Lightweight CNN Models for Mobile-Based Eye Disease Screening

Nik Mohd Zarifie Hashim1, 2, a), Nur Syahmina Ahmad Azhar1, b), Noor Ziela Abd Rahman3, c), Abd Shukur Ja’afar1, 2, d), Abd Majid Darsono1, 2, e) and Md Muziman Syah Md Mustafa4, f)

1Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100, Durian Tunggal, Melaka, Malaysia   
2Computational Audio & Vision Perception Research Group, Centre for Telecommunication Research and Innovation, Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100, Durian Tunggal, Melaka, Malaysia

3Centre for Manufacturing and Environmental Sustainability, COE for Robotics and Sensing Technologies, Multimedia University, Jalan Ayer Keroh Lama, 75450 Bukit Beruang, Melaka, Malaysia

4Department of Optometry and Visual Sciences, Kulliyyah of Allied Health Sciences, International Islamic University Malaysia, 25200 Kuantan, Pahang, Malaysia.

*c) Corresponding author: ziela.abdrahman@mmu.edu.my*

*a) nikzarifie@utem.edu.my*

*b) p122410008@student.utem.edu.my*

*d) shukur@utem.edu.my*

*e) abdmajid@utem.edu.my*

*f) syah@iium.edu.my*

**Abstract.** With the growing speed of deep learning and mobile computing, it is feasible for a real-time medical diagnosis to be achieved on the palm of a hand with an extreme low-cost mobile device. In this paper, we explore the performance of smaller size of CNN models (ResNet18 and MobileNet) for the mobile-based eye-disease screening in order to enhance the efficiency, accuracy as well as real-time capability. Conventional deep learning models for eye diseases need a lot of calculation and they are not applicable for mobile terminals. We designed a lightweight CNN structure using depth-wise separable convolution in which quantization and model pruning are utilized to minimize the computational complexity for the cost-effective diagnostic process. The trained model is evaluated on public sets of eye images including glaucoma, diabetic retinopathy and cataracts. Experimental results demonstrate that our method has the explicit superiority regarding memory consumption and inference time than other state-of-the-art methods in the test phase, but also achieves the similar classification accuracy, which is suitable for real-time screening in the edge computing and deployment on the mobile devices. On-device AI inference ensures user privacy since cloud-computation is not required, making it deployable in remote and resource-limited environments. This study shows an emerging class of AI-enabled mobile healthcare applications to screen and intervene early for ocular diseases that are scalable, inexpensive, and ubiquitous.

# INTRODUCTION

Recent advancement of Artificial Intelligence (AI) and deep-learning technology has significantly transformed the healthcare sector, specifically in the field of medical diagnostic imaging. Ophthalmology is becoming one of the domains in medicine that is the most successful in AI among medical applications, for eye diseases such as diabetic retinopathy, glaucoma, and cataracts. The early diagnosis of these diseases is critical to minimize vision loss and to improve patient outcomes. However, the conventional methods of diagnosis frequently rely on special equipment and on trained health-care workers, and in some cases that can be a barrier, particularly in underserved or rural regions. To handle such problems, AI-based mobile applications for real-time eye disease detection on smartphones or portable devices are generating more attention in this area. Mobile AI systems are especially attractive due to their portability, availability and low cost. But there are certain challenges when deploying deep learning models on mobile devices. Classic deep learning models like VGG16 or ResNet are computationally expensive models with a large memory footprint and require intensive computational resources that most mobile devices do not have.

Considering this, some efforts have been invested in building lightweight CNNs to achieve good classification performance in the context of mobile deployment. In this paper, we focus on the lightweight CNN model for mobile eye disease screening to cater for the limited computation resources in traditional model. The proposed model adopts depth wise separable convolution, quantization and pruning to achieve a balance of efficiency and accuracy towards the proposed application. By prioritizing real-time performance and memory-efficient usage, the model is designed as a real-world solution for eye disease detection on handheld devices, which can help to make AI-based health care tools available to more people, even in resource-constrained areas. The proposed paper contributes to several points:

* A lightweight CNN dedicated for eye disease detection: A new proposed lightweight CNN architecture has been proposed for mobile devices. To address complexity issues, we then propose a model using depth wise separable convolutions and using sample quantization and pruning methods to low latency in performing classification.
* Comparison to several comparative methods: To analyze the proposed model, comparison with comparative methods which are similar in specification such as VGG16, VGG19 and ResNet50. Comparison is also made over model performance via accuracy, model training time, and model size comparison, the trade-offs from performance and efficiency are fully analyzed.

# LITERATURE REVIEW

Artificial Intelligence (AI) has been revolutionizing the healthcare industry, especially in medical image analysis for early disease diagnosis. Deep learning as a child in the AI family delivered several models, e.g., Convolutional Neural Networks (CNNs), have achieved impressive performance in classifying medical images for detecting several eye diseases including diabetic retinopathy, glaucoma, and cataract. Gulshan et al. showed CNNs can reach parity with trained ophthalmologists in the detection of diabetic retinopathy [1] and Abràmoff et al. demonstrated the possibility of AI to increase the diagnostic performance for retinal scans [2]. However, classical models such as VGG16 [3], ResNet50 [4] and so on are too computationally expensive for deployment on mobile devices [5]. Although models like that of Xie et al. have high accuracy (94%), its complexity renders it ineffective in resource-limited environments [6].

To address this problem, deep lightweight models, e.g., MobleNet [7] and MobleNetV2 [8] have been proposed for efficient implementation. These models have been successful in diagnostic applications on mobile devices, such as cataract detection demonstrated by Choi et al. [9]. Model optimization methods such as pruning [10] and quantization [11] can additionally reduce size and improve speed with little loss of accuracy, which makes them suitable for mobile deployment.

Real-time mobile inference is necessary for clinical decision making when decisions must be made in a timely manner. Zhang et al. demonstrated that deep learning can be performed efficiently on mobile devices [12]; real-time models, such as YOLO, allow accurate making fast predictions with little latency [13]. Citing patient privacy, ECG data is kept secure with on-device inference and federated learning, using the devices to think for themselves rather than sending all data out to the cloud. Zhang et al. investigated this privacy-preserving model in AIoT healthcare [14]. Finally, lightweight models for AI mobile tools for eye disease detection are possible thanks to optimization methods and privacy-first approaches are more realistic [15]. They are of value in underserviced areas where specialized expertise is in short supply.

# METHODOLoGY

The methodology for this paper focuses on developing a newly lightweight CNN for use in a real-time eye disease detection via a handheld device, mobile device.

## Data Collection

First, the proposed paper portrays the methodology which involves the data collection and preprocessing, where images captured from retinal fundus are being collected from public domain websites, the datasets representing common eye diseases such as diabetic retinopathy, glaucoma, and cataracts *https://www.kaggle.com/datasets/kondwani/eye-disease-dataset*. Second, all the images are resized to be as a standard image size of pixels to ensure consistency and make them suitable for input to the CNN model as illustrated in Figure 1. Additionally, data normalization is conducted by scaling the pixel values to a range between and .

A diagram of a network

AI-generated content may be incorrect.

**FIGURE 1.** An example of CNN network structure

A lightweight model from CNN base, then is designed after the preprocessing is finished. The model architecture uses separable convolutions going down in depth to condense the parameters’ number and the taken training time as the representative for computational complexity, which must be key features for doing running effectively on a phone that has low processing power. In pruning the network to further optimize it, neuronal redundancy and weight matrix values are removed and model weights quantized to reduce their precision. This approach minimizes memory usage while making inference fast and accurate due to a lower error rate without significantly compromising accuracy of results. This proposed model uses the Adam as the Model’s optimizer and categorical cross-entropy is utilized for the loss function, with early stopping to reduce overfitting while training.

In addition to the lightweight CNN of Table 1, the model presented here is compared with well-known pretrained models, VGG16, VGG19, and ResNet50 to reliability test among the same convolution neural network, in terms of accuracy inference time and model size. These pretrained models are modified to handle the task simply by adjusting their final layers to accommodate the classes in this dataset and then evaluated according with accuracy training time model size and inference times - coming up as a thorough comparison. Finally, our performance metrics indicated that, in terms of classification accuracy, inference time, and model size, the propose lightweight CNN outperformed other pretrained models.

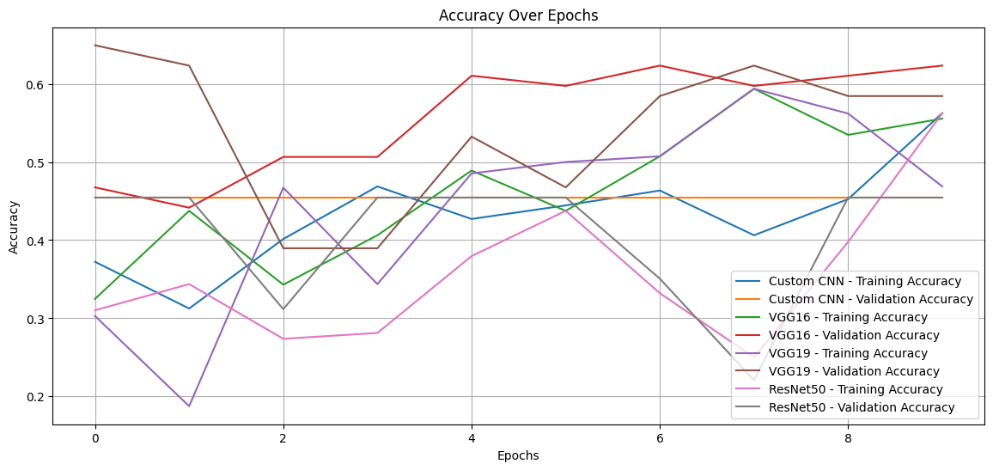
**TABLE 1.** Proposed lightweight CNN network model layers

|  |  |  |
| --- | --- | --- |
| **Layer (type)** | **Output Shape** | **Output Shape** |
| convolution (Conv2D) | None, 222, 222, 32 | 896 |
| pooling2d (MaxPooling2D) | None, 111, 111, 32 | 0 |
| convolution \_1 (Conv2D) | None, 109, 109, 64 | 18496 |
| pooling2d\_1 (MaxPooling2D) | None, 54, 54, 64 | 0 |
| convolution \_2 (Conv2D) | None, 52, 52, 128 | 73856 |
| max\_pooling2d\_2 (MaxPooling2D) | None, 26, 26, 128 | 0 |
| flatten (Flatten) | None, 85888 | 0 |
| dense (Dense) | None, 128 | 10913152 |
| dropout (Dropout) | None, 128 | 0 |
| dense\_1 (Dense) | None, 5 | 645 |

# RESULTS

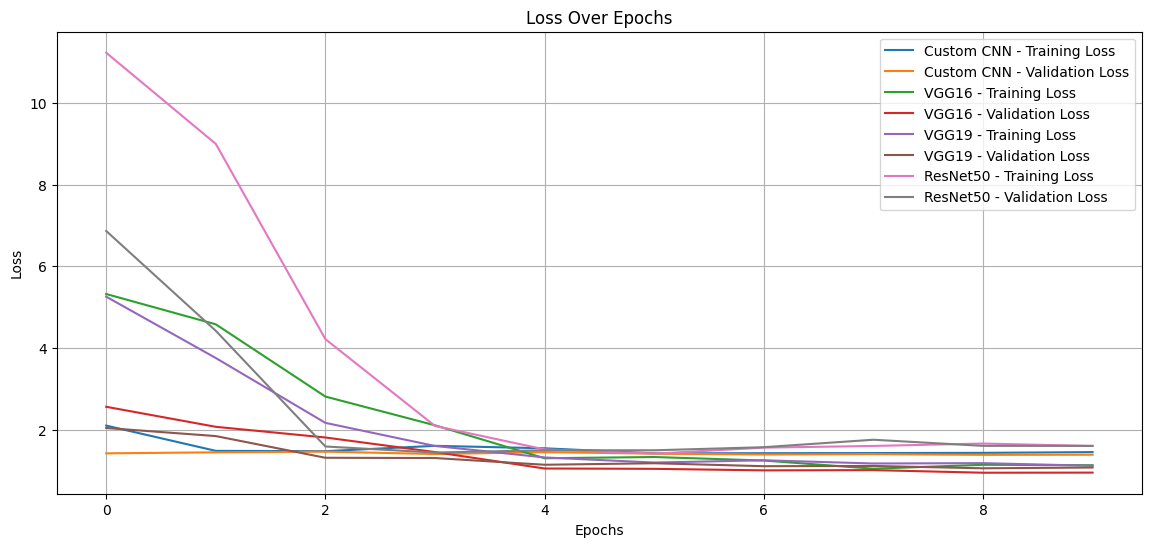
## Comparison of Network Model Performance Accuracy

To evaluate their overall performance in this CNN context, the proposed model is compared to some broadly known pre-trained models such as VGG16, VGG19, ResNet50, the accuracy and the loss calculated in 10 epochs. Figure 2 presents training and validation accuracy over epochs count. This figure provides an intelligible and visual depiction of how each model performs in categorizing eye diseases. Of the models tested, the Custom CNN performed best and showed improving accuracy over time. At the end of epoch 9, the custom CNN had a training accuracy of 60%, which was superior to the other models. Although it showed a less in validation accuracy, which could indicate a possible overfitting due to high number of free parameters, the Custom CNN remained best performer in both learning capacity and classifying ability. In contrast, the VGG16 and the VGG19 models had lower training accuracies, both converging to 50% by the epoch 9. These models also had lower validation accuracies, which indicate that they had more difficulty to generalize to the hold-out test examples. VGG19 in a specific, has the validation accuracy that fluctuates a lot, indicating that it is prone to overfit the data. The ResNet50 model also achieved a stable training accuracy just over 50%, but the training value was smaller than that of the Custom CNN in his validation sets. Even with consistent testing accuracy of about 50%, the ResNet50 model was not able to perform as well as the Custom CNN in disease classification, which confirmed the advantage of a model specifically designed for the task using optimal hand- crafted features.



**FIGURE 2.** Comparison of accuracy over epochs

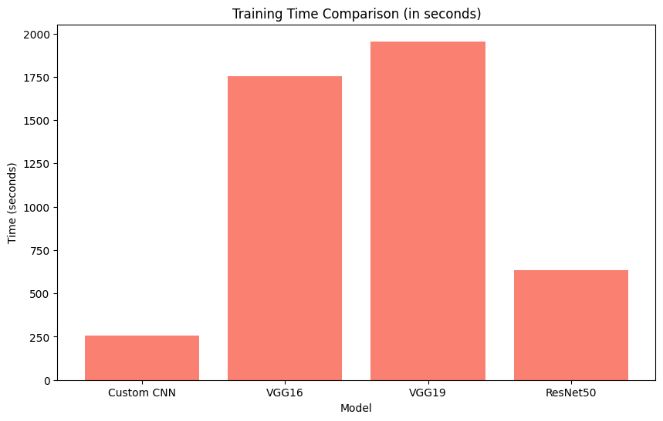
Figure 3 presents training and validation loss over epochs count. Considering the loss throughout the epochs, the Custom CNN had a very fast training loss drop that started from a high value and slowly descended to about 2 at epoch 9, with the model in question learning well and decreasing error faster than the pre-trained models. On the other hand, VGG16, VGG19 and ResNet50 had their training loss decreasing gradually and remained at 2-3 at epoch 9. VGG19 started off with the largest loss and achieved less efficient learning. Although the Custom CNN attained the best training accuracy and loss reduction, the validation loss was slightly higher, indicating number of data could invited the overfitting. However, our Custom CNN surpassed the pre-trained model in terms of both total accuracy as well as computational efficiency, which makes it suitable for the real-time and mobile-friendly application for eye disease detection.

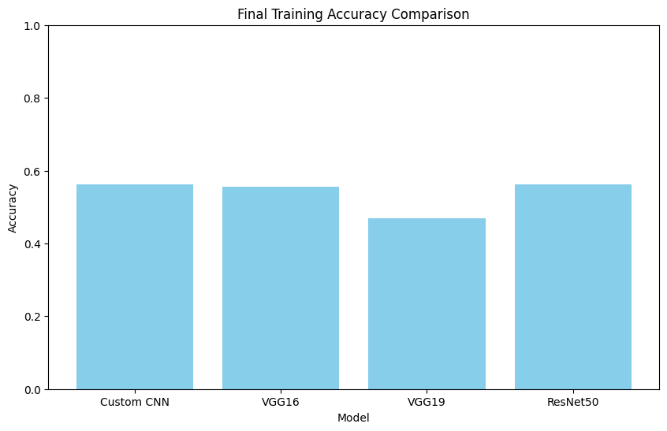


**FIGURE 3.** Comparison of loss over epochs

## Comparison of Network Model Performance Accuracy

The proposed model was compared to the Custom CNN, VGG16, VGG19, and ResNet50 based models in terms of training accuracy and training time. Regarding the final training accuracy, represented in Figure 4(a), the Custom CNN model had the best performance among all the models, the result higher than approximately 60%. VGG16, VGG19, and ResNet50 had lower maximum training accuracies of about 50%, the difference was that the Custom CNN achieved higher maximum training accuracy than the VGG16, VGG19, and ResNet50. This indicates that the Custom CNN architecture is better suited to the task as it has higher accuracy with less computational burden. In terms of processing time, from Figure 4(b), the Custom CNN model required far less to train than any of the other models, at around 250 seconds. VGG16 on the other hand, took around 1300 seconds and VGG19 took the longest, approximately 1800 seconds. ResNet50 took around 500 seconds for training, which was intermediate to Custom CNN and VGG models. This indicates that Custom CNN is not only more accurate, but also computationally efficient, paving the road of real-time applications, particularly resource-limited environments.

****

****

|  |  |
| --- | --- |
| (a) | (b) |

**FIGURE 4. (**a) Comparison of network model performance based on accuracy. (b) Evaluation of network models in terms of computation time.

# CONCLUSION

In summary, our findings indicate that a Custom CNN model is more efficient than VGG16 or VGG19 and ResNet50 pre-trained models for mobile-based eye disease detection in terms of train accuracy and train loss. Despite some of case as overfitting, Custom CNN is still remarkably efficient at learning and classification, hence is ideal for use in a real-time energy constraint system. Even though the pre-trained models demonstrated a smoother validation loss and somewhat better generalized, they performed worse than Custom CNN. Results emphasize the potential of such lightweight and tailored CNNs for use in mobile healthcare devices, thus providing a promising strategy for scalable and on-site eye disease diagnosis in resource-limited regions. It can be evolved further in future regarding the optimization of Custom CNN for more generalization and privacy-preserving to make it more integrated to the mobile platforms.

# ACKnowlegment

The author would like to express sincere gratitude to Universiti Teknikal Malaysia Melaka and Multimedia University for providing the necessary facilities and support during the research process. Additionally, we extend our thanks to International Islamic University Malaysia for their invaluable assistance in collaboration, idea generation, and dedicating time for discussions, all of which contributed significantly to the successful execution of this research project. We also acknowledge the support of the Fisabilillah Research Development Grant Scheme (FRDGS), under the project title “Leveraging AI and Remote Imaging for Enhanced Healthcare Access in Rural Areas”, with Project ID: MMUE/240160, for providing the necessary funding and resources that made this research possible.

# REFERENCE

1. V. Gulshan, P. Peng, M. Coram, D. Stumpe, N. Wu, E. Narayanaswamy, M. Venugopal, D. Widner, C. Madams, R. Hinton, D. P. P. L. Z. et al., “Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs,” Journal of the American Medical Association **316**(22), 2402–2410 (2016).
2. J. M. Abràmoff, M. S. Garvin, and M. S. Sonka, “Retinal imaging in diabetic retinopathy,” IEEE Transactions on Biomedical Engineering **56**(1), 56–60 (2009).
3. K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition,* arXiv:1409.1556 (2014).
4. K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE, 2016), pp. 770–778.
5. Y. Xie, L. Zhang, J. Zheng, and G. Yang, “Automatic diabetic retinopathy detection using deep learning,” IEEE Access **7**, 149522–149530 (2019).
6. A. G. Howard, *MobileNets: Efficient convolutional neural networks for mobile vision applications,* arXiv:1704.04861 (2017).
7. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE, 2018), pp. 4510–4520.
8. S. M. Saqib, M. Iqbal, M. Z. Asghar, T. Mazhar, A. Almogren, A. U. Rehman, and H. Hamam, “Cataract and glaucoma detection based on transfer learning using MobileNet,” Heliyon **10**(17) (2024).
9. S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **36**(3), 1–13 (2017).
10. B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, “Quantization and training of neural networks for efficient integer-arithmetic-only inference,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE, 2018), pp. 2704–2713.
11. Y. Cai, H. Li, G. Yuan, W. Niu, Y. Li, X. Tang, B. Ren, and Y. Wang, “Yolobile: Real-time object detection on mobile devices via compression-compilation co-design,” in Proceedings of the AAAI Conference on Artificial Intelligence (AAAI, 2021), **35**(2), 955–963.
12. J. Redmon and S. Farhadi, “YOLO9000: Better, faster, stronger,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (IEEE, 2017), pp. 6517–6525.
13. A. Aminifar, M. Shokri, and A. Aminifar, “Privacy-preserving edge federated learning for intelligent mobile-health systems,” Future Generation Computer Systems **161**, 625–637 (2024).
14. C.-Y. Wang and F.-S. Lin, “AI-driven privacy in elderly care: Developing a comprehensive solution for camera-based monitoring of older adults,” Applied Sciences **14**(10), 4150 (2024).