**Library Progress International**
*Vol.44 No.3, Jul-Dec 2024: P.20896-20906*

*Print version ISSN 0970 1052*
*Online version ISSN 2320 317X*

*Original Article* *Available online at www.bpasjournals.com*

# The Empirical Comparison of Deep Neural Network Optimizers for Binary Classification of OCT Images

**R. Loganathan[1] , S. Latha[2*]**

[1,2]Department of Electronics and Communication Engineering, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur Campus, Chengalpattu, Tamil Nadu, India
[1]email:lr1054@srmist.edu.in [1]ORCID ID : 0009-0006-2885-3672
[2]email: lathas3@srmist.edu.in [2]ORCID ID : 0000-0001-7978-7720 *Corresponding author

**How to cite this article:** R. Loganathan, S. Latha (2024) The Empirical Comparison of Deep Neural Network Optimizers for Binary Classification of OCT Images. *Library Progress International,* 44(3), 20896-20906

**Abstract**

*To determine the optimum solution to a problem, optimizers are employed in a variety of subjects, including statistical analysis, mathematics, and computing. For the past few years, the Adaptive Moment Estimation, sometimes known as Adam, has been more popular for use as an optimizer in deep learning models. It is efficient and utilizes minimal memory. The algorithm "gradient descent with momentum" and "Root Mean Square Propagation (RMSP)" are intuitively combined in this method. To classify normal vs. AMD binary data on the optical coherence tomography (OCT) image dataset, the main objective of this work is to determine which optimizer yields the minimum loss and maximum accuracy throughout the training and testing phases of CNN, DNN, and VGG16 models. SGD, SGD with momentum, Agadrad, Adam, and RMS prop are the optimizers under investigation at different learning rates. Optimizing the learning rate and other hyperparameters enhances the deep learning model's accuracy and loss according to experiments. Thus, the results demonstrate that the Adam optimizer provides superior performance over all optimizers. The Adam optimizer could train all binary convolutional neural networks based on these results.*

***Keywords-*** *OCT images; image classification; loss; accuracy; optimizer; deep learning;*

## 1. INTRODUCTION

The number of people aged 50 and more is expected to reach 9.8 billion worldwide by 2050. The population of people aged 80 and above is projected to triple, from 132 million in 2020 to 426 million in 2050 (Bourne et al., 2014). Fine vision, color vision, and other visual capabilities depend on the retina's macular region. Damage or lesions in the macular region may impair vision and cause blindness. Age-related macular degeneration (AMD), also known as senile macular degeneration, is a prevalent macular condition that affects persons over 45. One of the leading causes of blindness in older people is AMD, which rises with age. Early AMD identification is necessary for effective treatment (Bressler, 2002; Friedman, 2004). OCT is commonly used to diagnose AMD and other retinal disorders. Based on weak coherence light interference, OCT detects the return reflection or multiple dispersed signals in various layers of biological tissue. OCT scans biological tissue to provide two or three-dimensional structural images for eye disease diagnostics (Adhi & Duker, 2013). To segment images effectively, use spatial similarity and steerable filters to blend color and texture data, followed by SVM classification using FCM (Barui et al., 2018). Variable gradient summation makes speckle noise additive while preserving edge characteristics. Wavelet decomposition, NLM, VTV, and BM3D filters improve 300 carotid artery ultrasound images (Latha & Samiappan, 2019).

Deep learning methods, particularly Convolutional Neural Networks (CNN), have shown great success in medical image analysis, with automated diagnosis by artificial intelligence being the focus of specialist physicians due to its high speed and low error rate. The images were sent into a deep neural network trained to distinguish between

normal and AMD eyes. Accuracy was 87.63 percent, and the area under the ROC was 92.78% at the image level. Accuracy was 88.98%, and the area under the ROC was 93.83% at the macula level. Deep learning image classification is accurate and efficient. These results are crucial for using OCT in automated screening and developing computer-aided diagnostic systems.

To classify data using deep learning, trained a VGG16 convolutional neural network variant. SGD optimized 100 images. After 500 iterations, they compared the model loss to the validation set. The decreased loss and validation accuracy ended training (Lee et al., 2017). Eliminating a few deep convolutional layers from pre-trained deep neural networks (DNN) may improve retinal OCT image classification. They tested deep neural network setups for small and large OCT datasets. Experimentally, enhanced deep neural networks improve classification accuracy and computing load (Ji et al., 2019). The objective of this study is to identify the best optimizers for the binary classification of normal vs AMD OCT images.

## 1. RELATED WORK

The image-based deep learning approaches employ the Inception-v3 model that has been pre-trained. The weights of the convolutional layers are modified, and subsequently, the final dense (softmax) layer is fine-tuned. All sub-network layers are finetuned in this paper. Adam optimizer trains layers in batches of 32 with a learning rate of 0.001, decay rate of 0.3 (Inception-v3 and ResNet50) or 0.01 (DenseNet121), β1 of 0.9, and β2 of 0.999. A 20-epoch training period ensures convergence. Images in Inception-v3 are 299 x 299, whereas ResNet50 and Dense-Net121 are 224 x 224. A pre-processed dataset is trained and tested using 280 AMD, DME, and NOR images. All sub-network layers were finetuned as in the large-scale experiment. Thirty epochs instead of 20 trained the tiny dataset 10 times in each experiment. Mean ± SD accuracy, sensitivity, and specificity were computed (Kermany et al., 2018).

As a result of recent empirical comparisons, it has been shown that the meta-parameter search space is the most important factor when selecting an optimizer, and inclusion relationships between optimizers matter in practice, with more general optimizers always outperforming special cases (Herrera-Alcántara & Castelán-Aguilar, 2023). This paper analyses how optimizers affect CNN performance. Two experiments were done. From scratch, SGD, Adam, Adadelta, and AdaGrad optimized the VGG11 CNN. Second, the same four optimizers tweaked AlexNet CNN to classify Persian handwritten words. Adam and AdaGrad have superior training costs and recognition accuracy. Lower initial learning rates result in quicker convergence for the Adam optimizer (Błasiok et al., 2023).

Optimizers change weight parameters to reduce the error or loss function, making them essential to neural network training (Chen et al., 2023). The optimizer minimizes the error function, the difference between the actual and projected values (Ando et al., 2023). Select the optimizer and its algorithmic hyperparameters before training. The optimizer's choice affects model performance. The issue and dataset determine the optimizer (Choi et al., 2019). This research compares SGD, SGD with momentum, Adagrad, RMSprop, and Adam optimizers (Zohrevand & Imani, 2022).

SGD, one of the first neural network training algorithms, is simple. SGD with Momentum accelerates convergence by considering prior updates, overcoming oscillations, and high curvature in the loss landscape. Adagrad's historical gradient-based learning rate fits sparse data and informative features. Squared gradients decrease and increase the speed. RMSprop fixed the AdaGrad optimizer's monotonically declining learning rate. Adam combines AdaGrad and RMSProp techniques for performance and robustness. To maximize model performance, the optimizer and hyperparameters must be carefully chosen.

## 2. METHODOLOGY

### 3.1 Model

*Convolutional neural network (CNN) model*: A CNN model is constructed using several convolutional layers, and then max-pooling layers. Through non-linear activations and learnable filters, these layers expedite the process of

discovering spatial hierarchy in the input images. The self-features module, part of the CNN class, defines the convolutional layers used to extract features from the input images. ReLU activation functions are utilized in conjunction with max-pooling layers, which downsample the feature maps to achieve non-linearity (Das et al., 2021; Kim & Tran, 2020).

*Dense neural network (DNN) model*: The DNN model is based on a feed-forward neural network with many fully connected layers. The DNN class's self-flatten module maps the input image into a one-dimensional tensor to facilitate feature representation. At this phase, the data from the image's spatial components are transformed into a feature vector. The normalized feature vector is fed via fully linked layers activated by ReLU to learn the classification (Mathews & Anzar, 2022).

*Visual Geometry Subgroup 16 (VGG16)*: The VGG16 deep CNN architecture is well-known for being very complex and remarkably easy to understand. After that, multiple convolutional layers with tiny filter sizes come after many max-pooling layers. Component Detection Features extracted using convolutional layers are specified in the VGG16 classes self-features module. To learn and downsample the feature maps, employ ReLU activations and max-pooling layers. Data classification is handled by the self-classifier module of the VGG16 class, which also specifies the fully linked layers. Flattening the convolutional layer output and passing it through ReLU-activated fully connected layers yields the final classification decision. Classification is handled by the CNN class's self-classifier module, which creates the necessary fully linked layers. To learn the final classification decision, ReLU activates fully connected layers once the output of the convolutional layers has been flattened (Kim & Tran, 2021). The deep learning models are implemented in Python using TensorFlow Keras (Optimizer, n.d.; Tensorflow, n.d.).

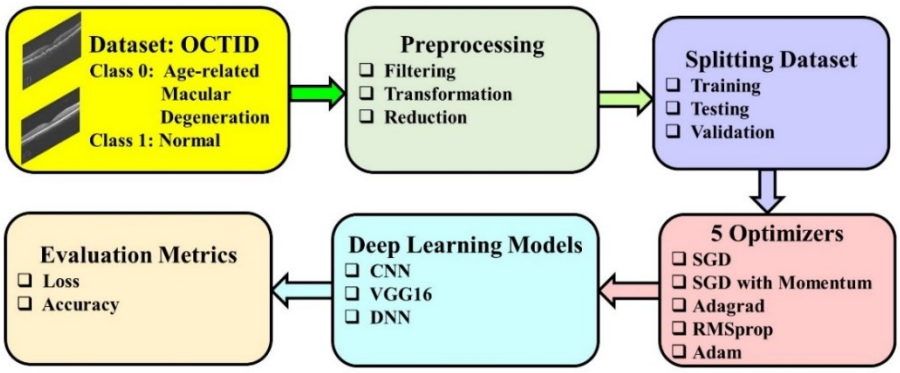



**Figure 1.** Optimization with Deep Learning Models

Each of the three models is trained independently, with the parameters defined by backpropagation and optimization iterations shown in Figure 1. The cross-entropy loss method calculates loss and solves problems involving multiple classification categories. Throughout the iteration of the training and testing loops, the forward pass, loss, and optimizer are all calculated using the continuous data stream. The model's parameters are then revised with the aid of the optimizers. CNN, DNN, and VGG16 model training and assessment using OCTID datasets may all use the described approach and scripts as a starting point

### 3.2 Dataset

OCT images from AMD patients and normal people were used in the investigation. This dataset supports comparative analysis and AMD retinal abnormality study (Gholami et al., 2020). It consists of a validation set, a training set, and a test set. All OCT scans employed a 224 × 224 × 3 image size with 3 colour channels. This protocol was implemented to ensure compatibility with deep learning (DL) based optimization algorithms in future generations. The dataset, including the training and test sets for both AMD and normal OCT images, is shown in Table 1.

**Table 1.** Dataset, training and testing set, OCTID database of AMD and Normal retina

| Category | Count | Training set | Testing set |
|---|---|---|---|
| Normal Retina | 206 | 144 | 21 |
| AMD | 55 | 38 | 6 |
| Total Images | 261 | 182 | 27 |

### 3.3 Evaluation metrics

*Train Loss*: It measures the mismatch between the model's predictions and actual labels. Smaller loss numbers suggest a higher level of accuracy in matching predictions with actual labels. Less train loss means the model is learning training data patterns accurately. Measures model adherence to training data.

*Test Loss*: The differences between the model's predictions and the actual scores on the testing set are quantified as the test loss. A lower value of test loss signifies that the model is generalizing its learned patterns to novel, unobserved data with greater accuracy. It evaluates model generalization on untrained data.

*Train Accuracy*: Training accuracy evaluates the ratio of accurate predictions generated by the model on the training dataset. A high score indicates that the model has learned the training data effectively.

*Test Accuracy*: A model's predictive power on new data is evaluated by testing its accuracy. Higher testing accuracy shows that the model has a larger generalization performance.

### 3.4 Experimental Setup

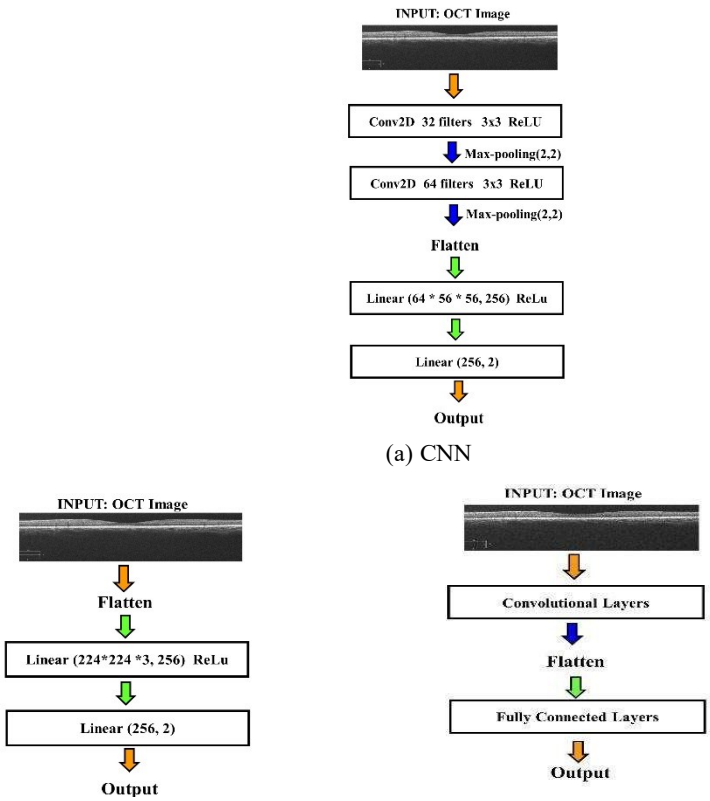


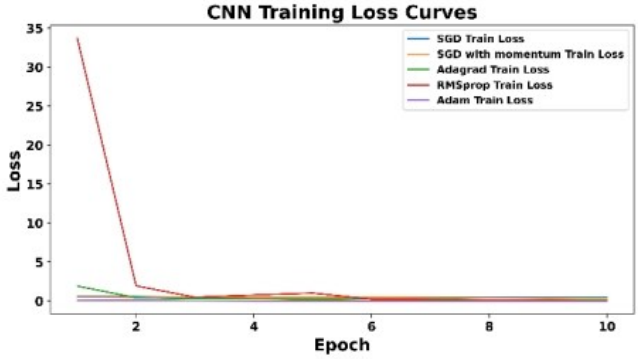

**Figure 2.** The architecture of the (a) CNN, (b) DNN, and (c) VGG16 networks

The current study investigates the impact of five optimization algorithms Adam, Adagrad, RMSprop, stochastic gradient descent (SGD), and SGD with momentum, on three different deep learning models. More precisely, the models used were CNN, DNN, and VGG16 deep neural networks shown in Figure 2. Each optimizer was used to train and test every model for a total of 10 epochs. OCT image data was gathered for each optimizer about performance measures such as loss, training accuracy, and testing accuracy. A comprehensive examination of experimental data was conducted to evaluate the efficacy of optimization techniques on the training of CNN, DNN, and VGG16 models for classification work.

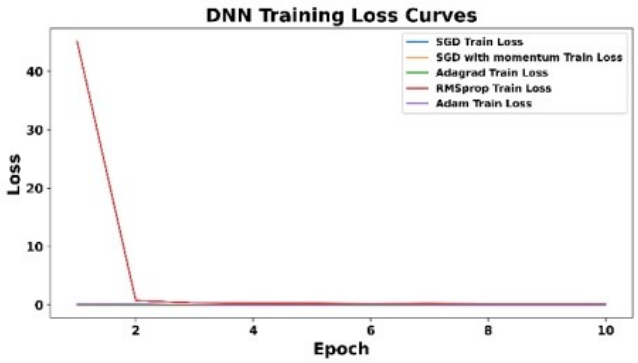
## 3. RESULTS AND DISCUSSION

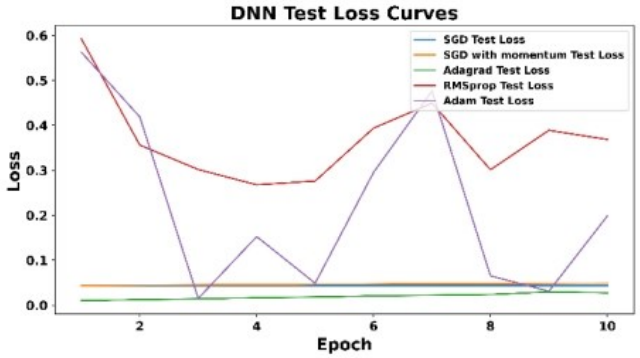
### 4.1 Findings

To experiment comparing the efficacy of five different optimizers (SGD, SGD with momentum, Adagrad, RMSprop, and Adam) across three distinct models (Convolutional Neural Network (CNN), Deep Neural Network (DNN), and VGG16). This experiment aims to compare and contrast the models' performance in training and testing using various optimization strategies.

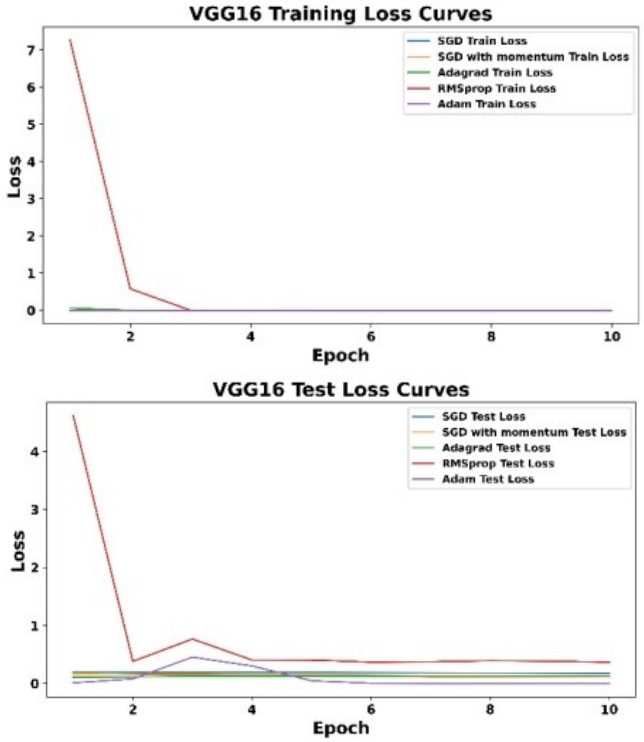



(a) CNN Model






(b) DNN Model




(c) VGG16 Model

**Figure 3.** Loss curves for various optimizers at Learning Rate = 0.001, for (a) CNN (b) DNN, and (c) VGG16 Model

Ten times the model was trained. To improve accuracy, use more epochs through the OCTID dataset. After training on the OCTID dataset, the model is tested on 27 test images. Accuracy and loss could evaluate model performance. Models with the lowest loss and highest accuracy perform well. This comparison shows which model is best.

The experiment's findings provide insight into how the efficacy of different optimizers affects the precision with which CNN, DNN, and VGG16 models are trained and tested. By comparing the outcomes, you may determine which optimizers result in better convergence and more accurate models for each kind of data. These findings might be a basis for future optimization strategy decisions for comparable tasks and models.

The train and test loss is too high in the RMSprop optimizer at the initial epoch for all 3 deep learning models shown in Figure 3(a) CNN, Figure 3(b) DNN, and Figure 3(c) VGG16 models. Adam optimizers are more suited to the binary classification of OCT images, and after 10 epochs of training, the loss is lower than with other optimizers. All 3 models exhibited the worst performance for SGD optimizers for producing the highest losses, and the RMSprop optimizer was an average performer for the CNN and DNN models. Therefore, it can be shown that all three deep learning models exhibit the lowest loss minimization performance when applied with the Adam optimizer.

**Table 2.** Train Loss analysis of deep learning models for different optimizers with different learning rates

| **Models** | **CNN** | | | **DNN** | | | **VGG16** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Optimizers / Learning rate** | **0.001** | **0.0001** | **0.00001** | **0.001** | **0.0001** | **0.00001** | **0.001** | **0.0001** | **0.00001** |
| SGD | 0.4917 | 0.5204 | 0.6669 | 0.0003 | 0.0043 | 0.6318 | 0.0001 | 0.0001 | 0.3253 |
| SGD with momentum | 0.3692 | 0.4818 | 0.5446 | 0.0003 | 0.0039 | 0.6193 | 0.0000 | 0.0001 | 0.3243 |
| Adagrad | 0.1278 | 0.4500 | 0.4542 | 0.0000 | 0.0023 | 0.5860 | 0.0000 | 0.0000 | 0.3100 |
| RMSprop | 0.0221 | 0.0831 | 0.6922 | 0.1759 | 0.0032 | 0.3857 | 0.0001 | 0.0004 | 0.2571 |
| Adam | 0.0004 | 0.0066 | 0.6437 | 0.0001 | 0.0001 | 0.3328 | 0.0000 | 0.0000 | 0.2228 |

**Table 3.** Test Loss analysis of deep learning models for different optimizers with different learning rates

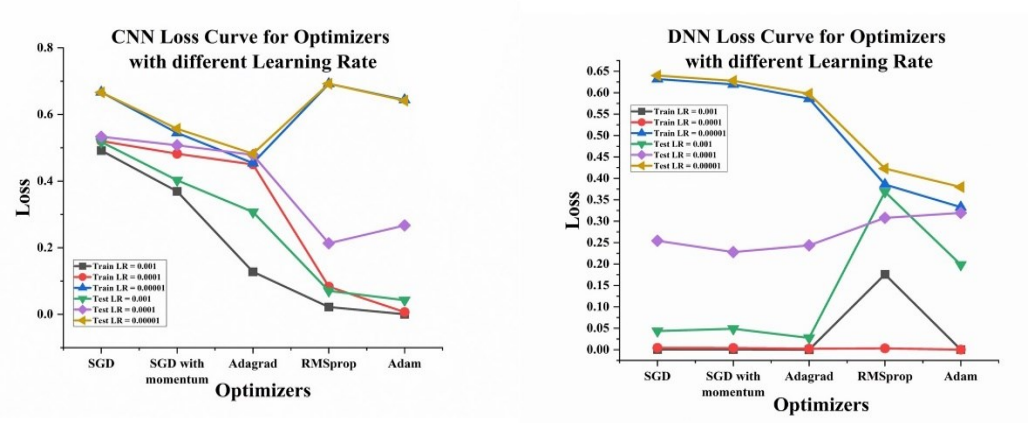
| **Models** | **CNN** | | | **DNN** | | | **VGG16** | | |
|---|---|---|---|---|---|---|---|---|---|
| **Optimizers / Learning rate** | **0.001** | **0.0001** | **0.00001** | **0.001** | **0.0001** | **0.00001** | **0.001** | **0.0001** | **0.00001** |
| SGD | 0.5175 | 0.5331 | 0.6662 | 0.0435 | 0.2542 | 0.6401 | 0.1770 | 0.3194 | 0.3788 |
| SGD with momentum | 0.4024 | 0.5076 | 0.5570 | 0.0489 | 0.2279 | 0.6277 | 0.1201 | 0.3186 | 0.3719 |
| Adagrad | 0.3071 | 0.4787 | 0.4823 | 0.0274 | 0.2435 | 0.5976 | 0.1161 | 0.3020 | 0.3629 |
| RMSprop | 0.0705 | 0.2132 | 0.6924 | 0.3685 | 0.3077 | 0.4228 | 0.3658 | 0.4000 | 0.3037 |
| Adam | 0.0425 | 0.2670 | 0.6415 | 0.1984 | 0.3195 | 0.3796 | 0.0012 | 0.3904 | 0.2797 |

### 4.1 Discussion

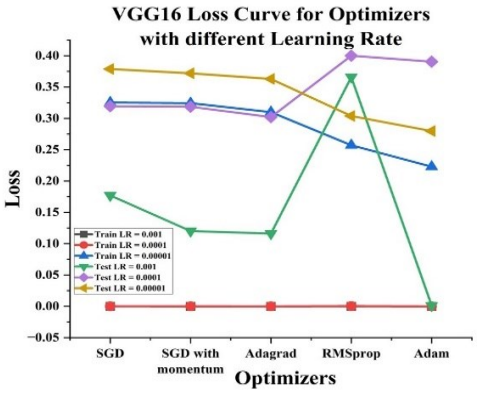
The outcomes of the efficacy of various optimizers and learning rates for the CNN, DNN, and VGG16 architectures are comprehensively presented in Table 2 and Table 3. Adam and Adagrad consistently exhibit superior performance across all models and learning rates. Overall learning rates, the Adam optimizer produced the lowest loss for the CNN model. More precisely, Adam optimizer loss is the least at the maximum learning rate (0.001). In contrast, the maximum loss resulted from the SGD consistently lagging, proved to be the least effective optimizer. This tendency remained constant for different learning rates.

While Adam optimizer once again outperformed all other approaches at medium learning rates (0.0001) in the DNN model, Adagrad unexpectedly achieved the lowest loss (0.00001) across the board. Among the VGG16 optimizers, Adagrad outperformed the others at the slowest learning rate, whereas Adam performed admirably at all learning rates.

Figure 4(a) shows that the Adam optimizer is better for the CNN model than other optimizers, with consistently reduced loss across varied training and test loss for all learning rates. SGD and SGD with momentum exhibit the highest losses whereas Adagrad and RMSprop make average performance. Overall, the Adam optimizer minimized loss best for all three deep learning models, especially at medium learning rates.




(a) CNN (b) DNN




(a) VGG16

**Figure 4.** Loss curves for various optimizers with different Learning Rates for (a) CNN (b) DNN and (c) VGG16 Model.

According to Figure 4(b), the DNN model that was optimized using the Adagrad and Adam optimizer observed the least amount of train and test loss when compared with all other optimizers for all learning rates. This Graph demonstrates loss and optimizer for different learning rates during the training, where RMSprop shows it produces moderate performance, while SGD and SGD with momentum consistently lag, with the highest losses.

According to Figure 4(c), the VGG16 model with Adam and Adagrad optimizer produced the lowest loss compared to other optimizers for all the learning rates. This Graph demonstrates loss and optimizer for different learning rates during the training, where SGD and SGD with momentum produce the highest loss for all available learning rates, while RMSprop performs moderately.

Table 4 summarizes the training accuracy values that were attained during the training of the CNN, DNN, and VGG16 models with various optimizers and learning rates. The Adam optimizer produced the highest training accuracy, 100.00% for all three deep learning models with the highest (0.001) and medium (0.0001) learning rates. In contrast, the training accuracy is 100 % for all the optimizers for the VGG16 model with the highest and medium learning rates.

For the medium learning rate (0.0001), all five optimizers perform best for the DNN and VGG16 models. For the lowest learning rate (0.00001), SGD and SGD with momentum performed the lowest training accuracy of 79.12% for all 4 optimizers except the Adam optimizer which performs well in the CNN model. Whereas the SGD optimizer performed best for the DNN model and Adam optimizer is best for the VGG16 model.

**Table 4.** Training accuracy analysis of deep learning models for optimizers with different learning rates

| Models | CNN | | | DNN | | | VGG16 | | |
|---|---|---|---|---|---|---|---|---|---|
| **Optimizers / Learning rate** | **0.001** | **0.0001** | **0.00001** | **0.001** | **0.0001** | **0.00001** | **0.001** | **0.0001** | **0.00001** |
| SGD | 79.12 | 79.12 | 79.12 | 100.00 | 100.00 | 92.85 | 100.00 | 100.00 | 84.07 |
| SGD with momentum | 81.86 | 79.12 | 79.12 | 100.00 | 100.00 | 87.36 | 100.00 | 100.00 | 85.71 |
| Adagrad | 95.60 | 79.12 | 79.12 | 100.00 | 100.00 | 80.77 | 100.00 | 100.00 | 86.81 |
| RMSprop | 100.00 | 98.35 | 79.12 | 79.67 | 100.00 | 79.67 | 100.00 | 100.00 | 91.21 |
| Adam | 100.00 | 100.00 | 94.50 | 100.00 | 100.00 | 84.07 | 100.00 | 100.00 | 92.31 |

Table 5 shows CNN, DNN, and VGG16 architecture testing accuracy for optimizers with varied learning rates. The CNN model had the best training accuracy (96.29%) using the RMSprop and Adam optimizer, while the DNN model had the highest learning rate (0.001). All VGG16 optimizers have 77.78% training accuracy. For the medium learning rate (0.0001), Adam and RMSprop optimizer work best for CNN and VGG16 models, whereas Adam, SGD, and SGD with momentum perform best for DNN.

For the lowest learning rate (0.00001), all 4 optimizers have the lowest testing accuracy of 77.78% for the CNN model, 3 optimizers for the DNN model, and SGD optimizer for the VGG16 model. For the CNN model with all learning rates, the Optimizer Adam reduces losses in iterations while maintaining its optimum accuracy. The Adam Optimizer reduces the amount of loss that occurs as the number of epochs increases, while simultaneously increasing the results' dependability. Adam Optimizer achieves optimal performance with an optimal score of 100% in VGG16 for the highest learning rate.

For the medium learning rate (0.0001), The Adam optimizer reduces loss for the duration of an epoch; it maintains the best test accuracy of 92.59% for all the 3 deep learning models, respectively. SGD and SGD with momentum optimizers perform best for the slowest learning rate (0.00001) for the DNN model. In comparison, Adam optimizers perform best for CNN and VGG16 models with the lowest loss and good accuracy.

**Table 5.** Testing accuracy analysis of deep learning models for optimizers with different learning rates

| Models | CNN | | | DNN | | | VGG16 | | |
|---|---|---|---|---|---|---|---|---|---|
| **Optimizers / Learning rate** | **0.001** | **0.0001** | **0.00001** | **0.001** | **0.0001** | **0.00001** | **0.001** | **0.0001** | **0.00001** |
| SGD | 77.78 | 77.78 | 77.78 | 96.29 | 92.59 | 88.89 | 96.29 | 92.59 | 77.78 |
| SGD with momentum | 77.78 | 77.78 | 77.78 | 96.29 | 92.59 | 81.48 | 96.29 | 92.59 | 81.48 |
| Adagrad | 88.89 | 77.78 | 77.78 | 96.29 | 92.59 | 77.78 | 96.29 | 92.59 | 81.89 |
| RMSprop | 96.29 | 88.89 | 77.78 | 77.78 | 92.59 | 77.78 | 96.29 | 92.59 | 88.89 |
| Adam | 96.29 | 92.59 | 88.89 | 96.29 | 92.59 | 77.78 | 100.00 | 92.59 | 88.89 |

## CONCLUSIONS

CNN easily performs the task of classifying the images. The function of optimizers and their effect on the classifier's performance are extensively discussed. For binary classification of normal vs. AMD OCT images, The ADAM optimizer has been shown to achieve a training accuracy of 100% at a maximum learning rate of 0.001 for all three deep learning models. The CNN model, coupled with the ADAM optimizer, is a great choice for the binary classification of OCT images compared with VGG16 and DNN. After precise testing, the Adam optimizer proved to be the most effective in accurately classifying OCT images as either AMD or Normal while minimizing the loss for all learning rates across CNN, DNN, and VGG16 models.

**References:**


R.R. Bourne, J.B. Jonas, S.R. Flaxman, J. Keeffe, J. Leasher, K. Naidoo, M.B. Parodi, K. Pesudovs, H. Price, R.A. White, and T.Y. Wong, (2014). Prevalence and causes of vision loss in high-income countries and in Eastern and Central Europe: 1990–2010. *British Journal of Ophthalmology*, 98(5), pp.629-638. doi: 10.1136/bjophthalmol-2013-304033.

D.S. Friedman, B.J. O'Colmain, B. Munoz, S.C. Tomany, C. McCarty, P.T. De Jong, B. Nemesure, P. Mitchell, and J. Kempen, (2004). Prevalence of age-related macular degeneration in the United States. *Archives of ophthalmology*, 122(4), pp.564-572. doi: 10.1001/archopht.122.4.564.

N.M. Bressler, (2002). Early detection and treatment of neovascular age-related macular degeneration. *The Journal of the American Board of Family Practice*, 15(2), 142-152.

M. Adhi, and J.S. Duker, (2013). Optical coherence tomography–current and future applications. *Current opinion in ophthalmology*, 24(3), pp.213-221. doi: 10.1097/ICU.0b013e32835f8bf8

S. Barui, S. Latha, D. Samiappan, & P. Muthu, (2018, April). SVM pixel classification on colour image segmentation. In *Journal of Physics: Conference Series*, Vol. 1000, No. 1, p. 012110. IOP Publishing. doi:10.1088/1742-6596/1000/1/012110

S. Latha, & D. Samiappan, (2019), Despeckling of carotid artery ultrasound images with a calculus approach. *Current Medical Imaging*, 15(4), 414-426. doi: 10.2174/1573405614666180402124438

Lee, C. S., D. M. Baughman, and A. Y. Lee. (2017). Deep learning is effective for the classification of OCT images of normal versus age-related macular degeneration. *Ophthalmology Retina*, 1(4), 322–327. https://doi.org/10.1016/j.oret.2016.12.009

Ji, Q., Huang, J., He, W., & Sun, Y. (2019). Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images. *Algorithms*, 12(3), 51. https://doi.org/10.3390/a12030051

D.S. Kermany, M. Goldbaum, W. Cai, C.C. Valentim, H. Liang, S.L. Baxter, and K. Zhang, (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *cell*, 172(5), 1122-1131. doi: 10.1016/j.cell.2018.02.010.

O. Herrera-Alcántara, & J.R. Castelán-Aguilar, (2023). Fractional Gradient Optimizers for PyTorch: Enhancing GAN and BERT. *Fractal and Fractional*, 7(7), 500. https://doi.org/10.3390/fractalfract7070500

J. Blasiok, P. Gopalan, L. Hu, and P. Nakkiran, (2024). When Does Optimizing a Proper Loss Yield Calibration?. *Advances in Neural Information Processing Systems*, 36.

C.H. Chen, J.P. Lai, Y.M. Chang, C.J. Lai, & P.F. Pai, (2023). A Study of Optimization in Deep Neural Networks for Regression. *Electronics*, 12(14), 3071.

R. Ando, Y. Fukuhara, and Y. Takefuji, (2023). Characterizing Adaptive Optimizer in CNN by Reverse Mode Differentiation from Full-Scratch. *Indian Journal of Artificial Intelligence and Neural Networking,* 3(4), 1-6. doi: 10.54105/ijainn.D1070.063423

D. Choi, C.J. Shallue, Z. Nado, J. Lee, C.J. Maddison, & G.E. Dahl, (2019). On empirical comparisons of optimizers for deep learning. *arXiv preprint arXiv:1910.05446*. https://doi.org/10.48550/arXiv.1910.05446

A. Zohrevand and Z. Imani, (2022, February) An Empirical Study of the Performance of Different Optimizers in the Deep Neural Networks. In *2022 International Conference on Machine Vision and Image Processing* (MVIP), Ahvaz, Iran, Islamic Republic of, 2022, pp. 1-5, doi: 10.1109/MVIP53647.2022.9738743.

V. Das, S. Dandapat and P. K. Bora, (2021). Automated Classification of Retinal OCT Images Using a Deep Multi-Scale Fusion CNN. In *IEEE Sensors Journal*, vol. 21, no. 20, pp. 23256-23265, doi: 10.1109/JSEN.2021.3108642

J. Kim and L. Tran, (2020, July). Ensemble Learning Based on Convolutional Neural Networks for the Classification of Retinal Diseases from Optical Coherence Tomography Images. In *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, USA, pp. 532-537, doi: 10.1109/CBMS49503.2020.00106.

M. R. Mathews and S. M. Anzar, (2022, August). Residual Networks and Deep-Densely Connected Networks for the Classification of retinal OCT Images. In *2022 International Conference on Connected Systems & Intelligence (CSI)*, Trivandrum, India, pp. 1-7, doi: 10.1109/CSI54720.2022.9923993

J. Kim and L. Tran, (2021, October). Retinal Disease Classification from OCT Images Using Deep Learning Algorithms. In *2021 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Melbourne, Australia, pp. 1-6, doi: 10.1109/CIBCB49929.2021.9562919.

Optimizer. (n.d.). *https://keras.io/api/optimizers/*.

Tensorflow. (n.d). *https://www.tensorflow.org/*

P Gholami, P Roy, MK Parthasarathy, V Lakshminarayanan, (2020). OCTID: Optical coherence tomography image database. *Computers & Electrical Engineering*, 81, 106532. https://doi.org/10.1016/j.compeleceng.2019.106532