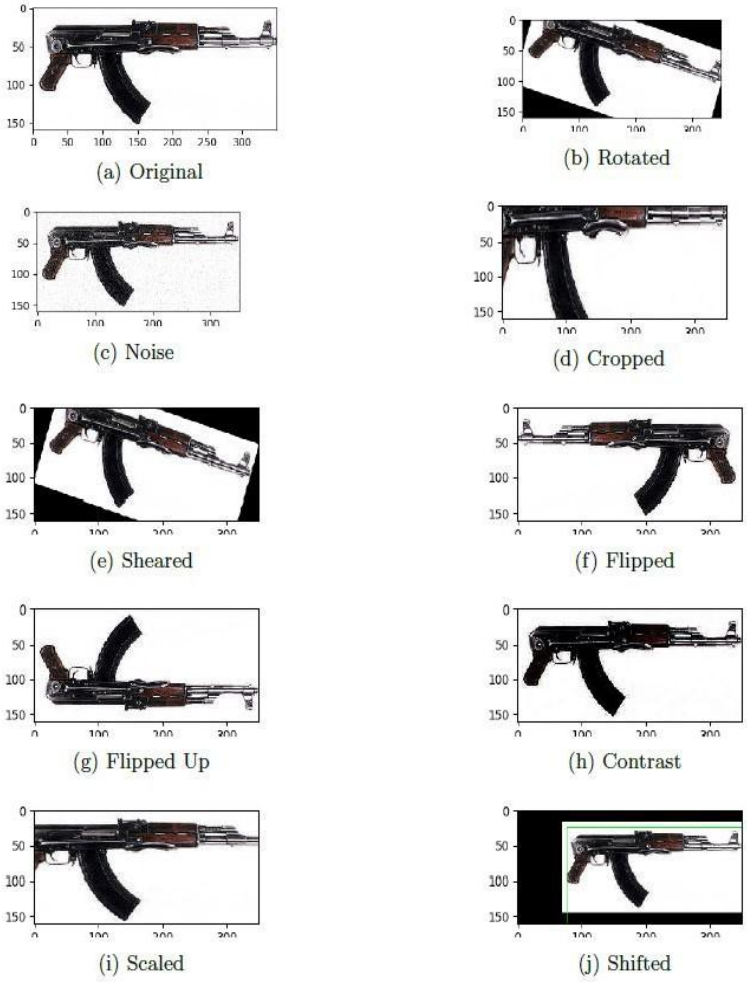

Figure 6 - Data Augmentation Applied on a Sample Caltech-256 data.

4.1. Data Augmentation

The simplest of all regularization techniques is Data Augmentation. Augmentation is the process of increasing the number of training samples by applying transformations like rotating, flipping, scaling, translating/shifting and shearing [27]. A visualization of various transformations applied to a sample image from Caltech-256 dataset is shown in Figure 6.

Data Augmentation enhances the possibility of viewing the objects in focus, in different perspectives and gives the model the ability to learn the patterns better and perform well on new sample set of data.

4.2. L_1 and L_2 Regularization

The goal of the training process is to guide the model towards the direction where the optimum value lies in the vector space of data distribution. The key component that monitors the progress of the training phase is a Regularizer. It is well known that in a supervised learning model, the network fine-tunes its neuron weights by calculating the loss value obtained by comparing the received outcome with the expected outcome. These loss values are obtained using appropriate Cost Functions as in Equation (2).

$$C(f, x, y) = L \qquad (2)$$

Hence every training process is a minimization problem with an objective to minimize this loss value as in Equation (3).

$$\min C(f, x, y) = L \qquad (3)$$

where 'C' refers to the Cost function, 'f' refers to the model, 'x' is the input value, 'y' is the expected outcome and 'L' is the loss value. By adding the component Regularizer, 'R(f)' to this objective function the performance of the learning phase is optimized as in Equation (4).

$$\min C(f, x, y) = L + R(f) \quad (4)$$

The commonly preferred forms of the Regularizers are L_1 or Lasso Regularizer and L_2 or Ridge Regularizer. L_1 Regularizer applies a penalty value and reduces the absolute weights of the neurons while backpropagating. This leads to pruning of unnecessary neurons in the training path, reducing the complexity of the network architecture. L_1 Regularizer can be added to the objective function as in Equation (5).

$$\min C(f, x, y) = L + L_1R(f) \quad (5)$$

where the Regularization variable, '$L_1R(f)$' is calculated as in Equation (6).

$$L_1R(f) = \lambda * \Sigma | w | \quad (6)$$

L_1 Regularizer is suitable to train larger network architectures, and hence pruning of neurons does not impact the performance of the model. But in case of smaller network architectures, the phenomenon of pruning neurons with the application of L_1 Regularizer might lead to diminishing some of the important neurons. This is better handled by the L_2 Regularizer [28]. L_2 Regularizer works similar to L_1 Regularizer while the penalty is applied to the square of the neuron weights as in Equation (8) and the Cost function is calculated as in Equation (7).

$$\min C(f, x, y) = Loss + L_2R(f) \quad (7)$$

$$L_2R(f) = \lambda * \Sigma | w^2 | \quad (8)$$

where, 'λ' refers to the learning rate value and w refers to the summation weights of neurons in the preceding layers.

4.3. Dropout

One of the pitfalls of training larger networks to learn all the features in the data, is the occurrence of Co-Adaptation. When the network is made to parallelly learn the weights of all the neurons, there arises a scenario where the connections with neurons having larger weights are trained well, while the connections having neurons with least weights are trained weakly and further ignored. This phenomenon is known as Co-Adaptation [29]. This could not be resolved with the traditional regularization techniques like, L_1 or L_2.

Consequently, building much deeper and wider networks also did not favour the need of reducing Co-Adaptation. The high predictive capability of some neurons in addition to Co-Adaptation can be regulated with help of Dropout technique. There are several kinds of Dropout techniques depending on the architecture it is incorporated. CNNs are best regulated using the Standard Dropout technique [30]. As the name suggests, to avoid the network to learn the weights of all the neurons, Dropout randomly removes neurons on the path of concern in such a way that it does not affect the efficiency of the model. Removing random neurons also reduces the complexity of the model and for each iteration the dropped out neurons vary as in Figure 7.

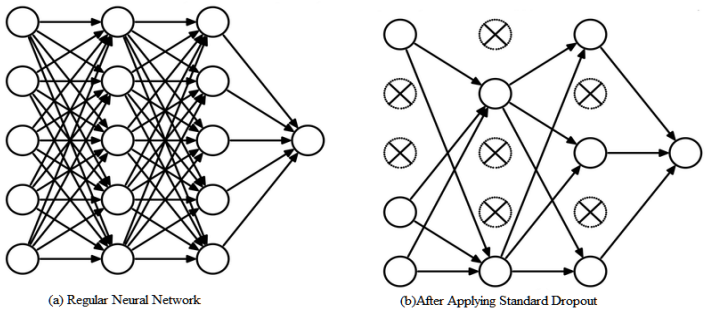

Figure 7 - Dropout Regularizer

V. EXPERIMENTAL ANALYSIS

The techniques discussed in the previous sections of this article are evaluated practically by building a ResNet architecture and trained with imbalanced benchmarking dataset, Caltech-256. Caltech-256 is an improvisation over the Caltech-101 image dataset. The number of classes in Caltech-256 dataset are doubled in comparison to Caltech-101 and the minimum number of samples in every class is increased to 80 images from 30 images as in Caltech-101. Both these datasets are imbalanced datasets with unequal number of samples in each class. This provides a data distribution quite similar to the real-time environment, enabling the network to be trained on a real-time scenario.

The performance is evaluated with a visual perception of learning curve in the form of a graph. In every graph, the X −axis denotes the number of epochs the model is trained, and the Y − axis denotes the Loss values in the decreasing order as well as the Accuracy values in the increasing order with the unit in percentage. Fluctuations in any of these curves denote the presence of Over-fitting. Increasing the number of training samples, increases the learning space of the model to extract the undiscovered patterns. The difference in the performance of the model without augmentation and with augmentation is depicted in Figure 8 and Figure 9.

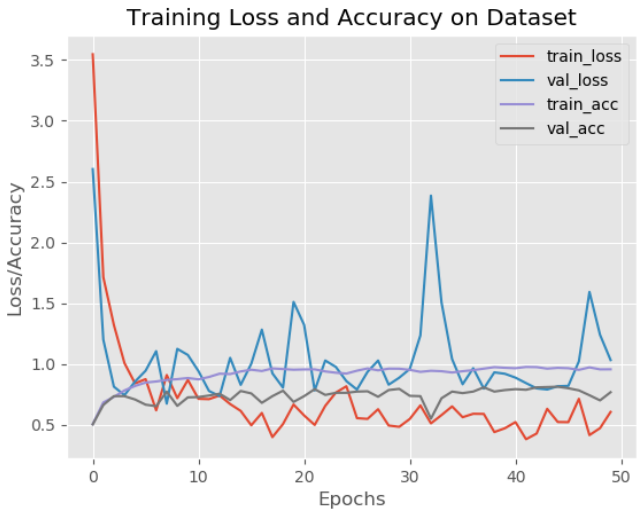

Figure 8 - Accuracy and Loss without Augmentation

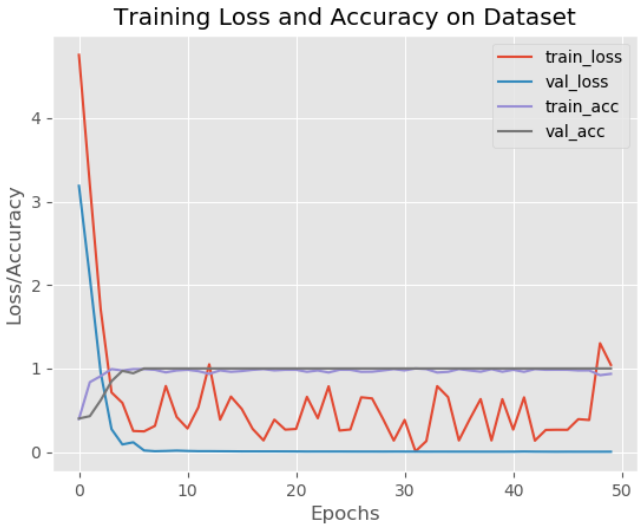

Figure 9 - Accuracy and Loss with Augmentation

Observing the flow of graph in Figure 8 which shows the model's learning progress with unaugmented Data, it is evident that, there is a sudden rise in the graph of validation loss during the epochs 18−21, 30−35 and 45−50, indicating the presence of Over-fitting. Over-fitting is not observed in curves of Training Accuracy, Training Loss and Validation Accuracy. The curves of both the Training (train_acc) and Validation Accuracy (val_acc) is expected to reduce and progress towards 1.0, indicating the model's performance reaching towards 100% accuracy level. While the curves of both the Training (train_loss) and the Validation Loss (val_loss) is expected to reduce and become minimum progressing towards 0.

Observing the graph in Figure 9, model trained with Augmented Data, it is visible that the learning curve is directed towards better accuracy levels and reduced loss. It shows fluctuations in the validation loss during the epochs 18 − 20. This indicates the presence of over-fitting in the validation data. A good model must not have over-fitting in both the training data and in the validation data.

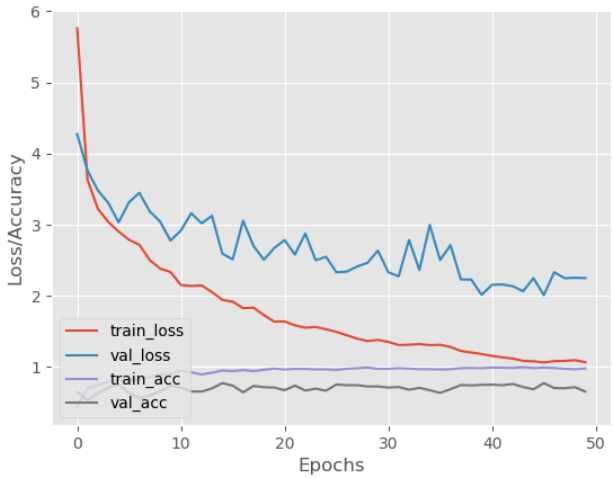

Figure 10 - Accuracy and Loss of L_1 Regularizer.

The performance of the model after incorporating L_1 or Lasso Regularizer is shown in Fig. 10. The learning curve of accuracy seems to be converging smoothly for both the training data and the validation data. The training loss is also greatly reduced without much oscillations. The fluctuations in the training loss have been reduced to a certain extent but the validation loss value is not reduced as much as in using Data Augmentation. Yet the presence of over-fitting in all four curves have been minimized using Data Augmentation.

Although L_2 or Ridge Regularizer handles the gradients in a similar way to L_1 regularizer, as the penalty value is applied to the squared weights of the neurons the performance of the model is further enhanced. This is evident in the graph as in Figure 11. The only adverse case in L_1 regularizer is the poor reduction in the loss of the validation data and this is handled much better by L_2 regularizer, but still it exhibits fluctuations in its validation loss curve.

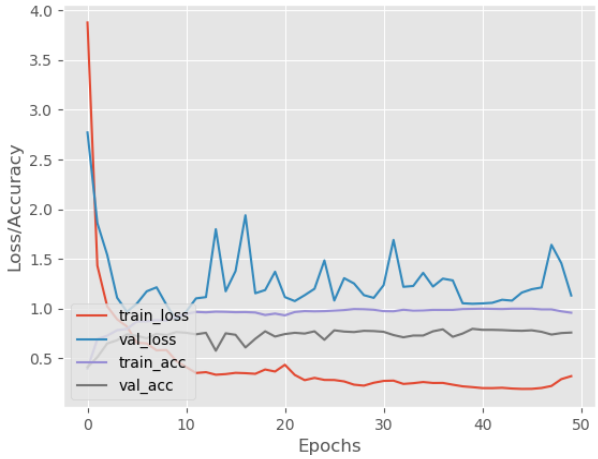

Figure 11 - Accuracy and Loss of L_2 Regularizer

The final regularizer in concern in this article is, Dropout. The Standard Dropout regularizer has been implemented in this model. The general idea that exists, is that, "increasing" the dropout ratio would reduce the overfitting in the model. And the best results are usually obtained with the Dropout ratio at 50%. As a contradiction, since the model in this article is trained on an imbalanced data, better results are obtained by "reducing" the Dropout ratio. The fluctuations of the learning curve were better reduced at a Dropout ratio of 5% and the graph is shown in Figure12.

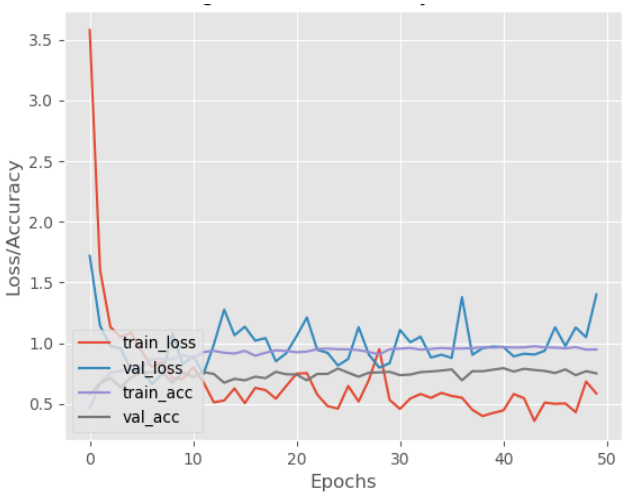

Figure 12 - Accuracy and Loss of Dropout

The values obtained from the comparison made using various kinds of regularizers focussed in this article are tabluated in Table I. It shows that training the ResNet model with the Caltech- 256 dataset, produced the best results with better accuracy levels and better reduced loss values while using the Dropout regularizer. And this improvisation in performance is found in both the training data and in the validation data, leading to a better generalized neural network, i.e., a model that predicts well both in training data as well as in validation data.

TABLE I
COMPARISON OF REGULARIZATION TECHNIQUES: CALTECH-256

Techniques	Train/Val	Accuracy	Loss
L_1	Training	0.9803	1.0698
	Validation	0.6550	2.2513
L_2	Training	0.9584	0.3251
	Validation	0.7600	1.1327
Dropout	Training	0.9433	0.5106
	Validation	0.8700	0.5577

VI. CONCLUSION

This article is a practical analysis inspired by the rapid increase in the number of techniques that handles the problem of overfitting while training a neural network. It provides a brief insight about the need to develop a Generalized model and the research works that are currently being done to satisfy the need. The most promising regularization techniques are discussed in detail and are also experimented with a unique benchmarking dataset, Caltech-256. These results provide an evident visualization of Dropout regularizer, outperforming all the other regularization techniques. This work can also be further extended to be implemented with other Deep Learning Architectures like Recurrent Neural Networks (RNN), Long Short-Term Memory Neural Networks (LSTM), Deep Belief Neural Networks (DBN), etc., as well on various other kinds of image datasets.

REFERENCES

[1]. Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving Deep Convolutional Neural Networks for Image Classification," *IEEE Transactionson Evolutionary Computation*, Vol. 24, no. 2, pp. 394–407, 2019.

[2]. C.-Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing Pooling Functionsin Convolutional Neural Networks: Mixed, Gated, and Tree," in *ArtificialIntelligence and Statistics*, 2016, pp. 464–472.

[3]. J. Yan, M. Avagyan, R. E. Colgan, D. Veske, I. Bartos, J. Wright,Z. Marka, and S. M ´ arka, "Generalized Approach to matched filtering using Neural Networks," *Physical Review D*, Vol. 105, no. 4, p. 043006, 2022.

[4]. B. Wu, Z. Liu, Z. Yuan, G. Sun, and C. Wu, "Reducing Overfitting in Deep Convolutional Neural Networks using Redundancy Regularizer,"in *International Conference on Artificial Neural Networks. Springer*, 2017, pp. 49–55.

[5]. C. Filipi Gonc¸alves dos Santos and J. P. Papa, "Avoiding Overfitting: A Survey on Regularization Methods for Convolutional Neural Networks,"*arXiv e-prints*, pp. ArXiv–2201, 2022.

[6]. M. T. Islam, B. N. K. Siddique, S. Rahman, and T. Jabid, "ImageRecognition with Deep Learning," in 2018 *International Conference onIntelligent Informatics and Biomedical Sciences (ICIIBMS)*, Vol. 3. IEEE, 2018, pp. 106–110.

[7]. Z. Zainuddin and O. Pauline, "Function Approximation using ArtificialNeural Networks," *WSEAS Transactions on Mathematics*, Vol. 7, no. 6, pp. 333–338, 2008.

[8]. E. A. Smirnov, D. M. Timoshenko, and S. N. Andrianov, "Comparisonof Regularization Methods for Imagenet Classification with Deep Convolutional Neural Networks," *Aasri Procedia*, Vol. 6, pp. 89–94, 2014.

[9]. N. D. Nguyen, T. Jin, and D. Wang, "Varmole: A Biologically Dropconnect Deep Neural Network Model for Prioritizing Disease Risk Variants and Genes," *Bioinformatics*, vol. 37, no. 12, pp. 1772–1775, 2021.

[10]. X. Liang, L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang,and T.-Y. Liu, "R-drop: regularized dropout for Neural Networks," *arXivpreprint arXiv:2106.14448*, 2021.

[11]. S. Xie, T. Yang, X. Wang, and Y. Lin, "Hyper-class Augmented and Regularized Deep Learning for fine-grained Image Classification," in*Proceedings of the IEEE Conference on Computer Vision and PatternRecognitio*n, 2015, pp. 2645–2654.

[12]. S. Park and N. Kwak, "Analysis on the Dropout Effect in ConvolutionalNeural Networks," in *Asian Conference on Computer Vision. Springer*, 2016, pp. 189–204.

[13]. G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Dropblock: A RegularizationMethod for Convolutional Networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 727–10 737.

[14]. Y. Wang, Z.-P. Bian, J. Hou, and L.-P. Chau, "Convolutional NeuralNetworks with Dynamic Regularization," *IEEE Transactions on NeuralNetworks and Learning Systems*, Vol. 32, no. 5, pp. 2299–2304, 2020.

[15]. A. Labach, H. Salehinejad, and S. Valaee, "Survey of Dropout Methodsfor Deep Neural Networks," *arXiv preprint arXiv:1904.13310*, 2019.

[16]. [16] L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, T.-Y. Liuet al., "R-drop: Regularized Dropout for Neural Networks," *Advances inNeural Information Processing Systems*, Vol. 34, 2021.

[17]. G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[18]. B. Dong, R. Wang, J. Yang, and L. Xue, "Multi-scale Feature Self-enhancement Network for few-shot learning," *Multimedia Tools andApplications*, pp. 1–19, 2021.

[19]. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov,D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper withConvolutions," in *Proceedings of the IEEE Conference on ComputerVision and Pattern Recognition*, 2015, pp. 1–9.

[20]. Q. Wang, Z. Qin, F. Nie, and Y. Yuan, "Convolutional 2D l-DA forNonlinear Dimensionality Reduction." in *IJCAI*, 2017, pp. 2929–2935.

[21]. S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of aConvolutional Neural Network," in 2017 *International Conference onEngineering and Technology* (ICET). IEEE, 2017, pp. 1–6.

[22]. T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran,"Deep Convolutional Neural Networks for LVCSR," in 2013 *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8614–8618.

[23]. U. Muhammad, W. Wang, S. P. Chattha, and S. Ali, "Pre-trainedVGG-net Architecture for Remote-sensing Image Scene Classification," in*2018 24th International Conference on Pattern Recognition* (ICPR). IEEE, 2018, pp. 1622–1627.

[24]. J. Kolbusz, P. Rozycki, and B. M. Wilamowski, "The Study ofArchitecture MLP with linear neurons in order to eliminate the "vanishing gradient" problem," in *International Conference on ArtificialIntelligence and Soft Computing. Springer*, 2017, pp. 97–106.

[25]. K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *European conference on computer vision. Springer*, 2016, pp. 630–645.

[26]. T. Yu and H. Zhu, "Hyper-parameter Optimization: A review ofAlgorithms and Applications," *arXiv preprint arXiv:2003.05689*, 2020.

[27]. D. Han, Q. Liu, and W. Fan, "A New Image Classification Method usingCNN Transfer Learning and Web Data Augmentation," *Expert Systems withApplications*, Vol. 95, pp. 43–56, 2018.

[28]. K. Yu, W. Xu, and Y. Gong, "Deep Learning with Kernel Regularizationfor Visual Recognition," in *Advances in Neural Information ProcessingSystems*, 2009, pp. 1889–1896.

[29]. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, andR. R. Salakhutdinov, "Improving Neural Networks by preventing Co-adaptation of Feature Detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[30]. A. Rusiecki, "Standard Dropout as remedy for training Deep Neural Networks with Label Noise," in *International Conference on Dependabilityand Complex Systems. Springer*, 2020, pp. 534–542.