Deep learning optimizer performance analysis for pomegranate fruit quality gradation

Rupali Santosh Kale
Computer Science and Technolgy, *Usha Mittal Institute of Technology, SNDT Women's University*
Juhu,Mumbai, India
kalerupalis@gmail.com

Sanjay Shitole
Information Technology, Usha Mittal Institute of Technology, SNDT Women's University
Juhu, Mumbai, India
sanjay.shitole@umit.sndt.ac.in

***Abstract*— Quality and safety are important factors in the food industry. In recent years automatic visual inspection technology has become more potential and important for fruit grading applications. This is because quality is an important factor for consumers and so essential for the market. This paper focuses on a comparative study of deep learning optimizers for pomegranate fruit quality grading. It plays an important role in maximizing the efficiency of the neural network model. Optimizers are mathematical functions or algorithms which are dependent on various parameters of the model i.e., weights and biases. This paper presents the performances of the various deep learning optimizers for pomegranate fruit quality grading. The dataset used for this study is named as Pomegranate Fruit dataset from the Kaggle dataset. Dataset has three grades G1, G2, and G3. Each grade is having four internal quality labels and has 90 images in it. Training is done using SGD, Adadelta, Adagrad, RMSprop, and Adam optimizers. This study helped in analyzing better optimizer and identifying the need for overall improvement in performance of the optimization.**

Keywords—SGD, Adam, Adadelta, Adagrad, RMSprop, Deep learning.

I. INTRODUCTION

In agriculture, quality grading is an essential process in post-harvesting operations. Optimizers are mathematical functions or algorithms which are dependent on various parameters of the model like weight and biases. Many techniques exist for optimization so it is not possible to provide a complete review in this limited article. Instead, efforts are made to concentrate on techniques that are commonly used in deep learning applications. Dataset from Kaggle is used to implement these optimizers in Python and observe the performances for pomegranate fruit quality grading. This data set is having 1093 files of images categorized as G1, G2, and G3 grades. Each grade again having for quality dimensions thus a total of 12 classes are there for study. Optimization tool is considered very important for continuous improvement of output quality to improve quality grading. Picking the right optimizer with appropriate parameters can help in improving the accuracy of the neural network model. Optimisers are responsible for model accuracy. During the training process of the model, weights are updated in the model and try to minimize the loss function and make predictions as correct and optimized as possible. The motivation behind this area of research is a keenness for a quality product. For everything which is being used or utilized in life has more concerned about its quality so the food should also be a quality food. Grading is the sorting or categorization of fruits in two different grades according to size, shape, color, and volume to fetching a high price in the market. Among these features, color is the most important factor when we are doing grading with an automatic system the first thing which attracts customers is its color. In this paper Section 2 describes the related work of optimizers, section 3 discusses novel optimizer results and also results of various optimizers and gives comparative analysis for the same, paper ends with a conclusion and future scope for the readers. This study aims to provide the readers, with a glimpse of the behaviour of different optimization algorithms for multiclass grading and quality assessment of the pomegranate dataset that will allow them to select appropriate optimizers to use. The main advantage of this work is to get an idea of performance with the help of actual results whereas previous literature represents only theoretical comparison of various optimizers.

II. RELATED WORK OF OPTIMIZERS

In this paper basically, six commonly used deep learning optimizers, gradient descent, stochastic gradient descent, Adagrad, RMS prop, Adam, and Adadelta are studied. Authors in [1] studied Adam's optimizer for stochastic optimization. The method studied is suitable for large data as well as for non-stationary objectives and very noisy or sparse gradients. The author has discussed in detail Adam's updation rule for weights and initial bias correction with detailed equations. The authors also discussed RMS Prop and Adagrad which are related to Adam's performance. The author in [2] has given an overview of gradient descent optimization algorithms. The main aim of the paper is to give a behavioural overview of all optimizers that will help users to select the correct optimizer. An author has provided a detailed discussion on the most common optimization algorithms and their architecture and a summary of challenges. In [3] author has provided a review of optimization techniques which are classified as local and global. The author has considered a simple basic genetic algorithm for an explanation. Authors in [4] have done performance analysis of different optimizers for image recognition. The dataset used for analysis is Fruit 360 and it is available on Kaggle. Training is done with SGD, Adam, and RMS Prop and analysed that

RMSprop is giving the best results for image recognition. Authors in [12] developed novel classification model by using some features of CNN along with RNN and LSTM architecture. Fuzzy enhancement is used for image pre-processing. Hyper-parameters are also tuned.

A. Gradient Descent

Gradient descent is the most popular optimizer and it is the grandfather of all optimizers. A discussion of optimizers always starts with gradient descent. This optimization algorithm is used for all machine learning problems. It is fast and flexible. In this method, individual weight is calculated and adjusted on its gradient. This process continues still loss function gets as low as possible. Gradients are partial derivatives; they connect loss function and weights. They tell us what operation we should do to network model weights so that the loss function will get lower. Learning rate is another hyperparameter that is used in optimization algorithms this is the rate at which the network model changes the weight by adding or subtracting. It is nothing but the steps at which the model changes. If these steps are too large then it can hinder the ability to minimize the loss function. Too small a learning rate also does not end up with the right values for weights. So, to avoid this learning rate as a parameter is used in the optimization algorithm which is a very small number. Usually, the learning rate is taken a 0.001, which will be multiplied by the gradients. Too small a learning rate also does not converge to the local minima and a large value of it may miss the optimized step. [6]

B. Stochastic Gradient Descent (SGD)

Like in the previous method, SGD does not calculate gradients for all training examples. Instead, this technique uses batches of examples at a time or random examples on each part in this we calculate the cost of one example for each step which speeds up the neural network model. The equation of SGD is used to update parameters in the neural network in backward pass using backpropagation to calculate the gradient.

$$\theta = \theta - \alpha . \nabla . J(\theta; x, y) \qquad (1)$$

In the above equation, Theta is weight or bias or activations alpha is learning rate, nabla (∇) is gradient which is taken of J and J is an objective function mostly called a cost function or loss function. This method converges slower than newer algorithms to improve the performance momentum added with SGD. Momentum is where the temporal element is added into the equation for updating the parameters of a neural network model. Without momentum it takes bigger oscillations up and down along the y-axis and slower progress along the x-axis. By adding momentum, it is moving faster in x-axis direction, towards the local minima. Advantage of this method is, momentum helps the model to reduce noise but at the same time disadvantage of this method is extra hyperparameters are added.[7]

C. Adagrad

The adaptive gradient descent method is superior to previous methods of optimization. In previous techniques, we studied that the learning rate remains constant. The idea behind Adagard is the use of different learning rates for every neuron for each hidden layer based on different iterations. The advantage of this technique is model can change the learning rate adaptively and also sparse data can be trained easily. If data is sparse then the model can afford a higher learning rate which will boost gradient results and if data is dense then the model can have slower learning. So, the solution for this is to have an adaptive learning rate that can update or modify according to the input given. Adagard technique tries to offer this adaptiveness by decaying the learning rate in proportion to the gradients which are updated. But the disadvantage of this technique is when the decade learning rate reaches to zero. [8]

D. Root Mean Square Propogation (RMSprop)

This technique is an improvement of the Adagard optimizer. The main goal of this optimizer is to reduce the aggressiveness of the learning rate by considering an exponential average of the gradients. This will not allow all gradients to accumulate. The good default value for the learning rate is 0.001. RMSprop and Adadelta both have developed independently to solve Adagrad diminishing learning rates.[9]

E. Adadelta

This optimization technique was developed to solve the problem of Adagrad optimizer for learning rate. When the learning rate becomes very small with a large number of iterations, it leads to slow convergence. Adadelta is solving this problem by taking an exponentially decaying average. [5] It does not allow all past gradients to accumulate but continuously learns the model even though many updates have been done. The advantage of this optimizer is model does not even need to set a default learning rate as it has been eliminated from the update rule.

F. Adaptive Moment estimation (Adam)

This optimizer is latest optimizer technique with improvements in it. It can be looked at as a combination of RMSprop and stochastic gradient descent with momentum. [10] Adam optimizer calculates adaptive learning rates for each parameter. Adam stores an exponentially decaying average of past squared gradients(v_t) like Adadelta and RMSprop and also keeps an exponentially decaying average of past gradients (m_t), similar to momentum. Hyperparameters β1, β 2 control the exponential decay rates of these moving averages m_t & v_t t are calculated as below:

$$m_t = \beta_1 * m_{t-1} + (1-\beta_1)\, g_t \qquad (2)$$

$$v_t = \beta_2 * v_{t-1} + (1-\beta_2)\, gt^2 \qquad (3)$$

In this study of various optimizers performance for the dataset of pomegranate it is reviewed that the Adam optimizer is the best accuracy given in a satisfactory amount of time than other optimizers. RMSprop shows similar accuracy to that of Adam but with comparatively much larger computation time. SGD algorithm took the least time to train and produces good results but to reach the accuracy of the Adam optimizer, it will require more iterations and therefore the computation time also will increase. SGD with momentum shows similar accuracy but with larger computation time. This means that the value of momentum taken needs to optimize. Adadelta shows poor results for both accuracy as well as time.

III. RESULT AND DISCUSSION

All optimizers have the same goal of minimizing a loss function. There are various packages available for the implementation of the optimizer. In this paper, implementation is done using Python and Keras framework. For this analysis, dataset which is used is taken from Kaggle dataset named as Pomegranate fruit dataset. This dataset is having 12 folders contains 90 images in each. These are nothing but 12 different grades given for pomegranate fruit. Some of the samples of different grades are as shown in following figures.

Fig. 1. Dataset sample for G1_Q1 grade

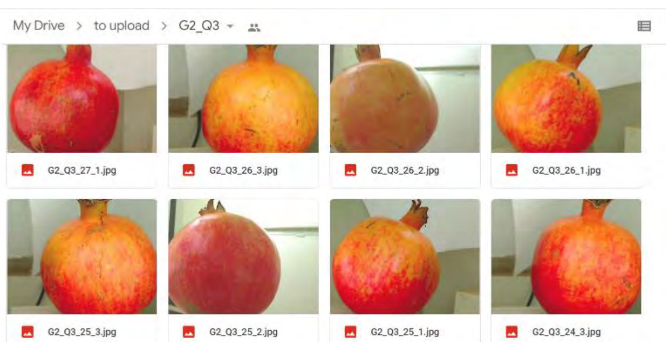

Fig. 2. Dataset sample for G2_Q3

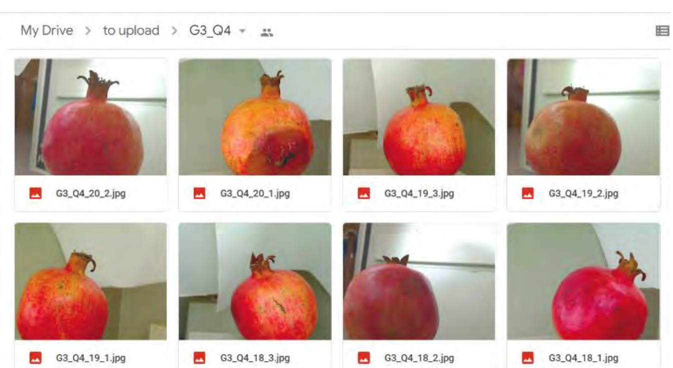

Fig. 3. Dataset sample for G3_Q4

TABLE I. OPTIMIZER COMPARATIVE ANALYSIS

Optimizer	*Loss*	*Val_Accuracy*	*Epochs*	*Exe. Time(ms)*
SGD	2.10	0.23	20	513
Adadelta	2.47	0.21	20	557
Adagrad	2.41	0.19	20	500
Adam	1.00	0.80	20	495
RMSprop	1.77	0.72	20	521

The above table I shows the implementation results of various deep learning optimizers for the pomegranate dataset which is available on Kaggle, we can observe the following points: 1. RMSprop gives good accuracy which is 72% with 1.00 loss in 521ms time whereas for this application Adam performance is 80% and it takes less time than RMSprop for completing 20 epochs. Loss is much less than Adam but accuracy is less.

2. Adam's performance faces better than all other optimization techniques. RMSprop in some applications gives the best results according to the literature survey.

3. As an application is having more classes for classification Adam optimizer is giving less accuracy.

4. There should be more improvement in optimization techniques to improve classification accuracy. Adagard, Adadelta can be used for sparse data classification.

In the implementation all optimization algorithms are used with convolutional neural network. According to the implemented results of various optimization algorithms RMSprop, Adam gives good results in grading applications. Graphical representation of the comparative analysis of various optimizers for loss and val_accuracy and their execution time is as shown in the following figures 4,5 and 6.

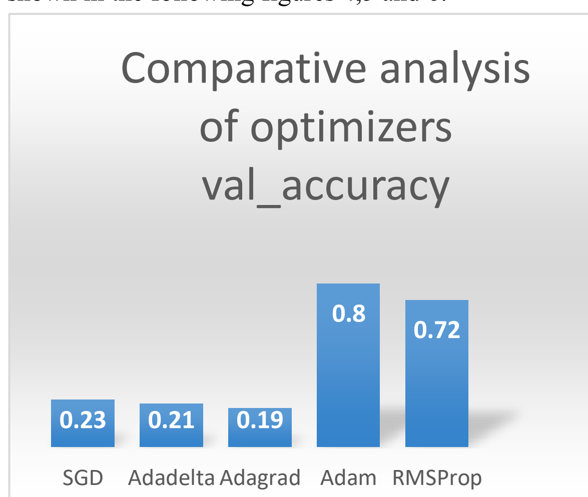

Fig. 4. Comparative analysis of optimizer accuracy

The graph above is showing that Adagrad is taking more execution time and also gives less accuracy, so not used widely. Adadelta is taking less time but still, accuracy is not satisfactory. Adam is taking more execution time but accuracy is more. So, there should be some modification required so that along with accuracy computation time also will be efficient.

Graphical analysis of loss for all 6 optimizers is shown in the figure 5. It is observed that Adam and RMSprop are giving less loss but execution time for RMSprop is less than Adam.

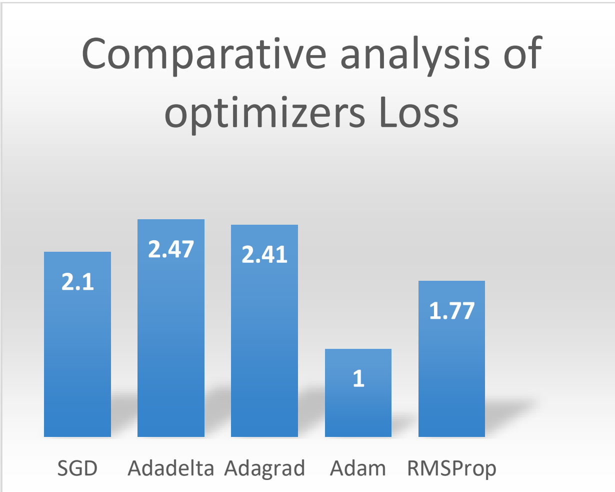

Fig. 5. Comparative analysis of optimizer loss

A. Comparative study of Optimizers with different framework

The main goal of the optimizer is to improve the learning rate and to reduce the loss in classification by improving the loss optimization as the size of the steps taken to reach minimized laws is depend on the learning rate. The Adam optimizer with CNN and MobileNet is executed for the same Pomegranate dataset and it has been observed that the accuracy of grading is 97.8%.CNN with MobileNet and VGG16 framework with Adam are used for comparing the grading results as shown in table 2. This result of grading can be improved by modifying classification framework and improvements in optimization algorithm. Classification with Adam optimizer, CNN and LSTM (Long short-term memory) can be implemented together. It has been observed after implementation of this framework result is further improved with 98.2% shown in fig 4. Further it can be improved by minimum 0.5 % and at the same time execution time can be improved by modifying optimization algorithm. Comparative results as mentioned above are shown in the table below.

TABLE II. COMPARATIVE ANALYSIS OF ADAM OPTIMIZER WITH EXISTING MODEL FRAMEWORK

Model	*Input Size*	*Iterations*	*Accuracy*
VGG16 with Adam	224 X 224	100	96 %
MobileNet with Adam	224 X 224	100	97.8 %

Execution time taken by all the optimizers is noted and shown graphically in the following figure 6. Execution time taken by Adam optimizer is less than all other mentioned optimizers. Previous work for analyzing performance of the optimizers is shown in table III.

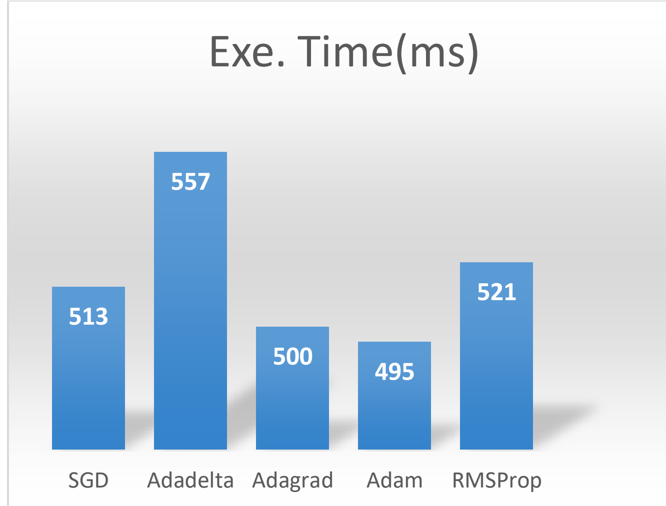

Fig. 6. Comparative analysis of optimizers execution time

The classification report of the Adam optimizer with modified framework is given in figure 7. Classification report gives value of precision, recall, f1-score and support of all the mentioned grades for pomegranate. It is also calculating accuracy based on it, which is 98.22%. So, this analytical study for various deep learning optimizers results that Adam optimization gives efficient results in efficient execution time but for applications having multi grading with multiple classes more accurate results are expected. So, for multigrading applications further improved optimizer is needed.

```
1/1 [==============================] - 0s 10ms/step
there were 21 in 54 tests for an accuracy of  98.22
Classification Report:
---------------------
              precision    recall  f1-score   support

       G1_Q1     1.0000    0.9768    0.9771         5
       G1_Q2     0.9318    0.9542    0.9651         4
       G1_Q3     1.0000    1.0000    1.0000         4
       G1_Q4     0.9861    0.9222    0.9140         4
       G2_Q1     0.9866    1.0000    0.9972         5
       G2_Q2     0.8290    1.0000    0.8965         5
       G2_Q3     1.0000    1.0000    1.0000         4
       G2_Q4     0.8412    1.0000    0.9380         5
       G3_Q1     0.8560    1.0000    0.9206         5
       G3_Q2     0.9798    1.0000    0.9822         5
       G3_Q3     0.8103    1.0000    0.9109         4
       G3_Q4     1.0000    1.0000    1.0000         4

    accuracy                         0.9823        54
   macro avg     0.9210    0.9750    0.9683        54
weighted avg     0.9110    0.9130    0.9772        54
```

Fig. 7. Classification report of Adam optimizer with LSTM framework accuracy for all grades

TABLE III. COMPARISION WITH PREVIOUS WORK

Sr.No	Optimizer	Dataset	Accuracy
1	Adam with CNN-RNN-LSTM [12]	Pomegranate	95%

2	RMSprop with CNN-RNN-LSTM	Pomegranate	88.4%
3	ANN segmentation and SVM classification [13]	Apple	90.3%
4	ANN segmentation and SVM classification [13]	Pomegranate	85%

IV. CONCLUSION

The comparative analysis of six deep learning optimization algorithms is done in this paper to comprehend their application to multi-grading classification problems and to grasp how one optimization technique problem can be solved by another. Approach of performance analysis followed in this paper is beneficial for researchers as live dataset is used for all optimization techniques implementation and then results are discussed. This is observed that Adam's combination with the momentum concept enables its efficient processing to overcome the RMSprop technique problem. It has also been observed that some optimizer for the same loss converges at different local minima. While adaptive learning optimization algorithms converge at sharper minima, other types of optimization algorithms converge at flatter minima. These techniques can help to some extent because as the application and dataset become complex, more efficient methods are required to get good and accurate results. Modified classification framework CNN with LSTM using Adam optimizer gives better accuracy as 98.2% but for multi-grading applications it should be improved more. This study helps in taking decision for selection of appropriate optimization technique. This study of optimizer performance can be used in various applications in agriculture domain like yield prediction, irrigation scheduling etc. The further work of this study is to develop modified optimization technique for multi-grading applications. A thermal image dataset can also be created for grading applications.

REFERENCES

[1] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[2] Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

[3] Venter, G. (2010). Review of optimization techniques.

[4] Postalcıoğlu, S. (2020). Performance analysis of different optimizers for deep learning-based image recognition. International Journal of Pattern Recognition and Artificial Intelligence, 34(02), 2051003.

[5] Alom, M. (2021). Adam optimization algorithm.

[6] Vani, S., & Rao, T. M. (2019, April). An experimental approach towards the performance assessment of various optimizers on convolutional neural network. In 2019 3rd international conference on trends in electronics and informatics (ICOEI) (pp. 331-336). IEEE.

[7] Saleem, M. H., Potgieter, J., & Arif, K. M. (2020). Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers. Plants, 9(10), 1319.

[8] Chen, X., Zhou, G., Chen, A., Pu, L., & Chen, W. (2021). The fruit classification algorithm based on the multi-optimization convolutional neural network. Multimedia Tools and Applications, 80(7), 11313-11330.

[9] Khaing, Z. M., Naung, Y., & Htut, P. H. (2018, January). Development of control system for fruit classification based on convolutional neural network. In 2018 IEEE conference of russian young researchers in electrical and electronic engineering (EIConRus) (pp. 1805-1807). IEEE.

[10] Devikanniga, D., Vetrivel, K., & Badrinath, N. (2019, November). Review of meta-heuristic optimization based artificial neural networks and its applications. In Journal of Physics: Conference Series (Vol. 1362, No. 1, p. 012074). IOP Publishing.

[11] Kumar R, A., Rajpurohit, V. S., & Bidari, K. Y. (2019). Multi class grading and quality assessment of pomegranate fruits based on physical and visual parameters. International Journal of Fruit Science, 19(4), 372-396.

[12] Gill, H. S., & Khehra, B. S. (2021). Hybrid classifier model for fruit classification. Multimedia Tools and Applications, 80(18), 27495-27530.

[13] Unay, D., and B. Gosselin (2005). Artificial neural network-based segmentation and apple grading by machine vision, p. II–630. In: Image Process., 2005. ICIP 2005. Genoa, Italy: IEEE Int. Conf..Vol. 2. IEEE, Sep.

[14] Amani, M. A., & Marinello, F. (2022). A deep learning-based model to reduce costs and increase productivity in the case of small datasets: A case study in cotton cultivation. Agriculture, 12(2), 267.

[15] Lachgar, M., Hrimech, H., & Kartit, A. (2022). Optimization techniques in deep convolutional neuronal networks applied to olive diseases classification. Artificial Intelligence in Agriculture.

[16] Mythili, K., & Rangaraj, R. (2021). Crop Recommendation for Better Crop Yield for Precision Agriculture Using Ant Colony Optimization with Deep Learning Method. Annals of the Romanian Society for Cell Biology, 4783-4794.

[17] Elaraby, A., Hamdy, W., & Alruwaili, M. (2022). Optimization of deep learning model for plant disease detection using particle swarm optimizer. Computers, Materials & Continua, 71(2), 4019-4031.

[18] Vignesh, K., Askarunisa, A., & Abirami, A. M. (2023). Optimized Deep Learning Methods for Crop Yield Prediction. COMPUTER SYSTEMS SCIENCE AND ENGINEERING, 44(2), 1051-1067.

[19] Zhu, N., Liu, X., Liu, Z., Hu, K., Wang, Y., Tan, J., ... & Guo, Y. (2018). Deep learning for smart agriculture: Concepts, tools, applications, and opportunities. International Journal of Agricultural and Biological Engineering, 11(4), 32-44.

[20] Nasir, I. M., Bibi, A., Shah, J. H., Khan, M. A., Sharif, M., Iqbal, K., ... & Kadry, S. (2021). Deep learning-based classification of fruit diseases: An application for precision agriculture.