**International Journal of Engineering and Computer Science Applications (IJECSA)**
Vol. 4, No. 2, September 2025, pp. 81∼90
ISSN: 2828-5611
DOI: doi.org/10.30812/ijecsa.v4i2.5064


# Feature Extraction in Eye Images Using Convolutional Neural Network to Determine Cataract Disease

**Khasnur Hidjah[1], Fitra Rizki Ramdhani[1], Muhammad Zulfikri[1], Hairani Hairani[1], Mayadi Mayadi[2], Ni Gusti Ayu Dasriani[1], Juvinal Ximenes Guterres[3]**
[1]Universitas Bumigora, Mataram, Indonesia
[2]University Teknologi Mara, Selangor, Malaysia
[3]Universidade Nacional Timor Lorosa'e, Dili, Timor-Leste

**Article Info**

**Article history:**

Received May 20, 2025
Revised May 27, 2025
Accepted June 06, 2025

**Keywords:**

Cataract
Convolutional Neural Network
Model Testing
Optimizer

***ABSTRACT***

The eye is one of the vital human senses and is the main organ for vision. One of the visual impairments that requires special attention is blindness, and cataracts are a major cause of it. A cataract is a condition in which the eye's lens becomes cloudy due to changes in the lens fibers or materials inside the capsule. This cloudiness blocks light from entering the eye and reaching the retina, significantly interfering with vision. Early detection of cataracts is essential to prevent blindness. An efficient image-based classification model is needed for cataract detection. **This study aims** to test the Convolutional Neural Network (CNN) model for early cataract detection by exploring the use of several optimization algorithms: Adaptive Moment Estimation (Adam), Root Mean Square Propagation (RMSprop), Adaptive Gradient Algorithm (AdaGrad), and Stochastic Gradient Descent (SGD). **The research method** follows an experimental approach, where eye image datasets are trained using the same CNN architecture but with different parameter configurations. **The results show** that the Adam optimizer, with a data split of 70% for training, 15% for validation, and 15% for testing over 50 epochs, produced the best results, achieving accuracies of 94%, 93%, and 93%, respectively. Other optimizers performed reasonably well but could not match Adam's stability and accuracy. **The implication of this research is** that the choice of optimizer and hyperparameter configuration plays a crucial role in improving the performance of image-based cataract detection models.



**Corresponding Author:**

Khasnur Hidjah,
Universitas Bumigora, Mataram, Indonesia,
Email: khasnur72.h@universitasbumigora.ac.id

**How to Cite:** K. Hidjah, F. R. Ramdhani, M. Zulfikri, H. Hairani, M. Mayadi, N. G. A. Dasriani, amd J. X. Guterres, "Feature Extraction in Eye Images Using Convolutional Neural Network to Determine Cataract Disease," *International Journal of Engineering and Computer Science Applications (IJECSA)*, vol. 4, no. 2, pp. 81-90, Sep. 2025. doi: 10.30812/ijecsa.v4i2.5064.

## 1. INTRODUCTION

The eye is one of the human sensory organs that functions as the organ of vision [1]. One of the visual impairments that requires serious attention is blindness [2]. Various types of eye disorders include cataracts, glaucoma, and retinal diseases [3]. A cataract is a condition in which cloudiness occurs in the lens fibers or materials inside the capsule, making it difficult for light to reach the retina and thus disturbing vision [4, 5]. According to estimates by the World Health Organization (WHO), cataracts are responsible for nearly 50% of all global cases of blindness [6]. Globally, Indonesia ranks second in the number of cataract cases, following Ethiopia. However, people often do not realize the symptoms of cataracts, as it is difficult to distinguish between normal eyes and those in the early stages of cataract development [7]. Therefore, early detection of cataracts is crucial to prevent blindness [8]. Cataracts can be classified based on eye images using digital image analysis techniques [9]. Eye images obtained through medical imaging methods, such as fundus photography or slit-lamp imaging [10], contain important visual information that helps distinguish between normal eyes and those affected by cataracts. By extracting features from the images, such as brightness, texture, contrast, and lens opacity, the classification system can detect patterns that indicate the presence of cataracts. Using image processing techniques and classification algorithms based on machine learning or deep learning, cataract identification can be carried out automatically, quickly, and accurately [11]. This approach not only supports early detection but also reduces the reliance on manual examinations that require specialists. Therefore, developing image-based classification systems is a strategic step toward more efficient cataract diagnosis.

Deep learning technology, especially through the use of Convolutional Neural Networks (CNN), is one of the most commonly used classification methods for solving problems related to Object Detection and Image Classification [11, 12]. In a study by [11], eye disease classification was performed using CNN with the VGG16 architecture, achieving an accuracy of 82.63%. Another study by [10] classified cataracts based on fundus images using a CNN model and achieved an accuracy of 92%. A further study by [13] also used CNN for cataract classification and achieved an accuracy of 85%, while [14] applied CNN for classifying banana leaf diseases and obtained an accuracy of 92%. **The research gap** lies in the limited exploration of hyperparameter tuning, despite its recognized importance in enhancing model performance. Although previous studies have employed various optimizers such as Adam, RMSprop, SGD, and AdaGrad, few have thoroughly examined how proper adjustments to hyperparameters, such as the number of epochs, data split (training, validation, and testing), and optimizer selection, affect model accuracy and effectiveness in classification tasks. This indicates that hyperparameter tuning is a crucial factor for achieving optimal results. Therefore, this study addresses the gap by emphasizing the significance of hyperparameter adjustments to improve the model's ability to recognize patterns more accurately and efficiently. It demonstrates that selecting the right hyperparameter values, including the appropriate optimizer, plays a vital role in boosting model accuracy, **distinguishing** this research from previous work.

**This study aims** to evaluate the CNN model for early cataract detection by exploring the use of several optimization algorithms, including Adaptive Moment Estimation (Adam), Root Mean Square Propagation (RMSprop), Adaptive Gradient Algorithm (AdaGrad), and Stochastic Gradient Descent (SGD). These schemes include setting the number of epochs, the proportion of data division (training, validation, and testing), and using different optimizers. Through these tests, the goal is to find the best combination of parameters that can produce optimal model accuracy in the classification process. **The main contribution of this study** is to provide deeper insight into how the choice of optimizer affects model performance in classification tasks, particularly by comparing the Adam optimizer with others such as RMSprop, SGD with Momentum, and AdaGrad. The study shows that Adam generally delivers better and more consistent results in training, validation, and testing accuracy than the other optimizers, significantly contributing to selecting the right optimizer for different deep learning models.

## 2. RESEARCH METHOD

Figure 1 illustrates the flow of this study to make the research process easier to understand. The figure shows the steps taken to obtain the results of this research. The initial stage begins with data collection, followed by dataset processing, preprocessing, feature extraction, and then the model training and testing stages.

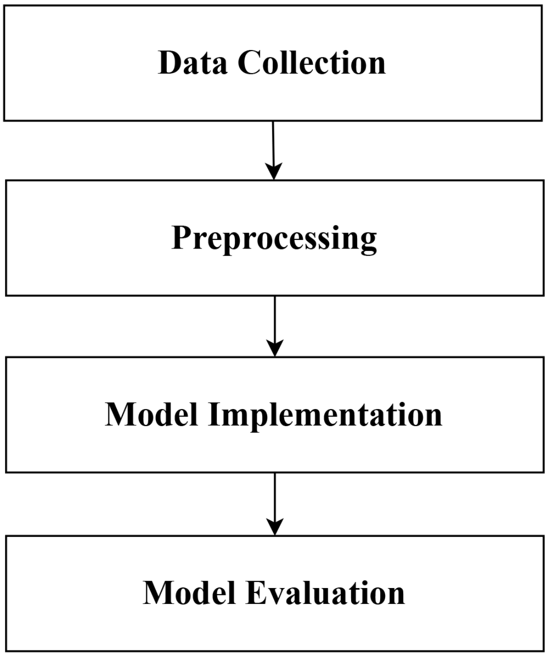
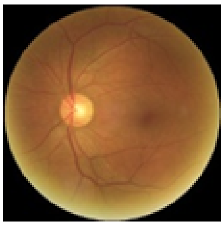



Figure 1. Research Flow

### 2.1. Data Collection

In this study, the data used to detect cataract disease consists of eye images. The secondary data was obtained from the Kaggle platform (https://www.kaggle.com/code/babajideabiola/eye-disease-classification). The dataset includes several images of eyes classified based on their health condition—normal eyes and eyes with cataracts. 1,433 images were divided into 719 images of cataract-affected eyes and 714 images of normal eyes. A sample of the dataset used in this study is shown in Table 1.

Table 1. Dataset sample

| No | Dataset Sample | | Total |
|---|---|---|---|
| 1 | Normal |  | 714 |
| 2 | Cataract |  | 719 |

Tables 2 and 3 show how a dataset of 1,433 image samples was divided into three main parts: training, validation, and testing, with different proportions for experimental purposes. In Table 2, the training data were split into three scenarios: 80%, 70%, and 60% of the total data, resulting in 1,146, 1,000, and 857 training samples, respectively. This division aimed to examine how the amount of training data affects model accuracy. Table 3 displays the data distribution for validation and testing, with proportions of 10%, 15%, and 20% each. For instance, when using 10% for validation and testing, there are 142 and 145 samples, respectively. These numbers vary according to the set proportions; for example, at 20%, there are 287 samples for validation and 289 for testing. This distribution strategy allows for a more comprehensive evaluation of the model under different data scenarios.

Table 2. Training data division

| | Number of Dataset | | |
|---|---|---|---|
| Data Type | 80% | 70% | 60% |
| Train | 1146 | 1000 | 857 |

Table 3. Division of validation and testing datasets

| | Number of Dataset | | |
|---|---|---|---|
| Data Type | 10% | 15% | 20% |
| Validation | 142 | 216 | 287 |
| Testing | 145 | 217 | 289 |

### 2.2. Preprocessing

Preprocessing is important in preparing the dataset and making the next processes easier. In this stage, the dataset is analyzed and processed before it goes into the feature extraction phase. The preprocessing steps carried out include resizing and image augmentation. The researcher resized the original images from 256x256 pixels to new dimensions of 150x200 pixels. This process involved adjusting the image scale while maintaining optimal proportions to ensure the images remained clear and informative after resizing. Figure 2 shows the result of the resizing process, where the original image size of 256x256 pixels was changed to 150x200 pixels. As is known, a large dataset is needed to achieve the best performance in deep learning. Since the variation of objects in this study is relatively limited, data augmentation was used to increase the diversity of the data. Data augmentation is a method of manipulating data without removing important elements. The augmentation processes include rescaling, shearing, rotation, zooming, filling, and flipping.

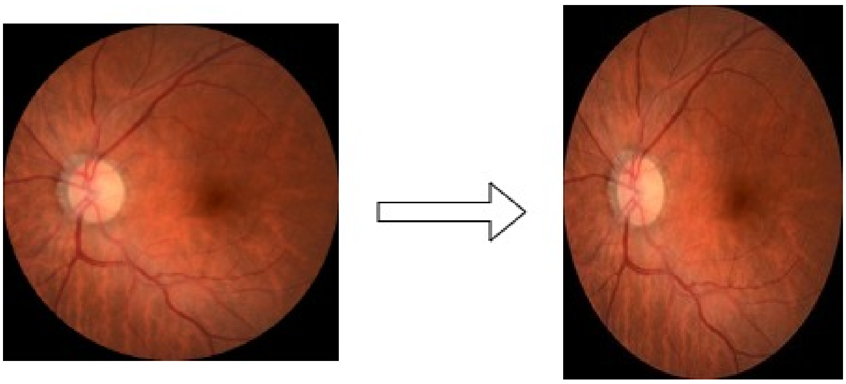


Figure 2. Resized sample image

### 2.3. Model implementation

To achieve high accuracy in eye image detection systems, cataract eye images must first be trained using a designed CNN model. Before training begins, the dataset is divided into several parts: training, validation, and testing data. The dataset is split using various ratios, including 80%:10%:10%, 75%:15%:15%, and 60%:20%:20%. This training process aims to help the model learn the unique features of each eye image, so it can determine which neurons should be activated when detecting the image. Additionally, different types of optimizers are used in this study to assess their impact on model performance. Optimizers are algorithms that update neural network weights during training so the model can reach convergence more quickly and accurately. The optimizers used include Adam, Adagrad, RMSprop, and SGD. After initializing the necessary parameters and hyperparameters, the next step is to define the CNN architecture for training. The designed CNN architecture is then trained to recognize patterns and characteristics in cataract eye images to accurately detect the presence of cataract disease. The details of the CNN model architecture used in this study are shown in Figure 3.

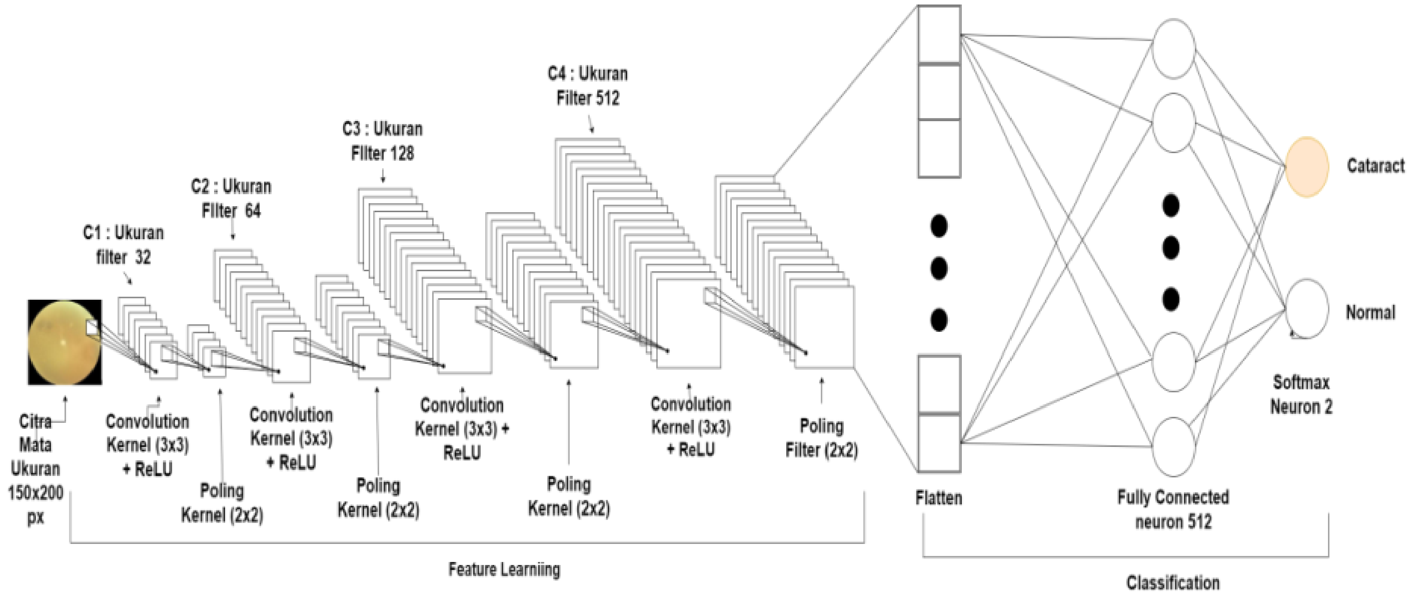



Figure 3. CNN Architecture Model

## 2.4. Model Evaluation

This study uses accuracy as the primary metric to evaluate the performance of the CNN method in detecting cataract eye conditions. Accuracy measures how correctly the model classifies eye images into the appropriate categories, either cataract or normal. A higher accuracy indicates better model performance in consistently recognizing visual patterns related to cataracts across training, validation, and testing data. The accuracy is calculated using Formula (1) [15–17].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (1)$$

# 3. RESULT AND ANALYSIS

One of the most important components for successful image-based disease detection is the quality of the training results. The training outcomes significantly affect the test performance. Testing was carried out using a simpler architecture consisting of 3 CNN layers. Various optimizers were used, including SGD, AdaGrad, RMSprop, and Adam, with different epoch settings such as 5, 15, 25, and 50. The dataset consisted of 1,433 images, divided into training, validation, and testing data with different ratios: 60%:40%, 70%:30%, and 80%:20%, for detecting cataracts in eye images. The following are the results of the dataset splits, epochs, and trials using several optimizers:

## 3.1. Adam Optimizer

The results in Table 4 show that using the Adam optimizer leads to a consistent increase in accuracy as the number of epochs grows. Training accuracy improves from about 73–77% at epoch 5 to 90–94% at epoch 50, depending on the proportion of training data (60%, 70%, 80%). This suggests the model is learning effectively over time. Validation accuracy also shows a significant increase, from 56–70% at the beginning to around 89–96% by epochs 25 to 50, indicating the model's ability to handle unseen data. Table 5 presents the model's accuracy on entirely new testing data after training. The testing accuracy remains high, ranging from

90% to 93%, depending on the test data proportion (10%, 15%, 20%). This consistent accuracy shows that the model performs stably and does not overfit, maintaining high results on previously unseen data. Overall, using the Adam optimizer in training improves model accuracy during training, validation, and testing. The stable improvement in accuracy across all stages shows the model can learn data patterns well and generalize effectively to new data, making Adam a suitable choice for optimizing model learning on this dataset.

Table 4. Training and validation accuracy results of the Adam optimizer

| Epoch | Training | | | Validation | | |
|---|---|---|---|---|---|---|
| | **80%** | **70%** | **60%** | **10%** | **15%** | **20%** |
| 5 | 74% | 77% | 73% | 57% | 70% | 56% |
| 15 | 85% | 84% | 83% | 92% | 90% | 73% |
| 25 | 86% | 88% | 86% | 91% | 96% | 89% |
| 50 | 92% | 94% | 90% | 92% | 93% | 92% |

Table 5. Adam optimizer testing accuracy results

| Test | | |
|---|---|---|
| 10% | 15% | 20% |
| 90% | 93% | 91% |

### 3.2. RMSprop Optimizer

The results in Table 6 show that using the RMSprop optimizer gradually increases training accuracy as the number of epochs increases. Training accuracy improves from 69–72% at epoch 5 to 85–88% at epoch 50, depending on the training data proportion. Validation accuracy increases, though not as significantly as training accuracy, from 64–71% at the beginning to 78–87% by epoch 50. This suggests that RMSprop can effectively train the model, although the improvement is not as rapid as with the Adam optimizer. Table 7 shows that the model's testing accuracy on new data remains relatively high and stable: 92% for 10% and 20% test data proportions, and slightly lower at 86% for 15%. This indicates that the RMSprop-trained model still generalizes well, although slight fluctuations depend on the test data proportion. Overall, RMSprop yields fairly good training and testing results with consistent accuracy improvements. However, compared to Adam, RMSprop tends to produce slightly lower validation and testing accuracy in some scenarios, suggesting that while effective, it may require further tuning of parameters or model structure to achieve optimal performance.

Table 6. Training accuracy results and validation of the RMSprop optimizer

| Epoch | Training | | | Validation | | |
|---|---|---|---|---|---|---|
| | **80%** | **70%** | **60%** | **10%** | **15%** | **20%** |
| 5 | 72% | 72% | 69% | 64% | 71% | 67% |
| 15 | 79% | 80% | 78% | 69% | 79% | 77% |
| 25 | 81% | 84% | 82% | 78% | 80% | 79% |
| 50 | 85% | 87% | 88% | 78% | 87% | 84% |

Table 7. Testing accuracy results using the RMSprop optimizer

| Test | | |
|---|---|---|
| 10% | 15% | 20% |
| 92% | 86% | 92% |

### 3.3. AdaGrad Optimizer

Table 8 presents the model's training and validation accuracy results using the AdaGrad optimizer with various training and validation data proportions. At epoch 5, training accuracy ranged from 70–72%, while validation accuracy was between 60–68%. As the epochs progressed to 50, training accuracy gradually improved to 83–86%, and validation accuracy increased to 80–84%. This shows that AdaGrad supports gradual learning, although its improvement rate is slower than optimizers like Adam. Table 9 shows the testing accuracy on previously unseen data, with results of 88% for a 10% test set, 86% for 15%, and 89% for 20%. While these results are fairly good, they are slightly lower than those obtained using Adam and RMSprop, indicating that AdaGrad can still generalize well, but not as effectively as the other two optimizers. Overall, AdaGrad provides a stable model with gradual accuracy improvements. However, its performance in training, validation, and testing is slightly lower than Adam and RMSprop, suggesting that alternative optimizers may be preferable for achieving better results, especially with more complex datasets and models.

Table 8. Training and validation accuracy results using the AdaGrad optimizer

| Epoch | Training | | | Validation | | |
|---|---|---|---|---|---|---|
| | **80%** | **70%** | **60%** | **10%** | **15%** | **20%** |
| 5 | 56% | 64% | 61% | 57% | 66% | 54% |
| 15 | 60% | 69% | 64% | 68% | 66% | 61% |
| 25 | 68% | 71% | 68% | 64% | 66% | 69% |
| 50 | 72% | 76% | 73% | 80% | 72% | 67% |

Table 9. Testing accuracy results using the AdaGrad optimizer

| Test | | |
|---|---|---|
| 10% | 15% | 20% |
| 80% | 73% | 73% |

### 3.4. Optimizer SGD

Table 10 shows that the SGD optimizer produced varied results during training and validation. Training accuracy increased from around 69–72% at epoch 5 to 72–91% by epoch 50, depending on the proportion of training data. A similar trend was seen in validation accuracy, which fluctuated but generally improved from 63–80% early on to 78–92% by epoch 50. Despite these improvements, epoch inconsistencies indicate that SGD training is more sensitive to parameter settings and data proportions. Table 11 presents the model's test accuracy on previously unseen data, ranging from 80–87% depending on the test data split (10%, 15%, 20%). These results show that models trained with SGD can still generalize fairly well, though not as effectively as those trained with Adam or RMSprop, which usually achieve over 90% accuracy. While SGD can improve accuracy over time, its results are less stable, suggesting that adaptive optimizers like Adam or RMSprop are generally better suited for achieving consistent and optimal performance.

Table 10. Training and validation accuracy results using the SGD optimizer

| Epoch | Training | | | Validation | | |
|---|---|---|---|---|---|---|
| | **80%** | **70%** | **60%** | **10%** | **15%** | **20%** |
| 5 | 69% | 72% | 70% | 80% | 63% | 64% |
| 15 | 79% | 80% | 83% | 77% | 75% | 81% |
| 25 | 68% | 85% | 84% | 64% | 78% | 82% |
| 50 | 72% | 89% | 91% | 80% | 78% | 92% |

Table 11. Testing accuracy results using the SGD optimizer

| Test | | |
|---|---|---|
| 10% | 15% | 20% |
| 80% | 82% | 87% |

Based on the results of various experiments, the best scheme using the Adam optimizer was found at 50 epochs with a dataset split of 70% for training and 30% for testing, resulting in 94% accuracy on training data, 93% on validation data, and 93% on testing data. For the RMSprop optimizer, the best scheme was also at 50 epochs with a 70%-30% dataset split, producing 87% accuracy on training data, 86% on validation data, and 87% on testing data. The best scheme using the AdaGrad optimizer was at 50 epochs with an 80%-20% dataset split, yielding 72% accuracy on training data, 80% on validation data, and 80% on testing data. The best result using the SGD optimizer was achieved at 50 epochs with a 60%-40% dataset split, reaching 91% accuracy on training data, 87% on validation data, and 92% on testing data. **The findings** of this research show that the CNN model with the Adam optimizer outperformed other optimizers under the 50-epoch and 70%-30% dataset split scheme. This model achieved 94% accuracy on training data, 93% on validation data, and 93% on testing data. These results are **consistent** with previous studies [18–21], which also found that Adam provides higher accuracy than other optimizers. Based on these findings, this model is considered the most optimal and will be used to detect eye images and identify potential cataract symptoms.

Adam performs better than other optimizers because it combines the advantages of Momentum and RMSprop techniques. By using the first and second moment estimates of gradients, Adam adaptively adjusts the learning rate for each parameter. This makes training more stable, faster to converge, and effective in handling sharp changes in gradients. Unlike AdaGrad, which slows learning over time, or SGD, which requires manual learning rate tuning, Adam maintains consistent accuracy even in early epochs and across different training data proportions. Adam is also more robust across various dataset sizes and distributions. The results in the table show that training, validation, and testing accuracy with Adam are generally high and stable, indicating good generalization ability. Compared to RMSprop and SGD, which show performance fluctuations, Adam consistently delivers the best results in almost all scenarios. This advantage makes Adam the preferred choice for training machine learning and deep learning models, especially when stability and efficiency are critical.

## 4. CONCLUSION

Based on the results of this study, it can be concluded that the Adam optimizer provides the best performance compared to RMSprop, AdaGrad, and SGD in terms of training, validation, and testing accuracy. Adam consistently maintains stability and improves accuracy across different proportions of training and validation data, making it more effective in handling the complexity and uncertainty of datasets. This advantage indicates that Adam is well-suited for various deep learning models requiring efficient training times without sacrificing result quality. The novelty of this study lies in testing and comparing four commonly used optimizers, such as Adam, RMSprop, AdaGrad, and SGD, on datasets with varying sizes and data proportions. The study highlights Adam's superiority in enhancing stability and achieving faster convergence compared to other optimizers, which often face challenges regarding stability and efficiency, particularly in the early epochs. Future studies should evaluate cataract data using various classification models and different parameter tuning combinations to develop a more effective classification model.

## REFERENCES

[1] S. Agustin, E. N. Putri, and I. N. Ichsan, "Design of A Cataract Detection System based on The Convolutional Neural Network," *Jurnal ELTIKOM*, vol. 8, no. 1, pp. 1–8, Jun. 2024. DOI: 10.31961/eltikom.v8i1.1019.

[2] V.-V. Nguyen and C.-L. Lin, "Enhancing Cataract Detection through Hybrid CNN Approach and Image Quadration: A Solution for Precise Diagnosis and Improved Patient Care," *Electronics*, vol. 13, no. 12, p. 2344, Jun. 2024. DOI: 10.3390/electronics13122344.

[3] A. B. Triyadi, A. Bustamam, and P. Anki, "Deep Learning in Image Classification using VGG-19 and Residual Networks for Cataract Detection," in *2022 2nd International Conference on Information Technology and Education (ICIT&E)*, Malang, Indonesia: IEEE, Jan. 2022, pp. 293–297. DOI: 10.1109/ICITE54466.2022.9759886.

[4] C.-J. Lai, P.-F. Pai, M. Marvin, H.-H. Hung, S.-H. Wang, and D.-N. Chen, "The Use of Convolutional Neural Networks and Digital Camera Images in Cataract Detection," *Electronics*, vol. 11, no. 6, p. 887, Mar. 2022. DOI: 10.3390/electronics11060887.

[5] T. Ganokratanaa, M. Ketcham, and P. Pramkeaw, "Advancements in Cataract Detection: The Systematic Development of LeNet-Convolutional Neural Network Models," *Journal of Imaging*, vol. 9, no. 10, p. 197, Sep. 2023. DOI: 10.3390/jimaging9100197.

[6] I. Santoso, A. M. Manurung, and E. R. Subhiyakto, "Comparison of ResNet-50, EfficientNet-B1, and VGG-16 Algorithms for Cataract Eye Image Classification," *Journal of Applied Informatics and Computing*, vol. 9, no. 2, pp. 284–294, Mar. 2025. DOI: 10.30871/jaic.v9i2.8968.

[7] T. D. Fikri, R. Sigit, D. M. Sari, and D. P. Trianggadewi, "Early Detection and Classification of Cataracts Using Smartphone Imagery Based on Support Vector Machine (SVM) and Certainly Factor Methods," in *2024 International Electronics Symposium (IES)*, Denpasar, Indonesia: IEEE, Aug. 2024, pp. 669–674. DOI: 10.1109/IES63037.2024.10665769.

[8] M. A. S. Yudono, F. F. Ridha, D. Mardiyana, F. Al-Ghozi, and A. Maulana, "Klasifikasi Katarak Berdasarkan Optic Disc Citra Fundus Smartphone: Perbandingan Ekstraksi Ciri Tekstur dan Metode Neural Network," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 12, no. 1, pp. 203–212, Feb. 2025. DOI: 10.25126/jtiik.20251219254.

[9] M. A. Alghozali, D. P. Pamungkas, and U. Mahdiyah, "Implementasi Metode Transformasi Wavelet Diskrit dengan K-Nearest Neighbor untuk Klasifikasi Penyakit Mata," *Nusantara of Engineering (NOE)*, vol. 7, no. 2, pp. 103–109, Oct. 2024. DOI: 10.29407/noe.v7i02.22889.

[10] R. Denandra, A. Fariza, and Y. R. Prayogi, "Eye Disease Classification Based on Fundus Images Using Convolutional Neural Network," in *2023 International Electronics Symposium (IES)*, Denpasar, Indonesia: IEEE, Aug. 2023, pp. 563–568. DOI: 10.1109/IES59143.2023.10242558.

[11] V. Wulandari and A. T. Putra, "Optimization of the Convolutional Neural Network Method Using Fine-Tuning for Image Classification of Eye Disease," *Recursive Journal of Informatics*, vol. 2, no. 1, pp. 54–61, Mar. 2024. DOI: 10.15294/rji.v2i1.73625.

[12] M. N. I. Muhlashin and A. Stefanie, "Klasifikasi Penyakit Mata Berdasarkan Citra Fundus Menggunakan YOLO V8," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 7, no. 2, pp. 1363–1368, Sep. 2023. DOI: 10.36040/jati.v7i2.6927.

[13] M. Maspaeni, B. Imran, A. Hidayat, and S. Erniwati, "Implementasi Machine Learning untuk Mendeteksi Penyakit Katarak menggunakan Kombinasi Ekstraksi Fitur dan Neural Network Berdasarkan Citra," *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 7, no. 2, pp. 232–251, Mar. 2025. DOI: 10.35746/jtim.v7i2.621.

[14] M. D. Pratama, R. Gustriansyah, and E. Purnamasari, "Klasifikasi Penyakit Daun Pisang menggunakan Convolutional Neural Network (CNN)," *Jurnal Teknologi Terpadu*, vol. 10, no. 1, pp. 1–6, Jul. 2024. DOI: 10.54914/jtt.v10i1.1167.

[15] L. G. R. Putra, K. Marzuki, and H. Hairani, "Correlation-based feature selection and Smote-Tomek Link to improve the performance of machine learning methods on cancer disease prediction," *Engineering and Applied Science Research*, vol. 50, p. 577 583, 2023. DOI: 10.14456/EASR.2023.59.

[16] C. M. Lauw, H. Hairani, I. Saifuddin, J. Ximenes Guterres, M. M. Huda, and M. Mayadi, "Combination of Smote and Random Forest Methods for Lung Cancer Classification," *International Journal of Engineering and Computer Science Applications (IJECSA)*, vol. 2, no. 2, pp. 59–64, Sep. 2023. DOI: 10.30812/ijecsa.v2i2.3333.

[17] H. Hairani, M. Janhasmadja, A. Tholib, J. Ximenes Guterres, and Y. Ariyanto, "Thesis Topic Modeling Study: Latent Dirichlet Allocation (LDA) and Machine Learning Approach," *International Journal of Engineering and Computer Science Applications (IJECSA)*, vol. 3, no. 2, pp. 51–60, Sep. 2024. DOI: 10.30812/ijecsa.v3i2.4375.

[18] A. I. Pradana and W. Wijiyanto, "Identifikasi Jenis Kelamin Otomatis Berdasarkan Mata Manusia Menggunakan Convolutional Neural Network (CNN) dan Haar Cascade Classifier," *G-Tech: Jurnal Teknologi Terapan*, vol. 8, no. 1, pp. 502–511, Jan. 2024. DOI: 10.33379/gtech.v8i1.3814.

[19] L. Syafa'ah, R. Prasetyono, and H. Hariyady, "Enhancing Qur'anic Recitation Experience with CNN and MFCC Features for Emotion Identification," *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, May 2024. DOI: 10.22219/kinetik.v9i2.2007.

[20] E. C. Seyrek and M. Uysal, "A comparative analysis of various activation functions and optimizers in a convolutional neural network for hyperspectral image classification," *Multimedia Tools and Applications*, vol. 83, no. 18, pp. 53 785–53 816, Nov. 2023. DOI: 10.1007/s11042-023-17546-5.

[21] H. Hairani and T. Widiyaningtyas, "Augmented Rice Plant Disease Detection with Convolutional Neural Networks," *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, vol. 8, no. 1, pp. 27–39, Feb. 2024. DOI: 10.29407/intensif.v8i1.21168.

**[This page intentionally left blank.]**