been rapidly developed due to their flexibility and scalability to be adopted in several fields for modeling real-world applications like object detection, image classification, etc. However, their high accuracy incurs intensive computations. Therefore, it is crucial to carefully choose a suitable computer platform and implementation methodology for CNN network architectures while achieving increased efficiency. Parallel architectures are prevalent in CNN implementation. Herein, we present a new Single Instruction Multi Data (SIMD) parallel implementation of the proposed CNN to speed up the execution process and make it suitable to deploy on low-cost, low-power consumption platforms. The proposed implementation produces an improved model of deep CNN executable on a cost-efficient platform and portability to work autonomously with multi-core processing units while maintaining working accuracy. Raspberry Pi 3 B is a low-power target device for implementing our model. The proposed approach is characterized by high diagnostic accuracy of up to 96.35 % while incurring power consumption of 3.65 Watts, achieving power reduction between 19.17 % and 68.45 % compared to the prior work. Meanwhile, it has a fine inference time for the selected platform. The outstanding results of this study reflect the success of employing parallel architectures to utilize the quad courses of the ARM processor on the target platform. The presented model can be an efficient medical assistant to provide automated detection and diagnosis for myopia ocular disease. Thus, it can be a promising healthcare toolkit that reduces the effort of the medical staff and increases the quality of the provided medical services for myopia patients

Keywords: CNN, multi-core, cost-effective, platform, prediction, myopia, ocular, ODIR, AIoT, SIMD

UDC 519
DOI: 10.15587/1729-4061.2023.289007

SIMD IMPLEMENTATION OF DEEP CNNS FOR MYOPIA DETECTION ON A SINGLE-BOARD COMPUTER SYSTEM

Mamoon A Al Jbaar
Corresponding author
Master of Science in Computer Engineering,
Assistant Lecturer
Department of Computer and Information Engineering
College of Electronics Engineering
Ninevah University
Al-Jusaq str., Cornish, Mosul, Iraq, 41001
E-mail: mamoon.thanoon@uoninevah.edu.iq
Shefa A. Dawwd
Professor of Computer Engineering, PhD
Department of Computer Engineering
University of Mosul
Al Majmoaa str., Mosul, Iraq, 41001

Received date 11.08.2023
Accepted date 18.10.2023
Published date 30.10.2023

How to Cite: *Al Jbaar, M. A., Dawwd, S. A. (2023). SIMD implementation of deep CNNs for myopia detection on a single-board computer system. Eastern-European Journal of Enterprise Technologies, 5 (9 (125)), 98–108. doi: https://doi.org/10.15587/1729-4061.2023.289007*

1. Introduction

In today's era, computers equipped with cutting-edge GPUs' efficiently handle and process most artificial intelligence and deep learning applications [1]. However, in specific contexts, there is a preference for compact, lightweight, affordable, and efficient computing devices that possess advanced capabilities instead of expensive and bulky computers. This is driven by the need to address specific challenges and provide effective solutions [2]. Single-board computer systems (SBCS) are the best choice for this purpose. These systems are constructed on a single circuit board and have all basic units of the computer system like microprocessors, memory, input/output, and other functional features [3]. Recently, the preference for single-board computer systems has grown due to high applicability and low cost, so they have become frequently used and can be combined with other technology fields [4]. With the technological advancements of single-board computers, artificial intelligence has embraced these systems and employed them in different fields. Their wide adoption can be attributed to their exceptional power efficiency, making them an ideal choice for different applications [5]. Various domains; invest deep learning on a single board, for example, in face detection [6, 7], object detection and classification [8, 9], motion detection [10], as well as medical aspects and human healthcare field [11, 12].

Ophthalmology is one of the most crucial domains that have been embraced by deep learning systems and implemented them in different approaches. These systems play a vital role in detecting and providing early diagnosis for various ocular diseases that impact the eye and disrupt the natural vision system: [13]. Early detection of these diseases is crucial to prevent visual damage. However, there exists a significant disparity between the number of ophthalmologists and the number of patients. Furthermore, the manual evaluation of the fundus is time-consuming and heavily reliant on the skills of ophthalmologists, making thorough fundus examination challenging. Consequently, computer-aided diagnostic procedures are imperative for identifying ocular issues with the assistance of robotics [14]. Ocular problems can manifest differently in various populations, be it in developed or underdeveloped nations [15]. Developing countries, especially those in Asia, often face high rates of untreated ocular conditions [16]. Therefore, embedded computer systems based on deep learning are considered the most suitable solution to provide quick and affordable medical services for the purpose of detection and early diagnosis of these diseases [17].

Most deep learning models for detecting and diagnosing ocular diseases were implemented using well-known and trained deep learning networks, like VGG16, VGG19, AlexNet, ResNet 50, etc. [18, 19]. These networks are characterized by their large size, massive number of calculations, and substantial network parameters, as they were implemented on computer systems with high specifications and contained advanced GPU cards with higher power consumption. All this has limited the flexibility of these networks and restricted the ability to implement those models on a low-cost platform with tolerable power consumption [20]. Therefore, there is a pressing need for a new methodology of deep network implementation and employing low-power platforms for this target. Consequently, there is a real benefit from using such models on a large scale. Nevertheless, embedded platforms like IoT devices and single-board computer systems are unable to accommodate such accelerators due to their limited resources [21]. Thus, the presented methodologies must be optimum and aim to utilize the overall hardware resources provided by the implementation platform. So, one of the most glowing ways is the parallel implementation of inference deep neural networks with whole device resource utilization.

Hence, scientific research for developing deep neural networks, introducing new models, and suggesting distinct implementation approaches to enhance their performance and reduce power consumption is of scientific relevance.

2. Literature review and problem statement

Ophthalmology and related detection and diagnosis of eye diseases is one of the most important fields that have adopted intelligent systems based on deep networks. These systems have been invested as supporting models that provide accurate diagnosis and detection services for various ocular diseases. In general, research workers can be classified into two main groups; the first one includes the intelligent systems that have trained and implemented their architectures by investing the integrated GPUs with computer systems, and they include [22–27].

The main goal of these works is to present new models and get superior results.

While [28–31] are the second group, which represents the researchers' seeking to prove the capability of embedded systems implemented on single-board systems with limited resources and low power consumption in the fields of artificial intelligence and various deep learning networks.

In [22], a deep learning model based on transfer learning was presented for myopia detection. The model consists of two ResNet18 deep-learning classification networks. The first is distinguishing normal and abnormal cases, while the second is classifying pathological and high myopia cases. Nine hundred thirty-two fundus images were preprocessed and used as a dataset for the presented model, which is trained for 30 epochs on NVIDIA GeForce RTX 2060 with 6.0 GB. The model achieved 81.82 %, 83.61 %, and 83.52 % as accuracy, precision, and sensitivity, respectively. However, the results of the proposed model indicate a real need to develop it; this can be achieved by conducting additional processing of the training dataset, which leads to improved classification results. Furthermore, for the second classifier, it is advisable to employ an alternative deep learning network than what was used in the first one, which gives more classification accuracy.

Also, in [23], two models of multi-class classification were employed for multi-ophthalmological disease detection. Both of the presented models' adopted transfer learning with VGG16 with some modifications. The input layer is removed while weights of the top five layers are kept frozen for transfer learning. In contrast, the final fully connected layers are removed. In the first architectural model, left and right fundus images are individually applied to parallel pre-trained VGG16. Then the two feature maps obtained from the parallel CNNs are combined and fed into the Global Average Pooling layer. On the other hand, both left and right fundus images are concatenated and supplied to a single VGG16 for feature extraction, followed by the Global Average Pooling layer. Both models were trained for 100 epochs on the ODIR dataset and optimized through SGD optimizer. So the evaluated metrics for the first architecture are 87.16 %, 84.93 %, and 85.87 % for accuracy, AUC, and F1 score, respectively, while these metrics became 89.06 %, 66.88 %, and 85.57 % for the second one. Still, the second approach exhibits some overfitting, which can be mitigated through augmenting the training data.

A new approach based on VGG19 deep neural network is presented in [24] for eye disease identification. Mainly, the methodology of the presented model is based on training the model on preprocessed dataset images to make a classification decision about whether a healthy or sick image. The model attains 88 % test accuracy, 93 % precision, and a recall of 83 % after 50 training epochs on the dataset collected from the Kaggle website. Nvidia Tesla K80 CUDA Cores (GPU), with 24 GB memory, was targeted as an implementation platform. Nevertheless, the model has a significant processing time, which needs 39.16 minutes for total processing operations.

Furthermore, a glaucoma prediction system was demonstrated in [25], which is based on U-net CNN. The system was trained on a synthesized database of 8,245 images created by combining two sets of fundus image databases LARGE database and the database of the Central Family Clinic Medical Center. The system achieved an accuracy of 94 %, and precision between 90 % to 95 %, while the F1 score was between 91 % and 94 %. However, the number of layers, besides the sitting with various filter sizes, makes the training time so long. Maybe choosing an alternative CNN architecture can address this aspect.

A U-net deep network architecture in [26]; was utilized to segment retinal vessels to help diagnose various dangerous eye disorders. The deep network was trained and tested relying on the DRIVE dataset of 40 retinal images, the presented model running by Intel XeonE5-2683 2.0 GHz processor, and NVIDIA Titan XP GPU framework. The implemented system effectively achieved an accuracy of 95.5 % and 82 %, 98 % for both sensitivity and specificity, respectively. Also, the system accomplished the segmentation process in 21.56 seconds. Nevertheless, the presented system shows a clear contrast between sensitivity and specificity, which the limitation of the training records may cause. Anyway, this can be mitigated by training the model on a different dataset or even through using various augmentation strategies.

Additionally, a combination of CNN and self-attention mechanism was employed in an MBSaNet model in [27] to identify multiple fundus diseases. The features were extracted through the AlexNet CNN, while the self-attention captures the complicated relationships between spatial positions, enabling the system to directly detect one or more diseases in fundus images. MB SaNet approach is utilized to detect diverse ocular diseases like age-related macular degeneration (AMD),

diabetic retinopathy (DR), glaucoma, and others. The proposed system was trained for 30 epochs on the ODIR dataset and implemented on a computer system of Intel Xeon Gold 6226R, 16 cores with 32 threads, and NVIDIA RTX5000, of 32 GB memory. The system's evaluation metrics were 0.88–0.879 for accuracy, and 0.891, 0.881 for both AUC and F1 score, respectively. Nonetheless, it should be noted that the model efficiency might be reinforced by improving the learning process by choosing a new CNN network for this model.

Moreover, in [28], multiple deep learning models were presented based on transfer learning for DR detection. EfficientNet-B6, EfficientNet-B5, Inception v3, VGG19, and ResNet 50; were tested after they were trained on a preprocessed APTOS dataset. The best accuracy score of 86.03 % was achieved by EfficientNet-B6. Consequently, this model was selected as an edge device for inference implementation on the low-cost and power-consuming Raspberry Pi platform, as a single-board system. However, the efficiency of the implemented model still needs to be improved, which can be achieved by adding more layers to the deep network. Since the deep network was implemented with its traditional architecture without any other modification or enhancement, the chosen implementation platform does not allow any other additions due to its limited resources. This obstacle can be addressed by providing a neural network appropriate for the implementation platform, considering the improved accuracy resulting from the proposed model.

Another single-board computer system leverages deep learning relying on transfer learning in [29]. Where Google Lenet was employed to detect cataract ocular disease. The deep network was developed using MATLAB Digital Image Processing paradigm and implemented on Raspberry Pi 3 Model B+ SBC. Whenever the Raspberry Pi camera captures an eye image, it sends it to a dedicated software program accountable for preparing this image as input for the deep network on Raspberry Pi. Even though this system achieves a high accuracy of 96.4 %, this accuracy is not fixed and is based mainly on the image processing carried out by the responsible person. In addition, the interference of the human factor limits the contribution of the system to providing medical support services automatically. In addition, it made the response time of the system too long. Anyway, the system can be modified through adding a preprocessing stage for the deep network as well as optimizing the model architecture to reduce time.

A Raspberry Pi 4 platform was also utilized to implement a specific U-Net deep network for fundus image segmentation in [30]. The network was trained on DRISHTI-GS and RIM-ONE-v3 datasets for both optic disc and optic cup segmentation. The trained network accomplished its function through 1.2 seconds per image as a response time. Thus, the overall architecture of the proposed model may be strengthened through a parallel architecture implementation strategy or by exploiting accelerators that are compatible with the Raspberry Pi board, to enhance the overall inference time.

In [31], two implementation methods were proposed for LeNet-5 of three convolutional layers. The methods strategy aimed to utilize the NEON unit for SIMD processing, which is integrated into each Cortex-A53 core to implement the network in SIMD fashion. However, these units are simple parallel processing elements, which made the inference time very long. The first implementation method achieved about 5.445 sec inference time, while it was about 4.075 sec for the second method. Of course, relying on ARM Cortex cores gives more flexibility and generality for the architectural design and presents powerful results.

After reviewing and analyzing both groups of the previous works, it is evident that studies in the first group were interested in developing a supplementary model for diagnosing one or more eye diseases through systems operating on GPU cards integrated into an advanced and costly computer system. These studies did not address essential aspects such as the amount of power consumed by the presented approach and its portability or ability to work independently.

While research works in group two employed the single-board low-power system to implement their models. However, the results of most of these researches were not at the worthy level, neither in terms of performance accuracy nor in terms of time, and this can be explained by the fact that the presented systems used large-sized deep learning networks without modification or improvement, as well as there is no suggestion of optimum deep networks that are more suitable for approved implementation platforms. These objectives are achieved through new deep networks implemented with new architectures that exploit parallelism in their structure to obtain acceptable results in terms of power consumption and mobility as well as the total cost of the system. Also, with considerable resulting accuracy and speed, making the proposed method more acceptable and practical.

Our research paper tackles the challenge of integrating an efficient deep learning network operating in a cost-effective, power-efficient environment while preserving the accuracy of this network and keeping it at the level of those networks implemented on expensive hardware resources. Whereas the proposed deep learning network was implemented based on the parallel SIMD architecture, which enabled this network to be easily implemented on a single-board multi-core system like ARM Cortex-A53 processor on Raspberry Pi board, which is characterized by its minimal power consumption as well as low cost compared to GPU cards, besides its ability to work as an accurate independent and mobile computer system with ease.

All this allows us to assert that it is expedient to conduct a study on employing parallel architectures for implementing deep learning networks to utilize low-cost platforms in AI applications and models.

3. The aim and objectives of the study

The aim of the study is to develop an efficient SIMD architecture of the proposed deep learning network that can be used as a low-power embedded model in medical diagnostics of myopia ocular disease.

To achieve this aim, the following objectives are accomplished:

– to assess the efficiency of the proposed deep network with the new SIMD parallel implementation on a multi-core single-board computer system;

– to evaluate and compare the power consumption and inference time of the proposed embedded system with previous research works.

4. Materials and methods of research

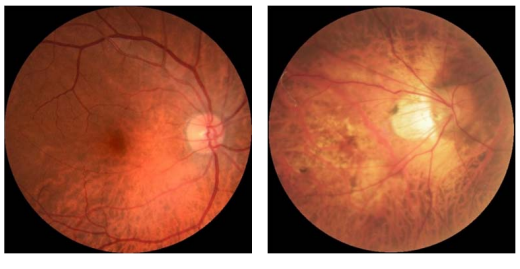
4. 1. Object and hypothesis of the study

This paper presents a new implementation of a deep neural network suggested for automatic myopia ocular disease detection. The proposed network utilizes SIMD architectures, offer-

ing efficient advanced capabilities, and operates efficiently on a compact, low-power single-board platform. The model was trained using the ODIR (Ocular Disease Intelligent Recognition) dataset, on a PC with an Intel® Core™ i9-9900K CPU @ 3.60 GHz 3.60 GHz, 32 GB RAM, and 64-bit operating system. For more practicality, the developed system was inference implemented on Raspberry Pi 3 B and provides an efficient classification with a low-power consumption embedded system. The power measurements were made through a Keweisi USB tester, which measures both voltage and current traffic.

4. 2. The Dataset

Ocular Disease Intelligent Recognition (ODIR) is a compiled ophthalmic database of 5,000 patients, which encompasses information such as age, color fundus photographs from the left and right eyes, as well as; diagnostic keywords provided by doctors. Within this dataset, a set of eye diseases are documented, and myopia is one of the perceptible. Fig. 1 clarifies pathological myopia case vs. normal one.



a *b*

Fig. 1. Pathological myopia: *a* — normal case; *b* — myopia case

Like other datasets, the ODIR dataset has its limitations. It encompasses dark and unclear images, which can adversely affect the accuracy of the results when training the suggested model. To mitigate these obstacles, a pre-processing stage has been introduced. This stage involves cropping and scaling the images to match the neural network inputs, as well as applying histogram equalization for improved visibility and contrast enhancement. Fig. 2 shows the effect of this processing in dataset photos.

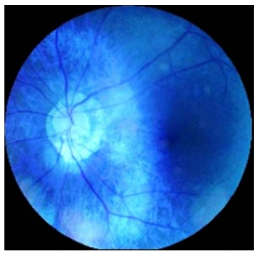


Fig. 2. Sample of dataset photos after preprocessing

Fig. 2 exhibits the effect of pre-processing on highlighting many details of the image, as well as removing the opacity and clarifying the overall image area.

4. 3. The proposed Convolutional Neural Network architecture

In this study, the proposed deep learning network with particular specifications was introduced. This network is presented to be more suitable for AIoT applications and hardware of low-power consumption. Fig. 3 depicts the network architecture layout.

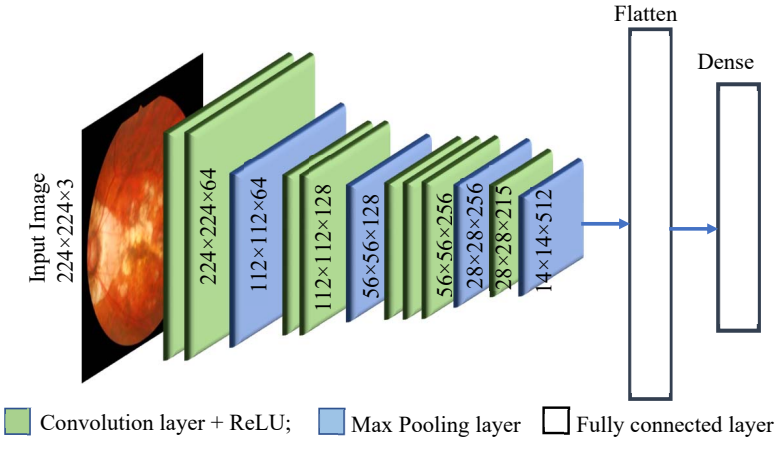


Fig. 3. The proposed deep network

As shown in Fig. 3, the proposed deep network is compact and has a limited number of layers, as compared with other deep learning networks like VGG16, VGG19, ResNet50, and InseptionV3, etc. The first two convolutional layers employ 64 filters of 3x3, and both layers utilize ReLU as an activation function, which ensures that the output is the input value if it is positive; otherwise, the output is zero. After that, the special dimension will be reduced via Max pooling layer. Subsequently, the convolutional and Max pooling layers are used until reaching the flattening layer. An adaptive Moment Estimation optimizer (Adam) was applied to estimate the parameters, with a learning rate=0.0001 and max epochs=70. Table 1 illustrates the deep network details.

Table 1

Proposed network details

Model: "sequential_3"		
Layer (type)	Output shape	Param#
Conv2d_1 (Conv2D)	(None, 224, 224, 64)	1,792
Conv2d_2 (Conv2D)	(None, 224, 224, 64)	36,928
Maxpooling2d_1 (MaxPooling2D)	(None, 112, 112, 64)	0
Conv2d_3 (Conv2D)	(None, 112, 112, 128)	73,856
Conv2d_4 (Conv2D)	(None, 112, 112, 128)	147,584
Maxpooling2d_2 (MaxPooling2D)	(None, 56, 56, 128)	0
Conv2d_5 (Conv2D)	(None, 56, 56, 256)	295,168
Conv2d_6 (Conv2D)	(None, 56, 56, 256)	590,080
Conv2d_7 (Conv2D)	(None, 56, 56, 256)	590,080
Maxpooling2d_3 (MaxPooling2D)	(None, 56, 56, 256)	0
Conv2d_8 (Conv2D)	(None, 28, 28, 512)	1,180,160
Maxpooling2d_4 (MaxPooling2D)	(None, 14, 14, 512)	0
Flatten_1 (Flatten)	(None, 100352)	0
Dense_1 (Dense)	(None, 1)	100,353
Total params	3,016,001	
Trainable params	3,016,001	
Non-trainable params	0	

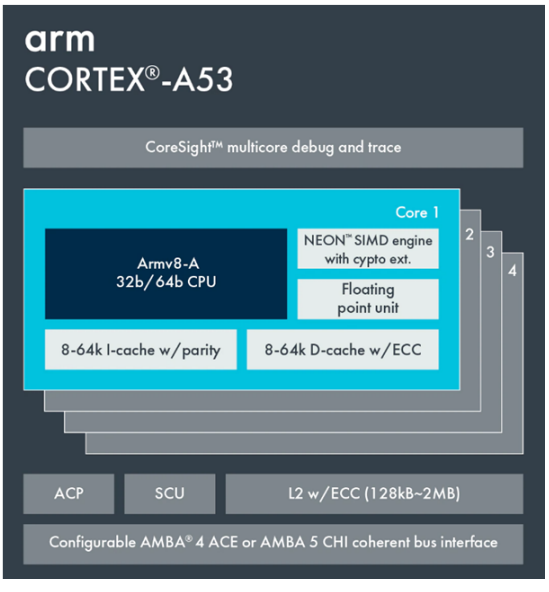
As shown in Table 1, the proposed network consists of only eight convolutional layers, and the overall network parameters are about 3M.

4. 4. SIMD implementation of the proposed Convolutional Neural Network architecture

The parallel implementation of deep learning networks is considered as an advanced approach that achieves faster execution while minimizing the instruction bandwidth as well as the need for repetitive memory access requests. To implement the proposed deep network based on the parallel architecture, the SIMD architecture was adopted, which enables the execution of a single instruction on a set of data simultaneously.

The inherent parallelism present in the proposed network was exploited to implement it using the SIMD parallel architecture. In this approach, all the convolution and Max pooling processes were divided into multiple threads running in parallel, allowing them to produce their results synchronously independently. In contrast, these processes run sequentially in traditional implementations of deep learning networks. The Raspberry Pi 3 Model B is a dedicated, power-efficient single-board embedded hardware, so we deploy it for implementing our deep learning inference model. The embedded platform has a quad-core ARM Cortex-A53 processor, delivering a maximum clock speed of 1.2 GHz. It is complemented by a 1GB LPDDR2 RAM module [32]. The internal architecture of the Raspberry Pi 3 B microcomputer is shown in Fig. 4, *a*, while Fig. 4, *b* reviews the Keweisi USB tester that is used for power measurement.

To utilize the capabilities of the quad-core ARM Cortex processor, we divided the convolution and Max pooling operations of our deep network into four threads that operate simultaneously in parallel. This approach allows the execution of a single instruction on a set of data simultaneously, achieving parallelism in the overall execution. Fig. 5 demonstrates the proposed implementation of the presented deep network with the SIMD architecture.



a

b

Fig. 4. Research materials:
a — Raspberry Pi 3 B; *b* — Keweisi USB tester

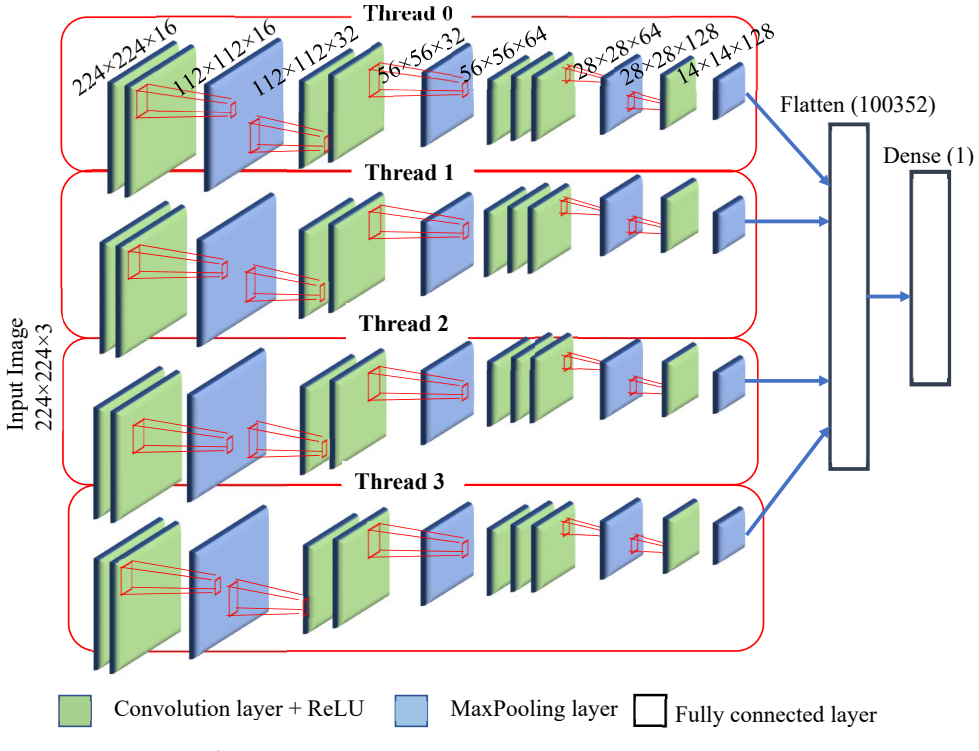


Fig. 5. SIMD parallel implementation of the proposed deep learning network

According to Fig. 5, the presented deep network was partitioned into four threads, one for each core of the Cortex CPU, and each thread consists of a succession of convolutional and Max-pooling layers for feature extraction and dimension reduction, respectively. Thus, in this context, all network functionality is implemented simultaneously, and this architecture achieves more speeding up than the sequential model that is implemented on a single core.

Within each core, the execution of the convolution and Max-pooling operations as repetitive and overlapping iterations with each other in a pipeline aspect is interspersed with caching operations of the intermediate results obtained from each group, as they are stored in the cache memory of this core to proceed for the next process. So the sequence of operations continues until the core completes all operations for the thread entrusted to it.

Selecting the nested pipeline as an operational framework for each core increases the core's productivity and speeds up the operations within a particular thread. Fig. 6 shows the successive iterations in each thread through the quad cores of the Cortex processor, as well as the pipeline implementation of each one.

Fig. 6 shows the details of the parallel implementation of the convolution process through the proposed SIMD architecture. The first two convolution layers, which are both 224×224×16 in each core, were accomplished through four iteration groups, each one consisting of four convolutional processes. The intermediate results of these operations are collected in the L1 cache of the core to be used in the following Max-pooling processes. The Max-pooling operations are also implemented in the same parallel manner. After that, the parallel implementation of the successive convolution and Max-pooling layers continues implementing the proposed deep network with a new architecture based on fulfilling parallelism in network operations.

The use of SIMD architectures in implementing the proposed network enhanced its performance by utilizing multi-core programming to accomplish the same processing operations in parallel and increase the speed up productivity.

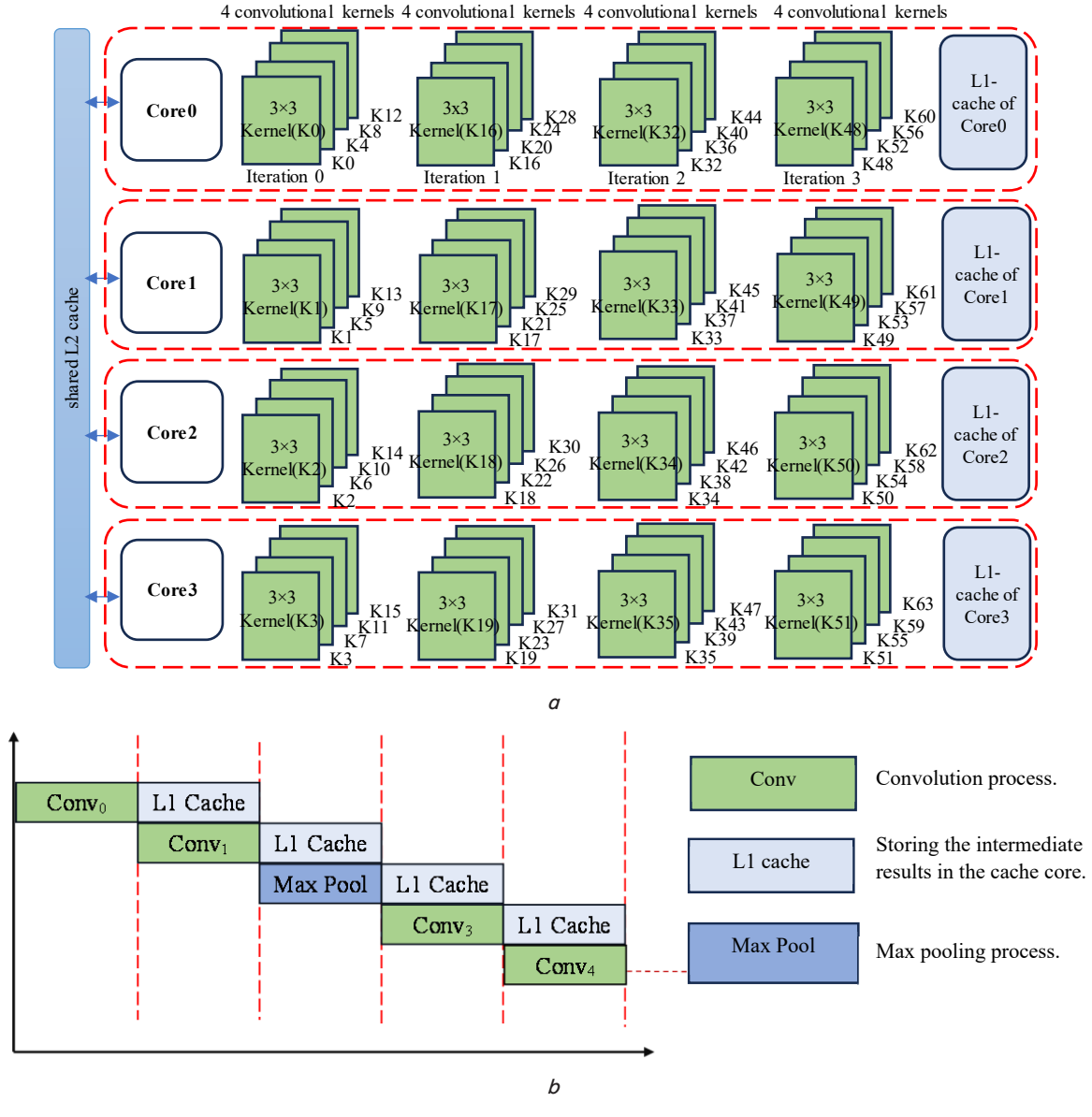


Fig. 6. Core implementation tasks: *a* — implementation details of threads in each core; *b* — pipeline overlapped operations/core

5. Results of the SIMD deep CNNs model

5. 1. Prediction accuracy results of the presented model

As mentioned earlier, the proposed system was trained and tested based on the ODIR database. This dataset was divided into 80:20, where 80 % of this data was used for system training and 20 % for testing. The system was trained for 70 epochs and achieved a prediction accuracy of 96.35 %, as clearly shown in Fig. 7, which expresses the high efficiency of the proposed system through its evaluation metrics.

```
767/767 [==============================] -4s 5ms/step -loss: 0.2372 -
accuracy: 0.9572 -val_loss: 0.3740 -val_accuracy: 0.9219
Epoch 68/70
767/767 [==============================] -4s 5ms/step -loss: 0.1734 -
accuracy: 1.0000 -val_loss: 0.4100 -val_accuracy: 0.9688
Epoch 69/70
767/767 [==============================] -4s 5ms/step -loss: 0.1510 -
accuracy: 1.0000 -val_loss: 0.4159 -val_accuracy: 0.9635
Epoch 70/70
767/767 [==============================] -4s 5ms/step -loss: 0.2559 -
accuracy: 0.9791 -val_loss: 0.4136 -val_accuracy: 0.9635
```

a

	precision	*recall*	*F1-score*
0	0.97	0.96	0.96
1	0.96	0.97	0.96
accuracy			0.96
Macro avg	0.96	0.96	0.96
Weighted avg	0.96	0.96	0.96

b

Fig. 7. Outcomes of the introduced network: *a* – model training epochs; *b* – efficiency evaluation metrics

The model behavior during the 70 training epochs is described in model accuracy and model loss, which are shown in Fig. 8.

However, relying on accuracy to evaluate the efficiency of the system is insufficient. Therefore, in order to give a comprehensive description of the performance of the proposed system, we have adopted integrated metrics to describe the efficiency of the system, including Precision, Recall, and F1 score.

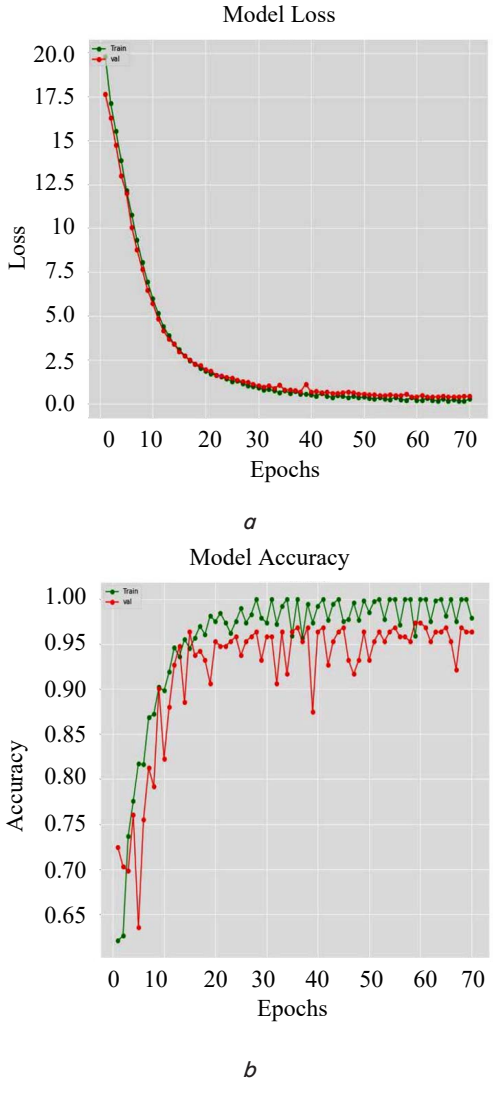
The Precision for our model, which is described in (1), gives an evaluation of the proportion of disease-positive patients correctly identified among the entire dataset:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}. \quad (1)$$

Recall, on the other hand, evaluates the number of true positives that have been precisely classified, which in our prediction models refers to individuals that are truly suffering and have been predicted by our model to be afflicted on the other hand. The Recall is calculated by (2):

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}. \quad (2)$$

In addition, a common evaluation metric that combines the precision and recall of a classifier into a single value is the F1 score, whereas this metric evaluates the overall performance of a binary classifier. The F1 score ranges from 0 to 1, with a higher score indicating better performance. The calculated equation of the F1 score is shown in (3):



a

b

Fig. 8. Myopia detection: *a* – model accuracy; *b* – model loss

$$\text{F1 Score} = \frac{\text{Precision}^* \text{ Recall}}{\text{Precision} + \text{Recall}}. \quad (3)$$

The outcomes of these metrics abstract the system performance, where the higher values depict the high efficiency of the system, and this is depicted in Fig. 7, *b* concerning our proposed system.

5. 2. Power and time consumption of the proposed model, compared with previous works

The significant factor for evaluating any system is the power consumption; thus, practical and efficient embedded systems must exhibit a defined and reasonable power consumption. This benchmark plays a critical role in specifying the overall system cost, and for remote and self-powered systems, this metric specifies the lifetime of the system work. Thus, the power consumption of the system is one of the essential criteria adopted in the development of the embedded system presented in this research work, where this aspect was addressed by building a finite-layer deep neural network besides partitioning all the sequential computations into parallel groups and finally by adopting the implementation platform with shallow power consumption. This board has

been selected for its low power consumption; it used about 1.15 Watts during idle mode and without any USB device connection, while its peak power consumption is about 3.6 Watts. Thus, it surpasses the Raspberry 4 by about 50 % in power consumption [33]. Fig. 9 shows the initial power consumption of Raspberry Pi 3 B in our experiment work by using the Keweisi USB power tester.

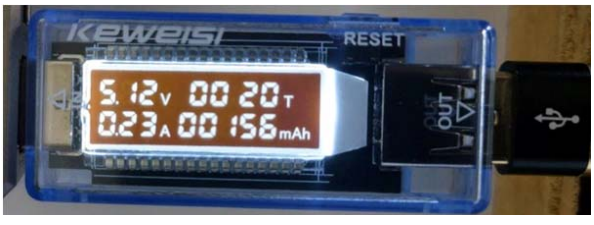


Fig. 9. Raspberry Pi 3 B initial power consumption

As shown in Fig. 9, the system board was driven by 5.12 V and 0.23 A, so the initial power for idle mode is about 1.1776 Watts (with USB wireless mouse connection and LCD monitor).

To emphasize the significance of incorporating parallel architectures in the construction of deep learning networks, our experimental work involves two steps. In the first one, only one core processor of the ARM processor was utilized to implement the entire proposed deep network without employing SIMD parallel architecture, and the power consumption of this step is shown in Fig. 10.

As shown in Fig. 10, our design architecture consumes about 2,783 Watts when it is implemented on a single core of an ARM processor. Furthermore, it is noteworthy that in this implementation, the proposed system needs about 16 seconds to give its final prediction result.

However, a new SIMD parallel architecture of the proposed deep network was implemented utilizing the overall quad-cores of the ARM Cortex processor in the second step, and its power consumption result is shown in Fig. 11.

Also, Fig. 11 clarifies that the total current needed to drive the SIMD architecture is 0.73 A, thus the total power consumption of the new implementation is 3.65 watts. Moreover, the system with a SIMD deep network needs only 3.25 seconds to give its prediction response.

Fig. 12 shows the difference between single-core and quad-core implementations of the proposed deep network on the Raspberry Pi 3 B platform.

Comparing the results of the prior two implementations, we note that the second implementation, which supported the parallel architectures in the structuring of deep networks, achieved a speedup of about 4.923 with a slight difference in power consumption not exceeding 0.867 Watts.

Hence, Fig. 12 highlights the importance of utilizing parallelism in deep neural networks and employing parallel architectures in constructing them. Of course, if ARM processors have more cores our embedded system can be faster.

Fig. 10. Raspberry Pi 3 B single-core power consumption

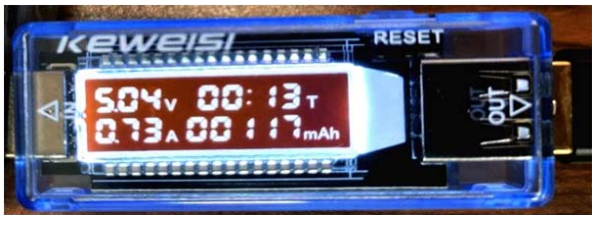


Fig. 11. Raspberry Pi 3 B quad-core power consumption

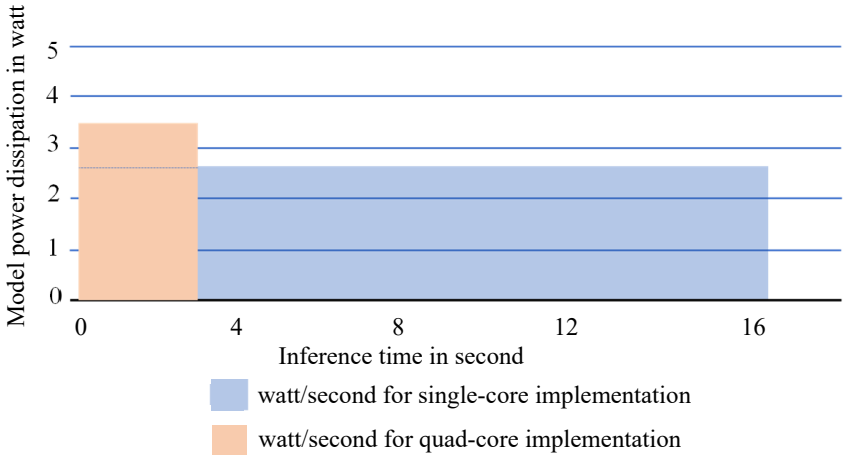


Fig. 12. Power consumption/inference time for both single and quad-core implementation

Our proposed model was compared with multiple designs and different deep-learning architectures. These comparisons were on two levels, the first is a comparison with published works of deep networks and intelligent systems that deal with detecting and diagnosing ocular diseases, as shown in Table 2.

Table 3, on the other hand, depicts the second level of these comparisons, including a comparison of our system's architecture with different implementations of deep learning models on the Raspberry Pi platform.

The comparisons reviewed in Table 2 show a clear superiority of the presented deep network and the proposed system over its counterparts of neural networks in previous and presented studies for ocular disease detection and diagnosing. The table also indicates the high accuracy concluded by the proposed network relative to its limited size, which may reach 18.667 and 3, compared to those in [36, 37]. The size of our network stands out with the possibility of implementing it on various implementation platforms with limited power consumption as a single-board mobile system, with a maximum power consumption of about 3.65 W, compared to those vast networks that can only be implemented on GPU with high specifications and significant power consumption, which is part of an integrated computer system with a very high cost.

Table 2

Model comparison with different ocular disease detection systems

Reference	Deep learning models	Network parameter size	Implementation platform	Power consumption (W)	Accuracy
[34]	EFFICIENT NET B3	10.71M	NVIDIA RTX 2080Ti,11GB	265	0.920
[35]	RESNET-50	23.9M	NVIDIA GEFORCE 1080Ti, 11GB	250	0.928
[36]	Sequential Model of INCEPTION RESNET AND DKC BLOCK	56M	NVIDIA T4 GPU, 16GB	70	0.9608
[23]	MBSANET	9.4M	NVIDIA RTX5000, 16GM	265	0.881
[37]	VGG16	528M	Nvidia GeForce RTX 2070, 8GB	175	90.28
	RESNET201	77M			89.49
	DEEP CNN OF 20 LAYERS	27M			93.81
Our design	THE PROPOSED CNN	3M	RASPBERRY PI 3 B	3.65	0.9635

Table 3

Model comparison with different deep learning models on Raspberry Pi

References	Deep network	Implementation platform	Aim of design or application	Inference Time (s)
[38]	InceptionV3	Raspberry Pi 4	Image classification	71
	VGG16			50
	MobileNet			52
[39]	CondenseNet	Raspberry Pi 3	Low-Power Image Classification on Embedded Devices	4.829
[7]	ResNet	Raspberry Pi 3	Face Detection & Recognition from Images	30
[40]	VGG16	Raspberry Pi 4	CNN Inference Acceleration in Edge Computing	6.73
[41]	MobileNetV3	Raspberry Pi 4 B	Distributed Deep Learning Inference Using Raspberry Pi	20.67
Our system	Deep CNN	Raspberry Pi 3 B	Myopia Ocular disease detection	3.25

Meanwhile, Table 3 shows the dominance of the parallel implementation of deep learning networks and utilization of all available resources provided by the accessible platform; over the traditional implementation methods that are used in other deep learning networks implemented on the same and similar implementation platforms.

The parallel SIMD implementation of the proposed network achieved optimum inference time and accelerated the system response compared to what has been achieved in previous systems and studies.

6. Discussion of the experimental results of the optimized SIMD embedded system

The SIMD implementation of the deep network structure presented in this research study, shown in Fig. 5, played an essential role in multiple aspects, as the implementation method made it possible to transform the successive computation tasks within the deep network into sets of parallel partials operations that can be run synchronously with each other. Thus, this provided a significant reduction in the overall execution time on the selected execution platform, compared to those in other architectures implemented on the same and similar implementation platforms as in [7, 38, 41], shown in Table 3.

Moreover, the results show that the implementation of the SIMD gave the possibility to invest all the material resources that the implementation platform provides, such as the investment of the four cores provided by the ARM Cortex processor, and make them work simultaneously by employing them all in parallel simultaneous operations, leading to a reduction in the total inference time.

On the other hand, the proposed deep network, along with the utilization of parallel architectures in its implementation, contributed to the possibility of carrying out this deep network on a low-cost single-board microcomputer system such as Raspberry Pi 3 B. This implementation platform is characterized by its deficient power consumption compared to other implementation platforms shown in Table 2, where a set of proposed systems that were presented as aids in detecting and diagnosing ocular diseases are highlighted in this table, wherein the results in Fig. 9–11 show that the power gain achieved in the proposed system compared to the systems presented in the same field alternated between 72.6 %, 68.49 %, 19.17 %, and 47.94 % for [34–37] respectively in Table 2.

However, one of the most significant restrictions in this research work is the resource limitations of the targeted board. Despite the optimality of the Raspberry Pi 3 in its power consumption, the processing unit of the board is only quad-core, which limits the possibility of achieving greater parallelism through the computational tasks of the deep neural network and thus achieving a more significant reduction in inference time. From this point, developing the processing unit on the Raspberry Pi 3 platform can be one of the crucial developments for this practical research by adding terminal accelerators, like Google Coral USB, which adds supplementary processing cores besides those in the ARM Cortex processor.

7. Conclusions

1. With about 3M parameters only, the proposed deep network contributed to the introduction of an advanced low-power consumption embedded system that can be employed efficiently to detect and diagnose myopia ocular dis-

ease with a high accuracy of up to 96.35 %. The suggested system is executable on low-cost single-board mobile platforms, like Raspberry Pi 3 B. Thus, the submitted network has helped dispense with large networks that need advanced substantial resources, high cost, and higher power consumption.

2. The SIMD parallel implementation of the deep network architecture gained a maximum speed-up of 21.84, compared with the deep network implementations in [38], with the optimum system response and shortened inference time.

The proposed system represents an ideal and supportive assistant for the ophthalmology field, especially as it is marked with high accuracy, low costs as well as low power consumption.

With appropriate modifications, our embedded system will be suitable for a wide range of AIoT applications that rely on deep learning.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

The manuscript has associated data in a data repository.

References

1. Suzen, A. A., Duman, B., Sen, B. (2020). Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN. 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). doi: https://doi.org/10.1109/hora49412.2020.9152915
2. Choi, K., Sobelman, G. E. (2022). An Efficient CNN Accelerator for Low-Cost Edge Systems. ACM Transactions on Embedded Computing Systems, 21 (4), 1–20. doi: https://doi.org/10.1145/3539224
3. Fernández-Cerero, D., Fernández-Rodríguez, J. Y., Álvarez-García, J. A., Soria-Morillo, L. M., Fernández-Montes, A. (2019). Single-Board-Computer Clusters for Cloudlet Computing in Internet of Things. Sensors, 19 (13), 3026. doi: https://doi.org/10.3390/s19133026
4. Saranya, V., Carmel Mary Belinda, M. J., Kanagachidambaresan, G. R. (2020). An Evolution of Innovations Protocols and Recent Technology in Industrial IoT. Internet of Things for Industry 4.0, 161–175. doi: https://doi.org/10.1007/978-3-030-32530-5_11
5. Srinivasan, V., Meudt, S., Schwenker, F. (2019). Deep Learning Algorithms for Emotion Recognition on Low Power Single Board Computers. Multimodal Pattern Recognition of Social Signals in Human-Computer-Interaction, 59–70. doi: https://doi.org/10.1007/978-3-030-20984-1_6
6. Dubovečak, M., Dumić, E., Bernik, A. (2023). Face Detection and Recognition Using Raspberry PI Computer. Tehnički Glasnik, 17 (3), 346–352. doi: https://doi.org/10.31803/tg-20220321232047
7. Zamir, M., Ali, N., Naseem, A., Ahmed Frasteen, A., Zafar, B., Assam, M., Othman, M., Attia, E.-A. (2022). Face Detection & Recognition from Images & Videos Based on CNN & Raspberry Pi. Computation, 10 (9), 148. doi: https://doi.org/10.3390/computation10090148
8. Huang, Z., Yang, S., Zhou, M., Gong, Z., Abusorrah, A., Lin, C., Huang, Z. (2021). Making accurate object detection at the edge: review and new approach. Artificial Intelligence Review, 55 (3), 2245–2274. doi: https://doi.org/10.1007/s10462-021-10059-3
9. Sonkar, S., Kumar, P., George, R. C., Yuvaraj, T. P., Philip, D., Ghosh, A. K. (2022). Real-Time Object Detection and Recognition Using Fixed-Wing LALE VTOL UAV. IEEE Sensors Journal, 22 (21), 20738–20747. doi: https://doi.org/10.1109/jsen.2022.3206345
10. Didi, Z., El Azami, I., Boumait, E. M. (2022). Design of a Security System Based on Raspberry Pi with Motion Detection. Digital Technologies and Applications, 427–434. doi: https://doi.org/10.1007/978-3-031-02447-4_44
11. Hammad, M., Abd El-Latif, A. A., Hussain, A., Abd El-Samie, F. E., Gupta, B. B., Ugail, H., Sedik, A. (2022). Deep Learning Models for Arrhythmia Detection in IoT Healthcare Applications. Computers and Electrical Engineering, 100, 108011. doi: https://doi.org/10.1016/j.compeleceng.2022.108011
12. Dhar, T., Dey, N., Borra, S., Sherratt, R. S. (2023). Challenges of Deep Learning in Medical Image Analysis—Improving Explainability and Trust. IEEE Transactions on Technology and Society, 4 (1), 68–75. doi: https://doi.org/10.1109/tts.2023.3234203
13. Vayadande, K., Ingale, V., Verma, V., Yeole, A., Zawar, S., Jamadar, Z. (2022). Ocular Disease Recognition using Deep Learning. 2022 International Conference on Signal and Information Processing (IConSIP). doi: https://doi.org/10.1109/iconsip49665.2022.10007470
14. Albahli, S., Ahmad Hassan Yar, G. N. (2022). Automated detection of diabetic retinopathy using custom convolutional neural network. Journal of X-Ray Science and Technology, 30 (2), 275–291. doi: https://doi.org/10.3233/xst-211073
15. Ebri, A. E., Govender, P., Naidoo, K. S. (2019). Prevalence of vision impairment and refractive error in school learners in Calabar, Nigeria. African Vision and Eye Health, 78 (1). doi: https://doi.org/10.4102/aveh.v78i1.487
16. Pakbin, M., Katibeh, M., Pakravan, M., Yaseri, M., Soleimanizad, R. (2015). Prevalence and causes of visual impairment and blindness in central Iran; The Yazd eye study. Journal of Ophthalmic and Vision Research, 10 (3), 279. doi: https://doi.org/10.4103/2008-322x.170362
17. Gibertoni, G., Borghi, G., Rovati, L. (2022). Vision-Based Eye Image Classification for Ophthalmic Measurement Systems. Sensors, 23 (1), 386. doi: https://doi.org/10.3390/s23010386

18. da Rocha, D. A., Ferreira, F. M. F., Peixoto, Z. M. A. (2022). Diabetic retinopathy classification using VGG16 neural network. Research on Biomedical Engineering, 38 (2), 761–772. doi: https://doi.org/10.1007/s42600-022-00200-8
19. Pan, Y., Liu, J., Cai, Y., Yang, X., Zhang, Z., Long, H. et al. (2023). Fundus image classification using Inception V3 and ResNet-50 for the early diagnostics of fundus diseases. Frontiers in Physiology, 14. doi: https://doi.org/10.3389/fphys.2023.1126780
20. Menghani, G. (2023). Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better. ACM Computing Surveys, 55 (12), 1–37. doi: https://doi.org/10.1145/3578938
21. Islam, S., Deng, J., Zhou, S., Pan, C., Ding, C., Xie, M. (2022). Enabling Fast Deep Learning on Tiny Energy-Harvesting IoT Devices. 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE). doi: https://doi.org/10.23919/date54114.2022.9774756
22. Dai, S., Chen, L., Lei, T., Zhou, C., Wen, Y. (2020). Automatic Detection Of Pathological Myopia And High Myopia On Fundus Images. 2020 IEEE International Conference on Multimedia and Expo (ICME). doi: https://doi.org/10.1109/icme46284.2020.9102787
23. Gour, N., Khanna, P. (2021). Multi-class multi-label ophthalmological disease detection using transfer learning based convolutional neural network. Biomedical Signal Processing and Control, 66, 102329. doi: https://doi.org/10.1016/j.bspc.2020.102329
24. Topaloglu, I. (2022). Deep Learning Based Convolutional Neural Network Structured New Image Classification Approach for Eye Disease Identification. Scientia Iranica, 30 (5), 1731–1742. doi: https://doi.org/10.24200/sci.2022.58049.5537
25. Rakhmetulayeva, S., Syrymbet, Z. (2022). Implementation of convolutional neural network for predicting glaucoma from fundus images. Eastern-European Journal of Enterprise Technologies, 6 (2 (120)), 70–77. doi: https://doi.org/10.15587/1729-4061.2022.269229
26. David, S. A., Mahesh, C., Kumar, V. D., Polat, K., Alhudhaif, A., Nour, M. (2022). Retinal Blood Vessels and Optic Disc Segmentation Using U-Net. Mathematical Problems in Engineering, 2022, 1–11. doi: https://doi.org/10.1155/2022/8030954
27. Wang, K., Xu, C., Li, G., Zhang, Y., Zheng, Y., Sun, C. (2023). Combining convolutional neural networks and self-attention for fundus diseases identification. Scientific Reports, 13 (1). doi: https://doi.org/10.1038/s41598-022-27358-6
28. Maqsood, Z., Gupta, M. K. (2022). Automatic Detection of Diabetic Retinopathy on the Edge. Cyber Security, Privacy and Networking, 129–139. doi: https://doi.org/10.1007/978-981-16-8664-1_12
29. Karamihan, K. C., Agustino, I. D. F., Bionesta, R. B. B., Tuason, F. C., Arellano, S. V. E., Esguerra, P. A. M. (2019). SBC-Based Cataract Detection System using Deep Convolutional Neural Network with Transfer Learning Algorithm. International Journal of Recent Technology and Engineering (IJRTE), 9(2), 4605–4613. doi: https://doi.org/10.35940/ijrte.b3368.078219
30. Civit-Masot, J., Luna-Perej n, F., Corral, J. M. R., Dom nguez-Morales, M., Morgado-Est vez, A., Civit, A. (2021). A study on the use of Edge TPUs for eye fundus image segmentation. Engineering Applications of Artificial Intelligence, 104, 104384. doi: https://doi.org/10.1016/j.engappai.2021.104384
31. Lee, S.-J., Park, S.-S., Chung, K.-S. (2018). Efficient SIMD implementation for accelerating convolutional neural network. Proceedings of the 4th International Conference on Communication and Information Processing. doi: https://doi.org/10.1145/3290420.3290444
32. Raspberry Pi 3 Model B. URL: https://www.raspberrypi.com/products/raspberry-pi-3-model-b/
33. Raspberry Pi Power Consumption Guide. URL: https://www.ecoenergygeek.com/raspberry-pi-power-consumption/
34. Wang, J., Yang, L., Huo, Z., He, W., Luo, J. (2020). Multi-Label Classification of Fundus Images With EfficientNet. IEEE Access, 8, 212499–212508. doi: https://doi.org/10.1109/access.2020.3040275
35. He, J., Li, C., Ye, J., Qiao, Y., Gu, L. (2021). Multi-label ocular disease classification with a dense correlation deep neural network. Biomedical Signal Processing and Control, 63, 102167. doi: https://doi.org/10.1016/j.bspc.2020.102167
36. Bhati, A., Gour, N., Khanna, P., Ojha, A. (2023). Discriminative kernel convolution network for multi-label ophthalmic disease detection on imbalanced fundus image dataset. Computers in Biology and Medicine, 153, 106519. doi: https://doi.org/10.1016/j.compbiomed.2022.106519
37. Jeny, A. A., Junayed, M. S., Islam, M. B. (2023). Deep Neural Network-Based Ensemble Model for Eye Diseases Detection and Classification. Image Analysis & Stereology, 42 (2), 77–91. doi: https://doi.org/10.5566/ias.2857
38. Kristiani, E., Yang, C.-T., Huang, C.-Y. (2020). iSEC: An Optimized Deep Learning Model for Image Classification on Edge Computing. IEEE Access, 8, 27267–27276. doi: https://doi.org/10.1109/access.2020.2971566
39. Goel, A., Aghajanzadeh, S., Tung, C., Chen, S.-H., Thiruvathukal, G. K., Lu, Y.-H. (2020). Modular Neural Networks for Low-Power Image Classification on Embedded Devices. ACM Transactions on Design Automation of Electronic Systems, 26 (1), 1–35. doi: https://doi.org/10.1145/3408062
40. Dong, Z., Li, N., Iosifidis, A., Zhang, Q. (2022). Design and Prototyping Distributed CNN Inference Acceleration in Edge Computing. arXiv. doi: https://doi.org/10.48550/arXiv.2211.13778
41. James, N., Ong, L.-Y., Leow, M.-C. (2022). Exploring Distributed Deep Learning Inference Using Raspberry Pi Spark Cluster. Future Internet, 14 (8), 220. doi: https://doi.org/10.3390/fi14080220