Available online at https://icsejournal.com/index.php/JCSE
Journal of Computer Science an Engineering (JCSE)
e-ISSN 2721-0251
Vol. 6, No. 1, February 2025, pp. 30-36

Employment of Convolutional Neural Networks in an Eye Disease Detection Application Leveraging Tensorflow.js

Zainal Arifin[1*], Firman Santoso[2], Adi Susanto[3]

[1,2,3]Departmen Information Technology at University of Ibrahimy, Situbondo, Indonesian
[1]zain.elibrahimy@gmail.com*; [2] firman4bi@gmail.com; [3] adisusanto@ibrahimy.ac.id
* corresponding author

ARTICLE INFO

Article History:
Received June 9, 2024
Revised July 15, 2024
Accepted February 12, 2025

Keywords:
Blindness and Vision Impairment
Convolutional Neural Network
Machine Learning
Classification
Tensorflow

Correspondence:
E-mail:
zain.elibrahimy@gmail.com

ABSTRACT

Vision impairment from cataract and glaucoma remains a leading global health challenge, with 2.2 billion people affected worldwide and Indonesia registering the highest prevalence in Southeast Asia and second highest globally (Ministry of Health, 2017–2030 Roadmap; WHO 2023). Early, accurate detection is essential to prevent irreversible blindness. In this study, we develop and evaluate a native-architecture Convolutional Neural Network (CNN) to classify cataract and glaucoma using three novel non-fundus ("real-eye") image subsets. Trained for 100 epochs on a balanced dataset, our CNN achieves 98.67 % accuracy, precision of 98.8 %, and recall of 98.5 % on a held-out test set. The resulting TensorFlow-saved model is deployed in the browser via TensorFlow.js, enabling real-time, platform-agnostic inference without specialized hardware. This work demonstrates that non-fundus imaging, combined with lightweight CNN design and web deployment, can provide a practical, high-accuracy tool for early eye-disease screening in low-resource settings.

1. Introduction

Vision impairment affects some 2.2 billion people worldwide, with cataract and glaucoma as the two leading causes of preventable blindness[1]. In Indonesia alone, 3 % of the population suffers from vision impairment the highest rate in Southeast Asia and second highest globally driven chiefly by cataracts and, to a lesser extent, glaucoma[2],[3]. Early, accurate screening is therefore critical to reduce the social and economic burden of vision loss[4][5].

Convolutional Neural Networks (CNNs) have shown great promise in medical image classification. AlexNet achieved 98.37 % accuracy on cataract versus normal fundus images but suffers from a large model size and slow inference [6], [7]. MobileNetV2 prioritizes speed and model compactness achieving 72 % accuracy yet sacrifices some classification performance [8]. Conversely, bespoke "native" CNN architectures can be highly efficient on constrained hardware, but often at the cost of reduced accuracy and flexibility [9], [10], [11].

Moreover, most prior work relies on high-resolution fundus photography, requiring specialized equipment and limiting accessibility in low-resource settings. Non-fundus ("real-eye") imaging via standard digital or smartphone cameras remains underexplored despite its potential for widespread screening [12], [13], [14].

To bridge this gap, we propose a lightweight, native-architecture CNN that classifies three non-fundus (real-eye) image classes—cataract, glaucoma, and normal—collected via standard digital cameras. Our pipeline resizes each image to 100×100 pixels, applies repeated convolution and MaxPooling layers, and uses ReLU and Softmax activations over 100 training epochs, achieving 98.67 % accuracy on a balanced 3 000-image dataset. Crucially, we export the trained model to TensorFlow.js, enabling in-browser inference without dedicated GPU hardware.

2. Method

The methodology of this research is structured into three distinct components: Research Method, Data Collection Method, and Data Analysis Method.

2.1 Research Method

The research methodology adopted in this study is an experimental method with an objective quantitative approach, involving the development and evaluation of a Convolutional Neural Network (CNN) model.

2.2 Data Collection Method

A comprehensive literature review served as the primary data collection method in this research, with the aim of tackling eye disease-related challenges and implementing deep learning techniques using classification methods to identify the optimal model based on training outcomes. The literature review encompassed a wide range of sources, including academic journals, electronic books, and relevant websites.

2.3 Data Analysis Method

Data analysis is employed to tackle the research problem by utilizing the gathered data and information. The data analysis methodology comprises the following stages:

Figure 1. Workflow of Data Analysis

2.3.1 Datasets

The data collection process involved gathering a total of 3000 images in .jpg format. These images comprised cataract, glaucoma, and normal eye images, acquired on February 28, 2024, from the Roboflow.com platform[15].

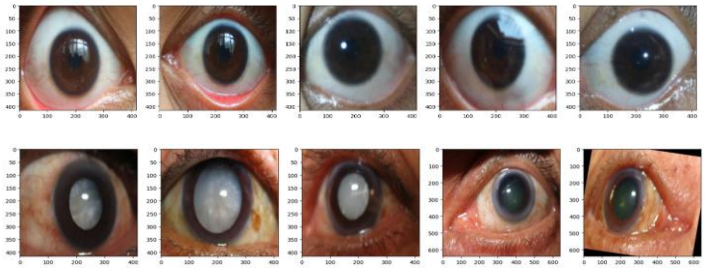

Figure 2. Normal and Cataract Eye Dataset

2.3.2 Preprocessing

Preprocessing is an essential phase in data preparation, aimed at addressing potential problems that could impair the effectiveness of data processing. This stage encompasses various techniques, including rescaling, shearing, zooming, and horizontal flipping.

2.3.3 Classification

Classification encompasses the convolution stage, utilizing image input. The convolution process involves applying filter values to a matrix, considering the three red, green, and blue channels of image pixels.

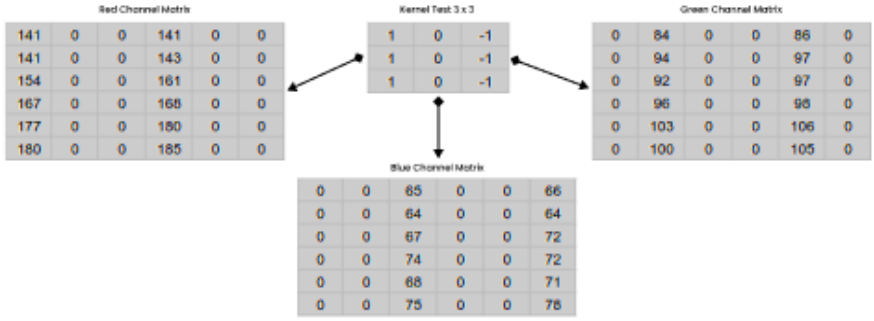

Figure 3. Image Pixel Values and Kernel Test Matrix

Figure 3 depicts the computational procedure for each channel (red, green, and blue) involving multiplication with a 3x3 kernel or filter. This iterative process involves shifting the kernel by 1 stride in each channel, generating the calculated values for each channel. The image pixel values are extracted using Google Colab to obtain the pixel values for each channel separately.

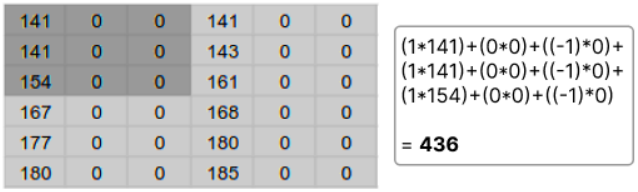

Figure 4. Convolutional Calculation with Filter 3x3

The convolution calculation is carried out on each channel matrix by multiplying a 3x3 matrix element-wise according to its rows and columns, followed by summation to produce a result value of 436. This process is executed with a stride of 1 from left to right.

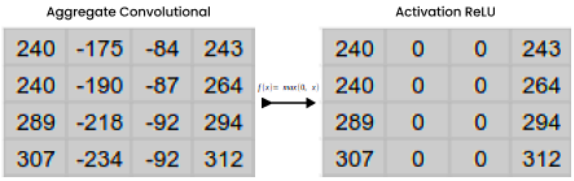

Figure 5. Activation of RGB Convolutional Values with ReLU

Following the convolution computation, the values from each channel are aggregated to yield the total RGB convolution value. Subsequently, ReLU activation is applied. During ReLU activation, all negative values are transformed into 0, producing the convolution outcome for figure 5.

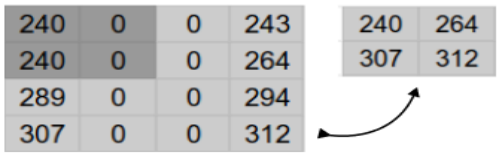

Figure 6. Pooling Layer

The max-pooling stage employs a 2x2 kernel and selects the maximum value at each stride position. This max-pooling operation utilizes a stride of 2, resulting in the output being fed into the dropout layer. The dropout phase serves to reduce the complexity of the constructed model by eliminating unused neural networks.

Figure 7. Flatten Layer

The flatten layer serves to convert a multidimensional matrix into a one-dimensional matrix or vector. This operation transforms the output of the 2x2x32 dropout layer into a new vector, as illustrated in Figure 7.

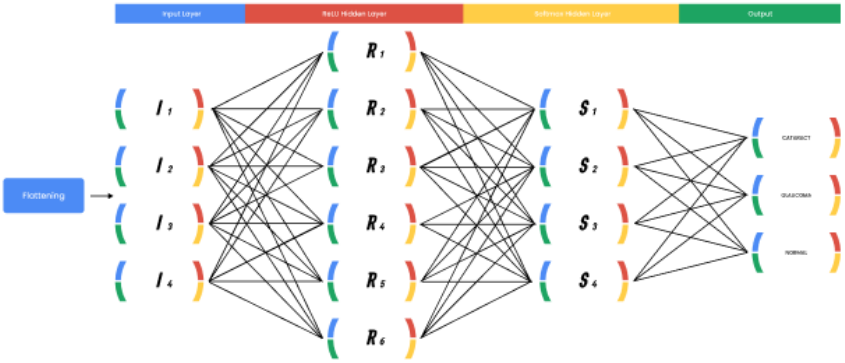

Figure 8. Illustration of Fully Connected Layers.

$$\sum_{i=1}^{N} J_i * w_{ij} = H_i \quad (1)$$

$$\sum_{i=1}^{N} I_i * v_{ij} = J_i \quad (2)$$

$$\sum_{i=1}^{N} H_i * x_{ij} = O_i \quad (3)$$

The following equation (1), (2), (3) is the hidden layer formula [16], [17]. The output values obtained from the flatten operation will be processed by the fully connected layer, utilizing computations from each of its hidden layers.

R_1 = (240*0,1) + (264*0,1) + (307*0,1) + (312*0,1) = 112,3

R_2 = (240*0,4) + (264*0,4) + (307*0,4) + (312*0,4) = 449,2

R_3 = (240*0,5) + (264*0,5) + (307*0,5) + (312*0,5) = 561,5

R_4 = (240*0,3) + (264*0,3) + (307*0,3) + (312*0,3) = 336,9

The first hidden layer undergoes multiplication with distinct weight values, yielding the values from the preceding computation. Subsequently, the results of this hidden layer calculation are further multiplied using different weight values to produce the values S_1, S_2, S_3, S_4 which constitute the next hidden layer.

S_1 = (112,3*0,3) + (449,2*0,3) + (561,5*0,3) + (336,9*0,3) = 437,97

S_2 = (112,3*0,5) + (449,2*0,5) + (561,5*0,5) + (336,9*0,5) = 729,95

S_3 = (112,3*0,2) + (449,2*0,2) + (561,5*0,2) + (336,9*0,2) = 291,98

S_4 = (112,3*0,4) + (449,2*0,4) + (561,5*0,4) + (336,9*0,4) = 583,96

Every neuron in S_1, S_2, S_3, S_4 will undergo multiplication with distinct weights once more, leading to the generation of the values Cataract, Glaucoma, and Normal.

Cataract = (437,97*0,5) + (729,95*0,5) + (291,98*0,5) + (583,96*0,5) = 1021,93

Glaucoma = (437,97*0,3) + (729,95*0,3) + (291,98*0,3) + (583,96*0,3) = 613,158

Normal = (437,97*0,1) + (729,95*0,1) + (291,98*0,1) + (583,96*0,1) = 204,386

The subsequent stage entails computing the softmax activation employing the exponential formula: Cataract divided by the sum of cataract + glaucoma + normal. This procedure is executed thrice, once each for cataract, glaucoma, and normal.

$$s(o_i) = \frac{e^{o_i}}{\sum_{j=1}^{n} e^{o_j}} \quad (4)$$

$$s(o_1) = \frac{e^{o_i}}{\sum_{j=1}^{n} e^{o_j}} \qquad s(Cataract) = \frac{e^{1021,93}}{e^{1021,93} + e^{613,158} + e^{204,386}} \qquad s(o_1) = 0,5$$

$$s(o_2) = \frac{e^{o_i}}{\sum_{j=1}^{n} e^{o_j}} \qquad s(Glaucoma) = \frac{e^{613,158}}{e^{1021,93} + e^{613,158} + e^{204,386}} \qquad s(o_2) = 0,3$$

$$s(o_3) = \frac{e^{o_i}}{\sum_{j=1}^{n} e^{o_j}} \qquad s(Normal) = \frac{e^{204,386}}{e^{1021,93} + e^{613,158} + e^{204,386}} \qquad s(o_3) = 0,1$$

$s(O_1) + s(O_2) + s(O_3) = 1$

0,5 + 0,3 + 0,1 = 1

3. Results and Discussion

Subsequent to achieving uniform image sizes, the image data distribution will advance to the classification stage. This stage in the research utilizes several parameters outlined in Table 1.

Table 1. Hyperparameters Employed

Params			
Image Dimensions	***Epoch***	***Batch Size***	***Optimizer***
100 x 100px	100	32	RMSprop

Utilizing the parameters employed in conjunction with the Native CNN architecture and the implemented model, the outcomes are depicted in Figure 9.

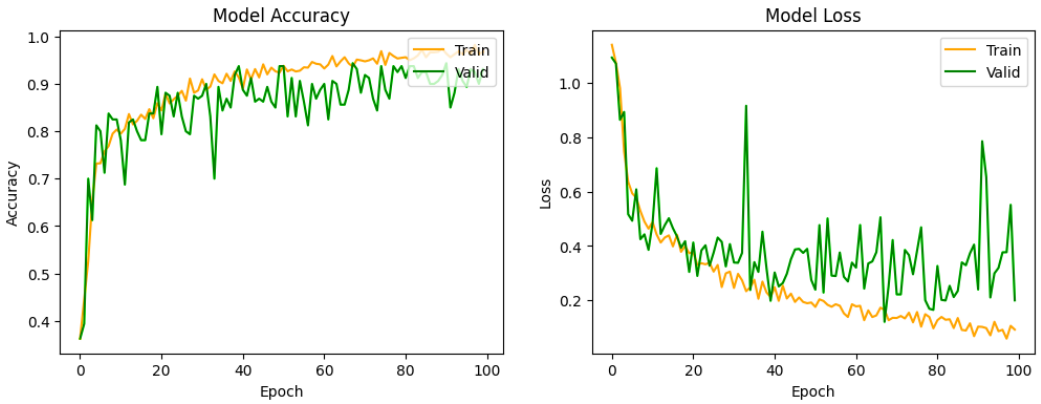

Figure 9. Accuracy and Loss Trajectory

Utilizing Convolutional Neural Network (CNN) for eye disease classification, a remarkable accuracy of 98.67% is attained, as illustrated in Figure 9.

The following equation (4) is the exponential formula for softmax activation. Consequently, the weight value corresponding to the probability of cataract emerges as the largest, at 0.5, implying that the input image depicts a cataract.

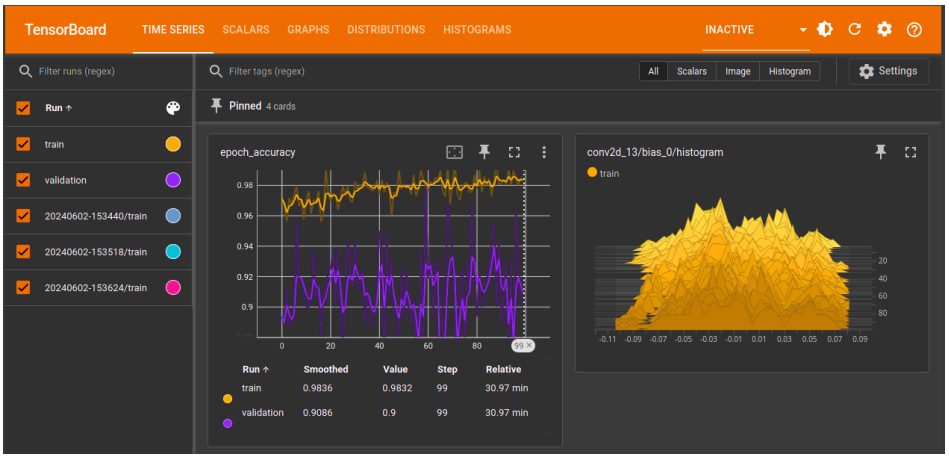

Figure 10. Employing TensorBoard for Model Monitoring

An evaluation of the classification algorithm was conducted using a sample dataset of 10 instances. The results demonstrated a classification testing accuracy of 98.67%. The saved model will be deployed as a web application utilizing TensorFlow.js.

4. Conclusion

This study investigates eye disease classification using the Convolutional Neural Network (CNN) method with a native CNN architecture. The study's novel contribution lies in the utilization of three non-fundus image classes: cataract, glaucoma, and normal. The preprocessing stage involves resizing images to 100x100 pixels. The CNN model is implemented using 100 epochs for its hyperparameters. The achieved classification accuracy is 98.67%.

References

[1] World Health Organization (WHO). "J Vision Impairment and Blindness,". Accessed: May 23, 2024. [Online]. Available: https://www.who.int/news room/fact-sheets/detail/blindness-and-visual-impairment

[2] Kemenkes. "Roadmap of Visual Impairment Control Program in Indonesia 2017 - 2030"., 2018. Accessed: May 24, 2024. [Online]. Available:

https://p2ptm.kemkes.go.id/uploads/VHcrbkVobjRzUDN3UCs4eUJ0dVBndz09/2018/04/Roadmap_of_Visual_Impairment_Control_In_Indonesia_2017_2030.pdf

[3] E. Y. Wang et al., "Global Trends in Blindness and Vision Impairment Resulting from Corneal Opacity 1984–2020," Ophthalmology, vol. 130, no. 8, pp. 863–871, 2023, doi: 10.1016/j.ophtha.2023.03.012.

[4] K. S. Naidoo et al., "Potential Lost Productivity Resulting from the Global Burden of Myopia," Ophthalmology, vol. 126, no. 3, pp. 338–346, 2019, doi: 10.1016/j.ophtha.2018.10.029.

[5] N. A. Brown and A. R. Hill, "Cataract: the relation between myopia and cataract morphology.," British Journal of Ophthalmology, vol. 71, no. 6, pp. 405–414, Jun. 1987, doi: 10.1136/bjo.71.6.405.

[6] F. N. Cahya, N. Hardi, D. Riana, and S. Hadiyanti, "Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)," SISTEMASI, vol. 10, no. 3, p. 618, Sep. 2021, doi: 10.32520/stmsi.v10i3.1248.

[7] R. Indraswari, W. Herulambang, and R. Rokhana, "Deteksi Penyakit Mata Pada Citra Fundus Menggunakan Convolutional Neural Network (CNN)," tc, vol. 21, no. 2, pp. 378–389, May 2022, doi: 10.33633/tc.v21i2.6162.

[8] I. Wulandari, H. Yasin, and T. Widiharih, "Klasifikasi Citra Digital Bumbu Dan Rempah Dengan Algoritma Convolutional Neural Network (CNN)," J.Gauss, vol. 9, no. 3, pp. 273–282, Aug. 2020, doi: 10.14710/j.gauss.v9i3.27416.

[9] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification," Artificial Intelligence in Agriculture, vol. 9, pp. 22–35, 2023, doi: 10.1016/j.aiia.2023.07.001.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[11] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv, 2017. doi: 10.48550/ARXIV.1704.04861.

[12] D. Ghimire, D. Kil, and S. Kim, "A Survey on Efficient Convolutional Neural Networks and Hardware Acceleration," Electronics, vol. 11, no. 6, p. 945, Jan. 2022, doi: 10.3390/electronics11060945

[13] M. Capra, B. Bussolino, A. Marchisio, M. Shafique, G. Masera, and M. Martina, "An Updated Survey of Efficient Hardware Architectures for Accelerating Deep Convolutional Neural Networks," Future Internet, vol. 12, no. 7, p. 113, Jul. 2020, doi: 10.3390/fi1207011

[14] P. Dhilleswararao, S. Boppu, M. S. Manikandan and L. R. Cenkeramaddi, "Efficient Hardware Architectures for Accelerating Deep Neural Networks: Survey," in IEEE Access, vol. 10, pp. 131788-131828, 2022, doi: 10.1109/ACCESS.2022.3229767

[15] Roboflow.com, "Roboflow: Image dataset platform," Feb. 28, 2024. [Online]. Available: https://roboflow.com.

[16] J. Heaton, "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning," Genetic Programming and Evolvable Machines, vol. 19, no. 1–2, pp. 305–307, Oct. 2017, doi: 10.1007/s10710-017-9314-z.

[17] S. L. Brunton and J. Nathan Kutz, "Neural Networks and Deep Learning," Cambridge University Press eBooks, pp. 195–226, Feb. 2019, doi: 10.1017/9781108380690.007.