

An Approach to Assist Ophthalmologists in Glaucoma Detection Using Deep Learning

27

Tatiane Martins Bistulfi, Marcelli Marques Monteiro,
and Juliano de Almeida Monte-Mor

Abstract

Artificial neural networks consist of computational models inspired by a central nervous system, being capable of machine learning and pattern recognition. Based on the technological advancement of these networks and the possibility of applying new technologies of artificial intelligence and image processing to problems in the field of medicine, this project seeks to apply the *deep learning* technique with different architectures as *ResNet50*, *DenseNet* and *VggNet – 16* along with digital image processing in which different filters were applied to a set of 512 images in order to develop a system to aid in the identification of glaucoma optic neuropathy, considering retinal images as input for training a network capable of detecting glaucomatous patterns. Through this present study, in which a combination of different architectures, activation functions as *Softmax*, *ReLU* and *Sigmoid*, were used for image classification. It is expected that the system will be able to help specialist physicians in detecting the disease during examinations performed on patients' eyes, considering that three architectures obtained satisfactory results above 80% accuracy, among them a model using an image filter called malachite that was specially developed for this study.

Keywords

Activation function · Artificial neural networks · Deep learning · DenseNet · Digital image processing · ReLU · ResNet50 · Sigmoid · Softmax · VggNet – 16

T. M. Bistulfi (✉) · M. M. Monteiro · J. de Almeida Monte-Mor
Federal University of Itajubá – Itabira Campus, Itabira, MG, Brazil
e-mail: anebistulfi@unifei.edu.br; marcellimonteiro@unifei.edu.br;
jmontemor@unifei.edu.br

27.1 Background

Glaucoma is an optic neuropathy that can be caused by elevated intraocular pressure due to an accumulation of aqueous humor in the anterior chamber of the eye, and by a failure in Schelmm's canal, which makes it impossible for this liquid to drain into the bloodstream.

The projections by the International Agency for the Prevention of Blindness (IAPB) to 2020 indicated that there would be approximately 80 million people with glaucoma worldwide, an increase of about 20 million since 2010. Further, it was estimated that 3.2 million people there would be blind due to glaucoma by 2020 and that the number of people with glaucoma worldwide will increase to 111.8 million by 2040 [1].

Diagnosis can be made from digital retinal image analysis, because the amount of optic nerve fiber loss has a direct effect on the configuration of the Neuro Retinal Rim (NRR). As the optic nerve fibers die, the cup becomes wider in relation to the optic disc (OD), which results in an increased Cup to Disc Ratio (CDR) value [2].

The main symptom caused by the loss of the optic fibers is the alteration or loss of the visual field, because this nerve is responsible for transmitting the lateral, peripheral, and central visual messages to the brain. When detected at a less advanced stage, the progression of the optic nerve damage can be controlled by using specific medications. There are patients with the disease who can have normal intraocular pressure on retinal scans and still have glaucoma. In this case, they still had a loss of optic nerve fibers, and they were also diagnosed by retinal imaging.

In the current scenario, where technology is in all segments of our society, computational techniques have been successfully adopted to solve problems in medicine. Many

studies are aimed at helping in the diagnosis of diseases, applying intelligent techniques, such as Deep Learning (DL) and Digital Image Processing (DIP), to aid in medical diagnoses.

DL seek to discover a model using a set of data and a method to guide the learning of the model from this data. At the end of the learning process, there is a function capable of taking the raw data as input and providing as output an adequate representation of the problem at hand [3]. Convolutional Neural Networks (CNNs) are DL network models composed of convolutional layers, which process inputs by considering local receptive fields.

In this context, this study considers applying DL to detect the existence of glaucoma, from a digital image set of patients' optic nerve, using different parameterizations and DL models. We expect that the proposed approach based on the DL technique can assist ophthalmology physicians during the process of detecting glaucoma patterns. For this, we compare some hyper-parameters of network architectures and performance metrics, aiming to choose a DL model suitable to the problem described here.

We consider the evaluation of the following different network architectures, with DL characteristics, as pertinent for this study: ResNet50 Architecture [4], DenseNet Architecture [5], and VGGNet-16 Architecture [6].

The rest of this article is organized as follows: Sect. 27.2 describes the methodology adopted for development the work; Sect. 27.3 presents the results analysis; and, finally, Sect. 27.4 presents our conclusions for this study.

27.2 Methodology

This section will describe the processes and techniques adopted in the development of this work, which served as a basis for the experimental studies analyzed in the next section. The development of this work followed the four steps described below: database construction; preprocessing of images; network model development; training; and, finally, testing and validation.

27.2.1 Database Construction

Initially, we collected retinal images of patients obtained by funduscopy examination for the modeling of the training, validation and testing bases. In this regard, we found three image bases: the STARE (STructured Analysis of the REtina) [7], Messidor database [8] and High-Resolution Fundus (HRF). We chose 512 images to obtain a balanced dataset. Figure 27.1 illustrates an example of a database image for the neural network development.

27.2.2 Preprocessing of Images

Sequently, we subjected the RGB database to image preprocessing, in which an area of interest (ROI) was determined, the optical disk, thus avoiding interference from parts that was not the interest to the study.

After the images were obtained, they were treated through DIP methods, by applying filtering techniques. The filtering process consisted in transforming the image into a pixel by pixel matrix, called mask, which transforms each pixel into a corresponding gray level. The Gray-Level Co-occurrence Matrix (GLCM) filter is defined as a matrix over an image for the distributing of co-occurring pixel values at a given offset. It has been adopted for texture analysis in various applications, particularly, in medical image analysis [10]. For this reason, we addressed GLCM as a filters for image processing during the development of this project. Our Malachite filter, described in the next section, was applied for analytical comparison of the GLCM filter performance.

After the image bank was pre-processed, we fractionated the image set for each filter into 60% for training (308 images), 20% for testing (102 images) and 20% for validation (102 images).

27.2.2.1 Malachite Filter

After importing the dataset, we generated activations on the red (R) and blue (B) channels of the RGB images, keeping the green (G) channels disabled. A high-pass filter was then used to highlight edges or regions of interest with sharp intensity transitions and abrupt transitions in intensity. Examples of masks for high-pass filters are given in Fig. 27.2.

Thus, after we performed the matrix convolutions on the (R) and (B) channels, the generated image showed enhancement in the nerves of the optic disc, as well as greenish nuances. In the present study, we named this preprocessing method as Malachite Filter.

27.2.3 Network Model Development

After the database processing step, the images were suitable to be used as input data in the proposed CNNs.

The definition of the neural network architecture is a critical step, since it directly affects the network's processing capacity. Caution was required so that the network would not suffer from excessive numbers of neurons/layers, causing overfitting, or lack of processing units causing network underfitting.

For developing the proposed CNN, we adopted the open source TensorFlow Federated (TFF) library, which is widely used in the DL area. Thus, the TFF allowed the importation of the image database until the validation of the CNNs proposed in this experiment.

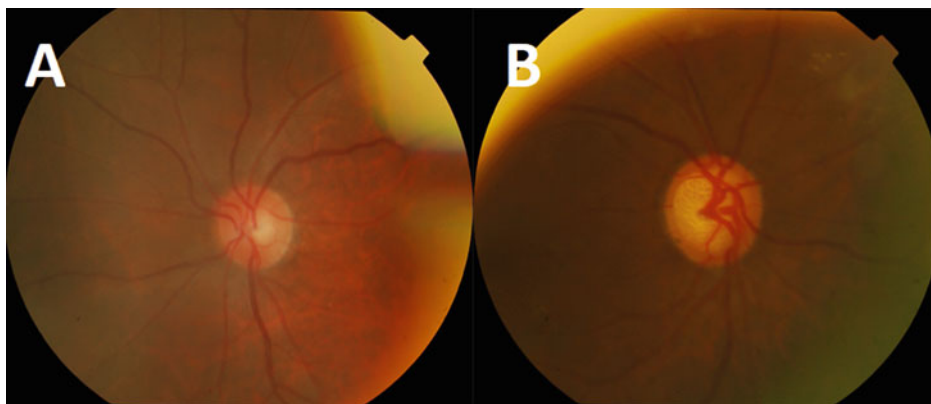


Fig. 27.1 (a) On the left, a healthy optic nerve (b) on the right, an optic nerve with glaucoma. Provided by Medical Image Processing (MIP) group, IIIT Hyderabad. (Source: Ref. [9])

$$h(i,j) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad h(i,j) = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Fig. 27.2 Mask for a high-pass filter, note that the sum of the values inside the mask sums to zero. (Source: Ref. [11])

Before training the CNN model, the retinal images were classified into two classes, glaucomatous with 229 images and normal with 283 images. For this, we used the Application Programming Interface (API) tf.keras, considered a high-level API, during the construction and training of the CNNs. Then, we imported the acquired databases through the API, and four datasets were returned for each image bank, referred to the training set, that is, the model data used for learning the CNN.

The images were passed to the network input in 32×32 format. The labels, that is, the target classifications of the images contained binary values, representing the classifications of the images: normal as value 0, or glaucomatous as value 1. Thus, we mapped each image with only one of these values. However, before starting to train the network, we performed a data preprocessing, in which we normalized the pixel values of the images for the training and testing sets in a range between 0 and 1.

Next, we configured the layers to build the CNN, so that data representations were inserted into the network could be extracted. We used a Dense layer as the input layer, containing 1024 neurons, while used Softmax as the output layer, composed of 2 neurons, which returns two probabilities. The neurons presented values capable of indicating the image classification falling into one of the two previously defined classes: normal or glaucomatous.

Before the model was ready for training, it was necessary to establish some configurations: the loss Cross Entropy

Table 27.1 Parameterization table for DenseNet, ResNet, and VGG16 networks

Filter	Network	LR	Loss function	Optimizer function	Batch size
GLCM	VGG16	0.0001	Cross entropy	Adam	32
GLCM	DenseNET	0.0001	Cross entropy	Adam	32
GLCM	ResNet	0.0001	Cross entropy	Adam	32
Malachite	VGG16	0.0001	Cross entropy	Adam	32
Malachite	DenseNET	0.0001	Cross entropy	Adam	32
Malachite	ResNet	0.0001	Cross entropy	Adam	32
RGB	VGG16	0.0001	Cross entropy	Adam	32
RGB	DenseNET	0.0001	Cross entropy	Adam	32
RGB	ResNet	0.0001	Cross entropy	Adam	32

function [12], responsible for measuring model accuracy during training; the optimizer Adam function [13], which updated the ANN based on what is obtained from the loss function; the Receiver Operating Characteristic Curve (ROC) [14] and the Area Under the Curve (AUC) [15], used to validate the DL training; in addition, the hyper-parameters such as Lr, batch size, and number of epochs.

To define the network parameters, we performed tests considering the VGG16, DenseNet, and ResNet models. We considered the best results during these tests for the rest of the experiments. Table 27.1 represents each configuration realized according to each network model analyzed.

After the parameterization of each analyzed model, we start the model training from a predefined set of images.

27.2.4 Training

During training, the model learned how to associate images with labels. Each CNN model was fed with databases containing ROI images and their respective filters.

27.2.5 Testing and Validation

Subsequently, we subjected the trained networks to the testing set, seeking to verify whether the predictions matched the defined labels. We obtained the predictions from the confusion matrix [16], and then calculated metrics for accuracy comparison and validation of the elaborated DL.

27.3 Results Analysis

This section is dedicated to the presentation and discussion of the results obtained in the experiments carried out, being divided according to the performance of the best models. Initially, we discussed the results with the stopping criterion of 50 epochs, analyzing the best models for each filter in Sect. 27.3.1. Next, we presented the results with the 30-season stopping criterion in Sect. 27.3.2. Finally, Sect. 27.3.3 describes the results of the best models under both criteria.

The experiments were conducted on a database accounting for 512 RGB images, subjected to the GLCM and Malachite filters. Figure 27.3 shows an example of the images in RGB application and with the GLCM and Malachite filters applied during the preprocessing phase.

27.3.1 Analysis of Results with 50 Seasons

Initially, we trained models individually with 50 epochs. In Table 27.2, we presented all training nets for comparative analysis, but only the best models between training runs were considered for the evaluations. Therefore, Table 27.2 shows the activation functions [17], training errors (Loss Train), validation errors (Loss Val), training accuracy (Train Acu.), testing accuracy (Test Acu.), variance of loss (Var.Loss) and

variance of accuracy (Var.Acu.), generated for each training with their respective hyperparameterizations.

As shown in Table 27.2, model 2 obtained the best result among the trainings belonging to the GLCM filter group. While compared to the other models, the result generated by model 5 achieved a higher variance in both analyses.

Figure 27.4 presents the performance of models 2 and 5 for a comparative analysis of the training and testing phases for the metrics of accuracy and loss. In agreement with Fig. 27.4, model 2 achieved the lowest variance of loss and variance of accuracy between the training and testing processes. However, this was not the case for model 5, as in all variance analyses it obtained high results during the training and testing phases. Thus, Model 5 was presented to show an example of the worst trained nets, and was disregarded for AUC metrics later, as were all other models not mentioned.

Then, among the models trained with the Malachite filter, we achieved the best results with models 6 and 7, as shown in Table 27.2. Figure 27.5 is intended to present the comparative analysis of the performance of models 6 and 7 in relation to the training and testing phases in the accuracy and loss analyses.

Thus, as models 6 and 7 showed close results, during the analyses of the loss and accuracy metrics, as noted in Fig. 27.5, the AUC metric was addressed next for a more assertive study of the results.

According to the models trained from the RGB images, presented in Table 27.2, models 11 and 12 conditioned to the VGG16 network reaching the lowest variances both in the RGB set and in the total set involving the GLCM and Malachite filters. Regarding the variance of accuracy, both models 11 and 12 obtained the lowest variances of loss in the set of all image types. However, model 11 excelled among the RGB filter training with respect to the accuracy training and accuracy testing metrics. Thus, Table 27.2 highlights in bold the performance of the best models for 50 epochs.

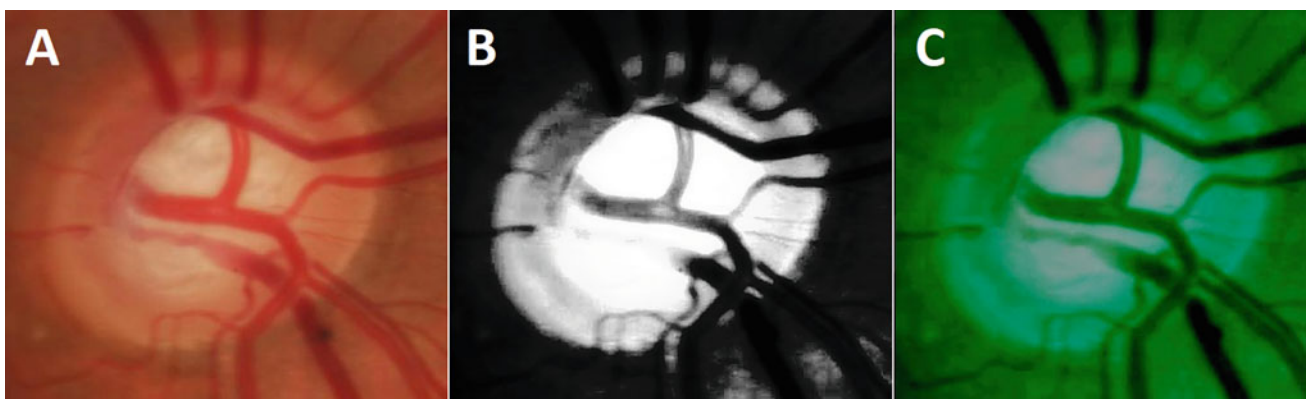


Fig. 27.3 (a) On the left, the original optical disk (RGB), in the center (b) the optical disk binarized with GLCM, and on the left (c) the optical disk with Malachite filter

Table 27.2 50-season parameterization of the DenseNet, ResNet, and VGG16 networks

Model	Filter	Network	Activation function	Loss train	Loss Val	Train Acu.(%)	Test Acu.(%)	Var. Loss	Var. Acu.(%)
1	GLCM	VGG16	SOFTMAX	0.2	0.5	94.1	78.1	0.18	64.00
2	GLCM	VGG16	SIGMOID	0.2	0.3	92.0	90.6	0.03	0.50
3	GLCM	VGG16	ReLU	0.1	0.5	40.8	50.0	0.3	21.02
4	GLCM	DenseNET	ReLU	0.6	0.2	97.9	53.1	0.6	501.76
5	GLCM	ResNet	ReLU	0.3	1.7	91.0	53.2	4.9	357.21
6	Malachite	VGG16	SOFTMAX	0.3	0.5	88.6	78.1	0.05	27.35
7	Malachite	VGG16	SIGMOID	0.3	0.5	88.4	78.1	0.12	26.63
8	Malachite	VGG16	ReLU	Nan	Nan	42.2	56.3	Nan	49.28
9	Malachite	DenseNET	ReLU	0.2	0.8	99.7	84.4	0.87	58.60
10	Malachite	ResNet	ReLU	0.03	2.5	98.8	65.6	15.78	274.40
11	RGB	VGG16	SOFTMAX	0.3	0.2	90.0	96.9	0.004	11.94
12	RGB	VGG16	SIGMOID	0.3	0.4	88.6	81.3	0.02	13.43
13	RGB	VGG16	ReLU	Nan	Nan	43.0	50.0	Nan	12.25
14	RGB	DenseNET	ReLU	0.05	0.4	99.0	87.5	0.3	33.06
15	RGB	ResNet	ReLU	0.2	1.9	95.0	63.0	6.97	256.00

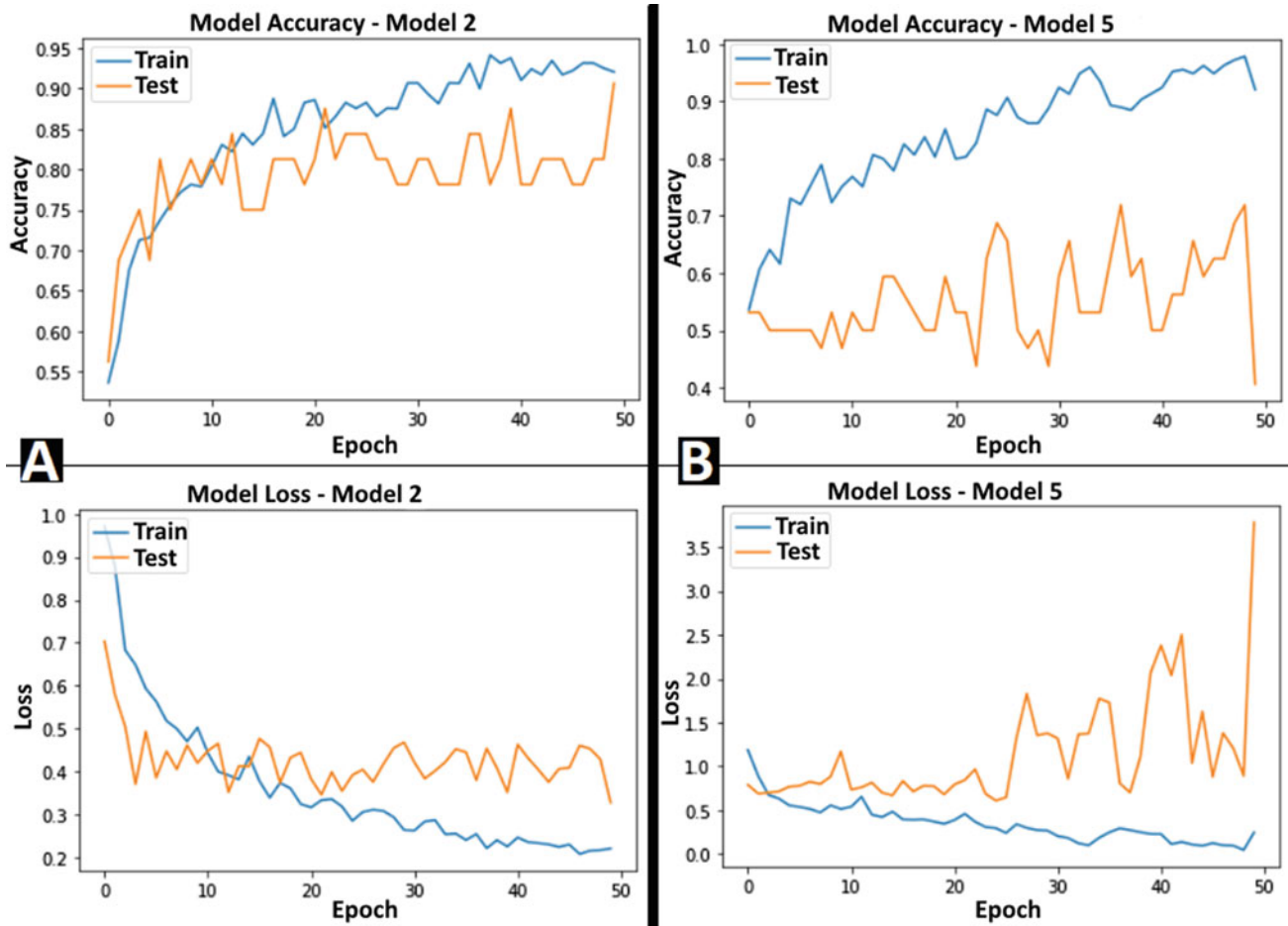


Fig. 27.4 (a) On the left, training and testing of model 2 at 50 epochs with a GLCM filter, (b) on the right, training and testing of model 5 at 50 epochs with a GLCM filter

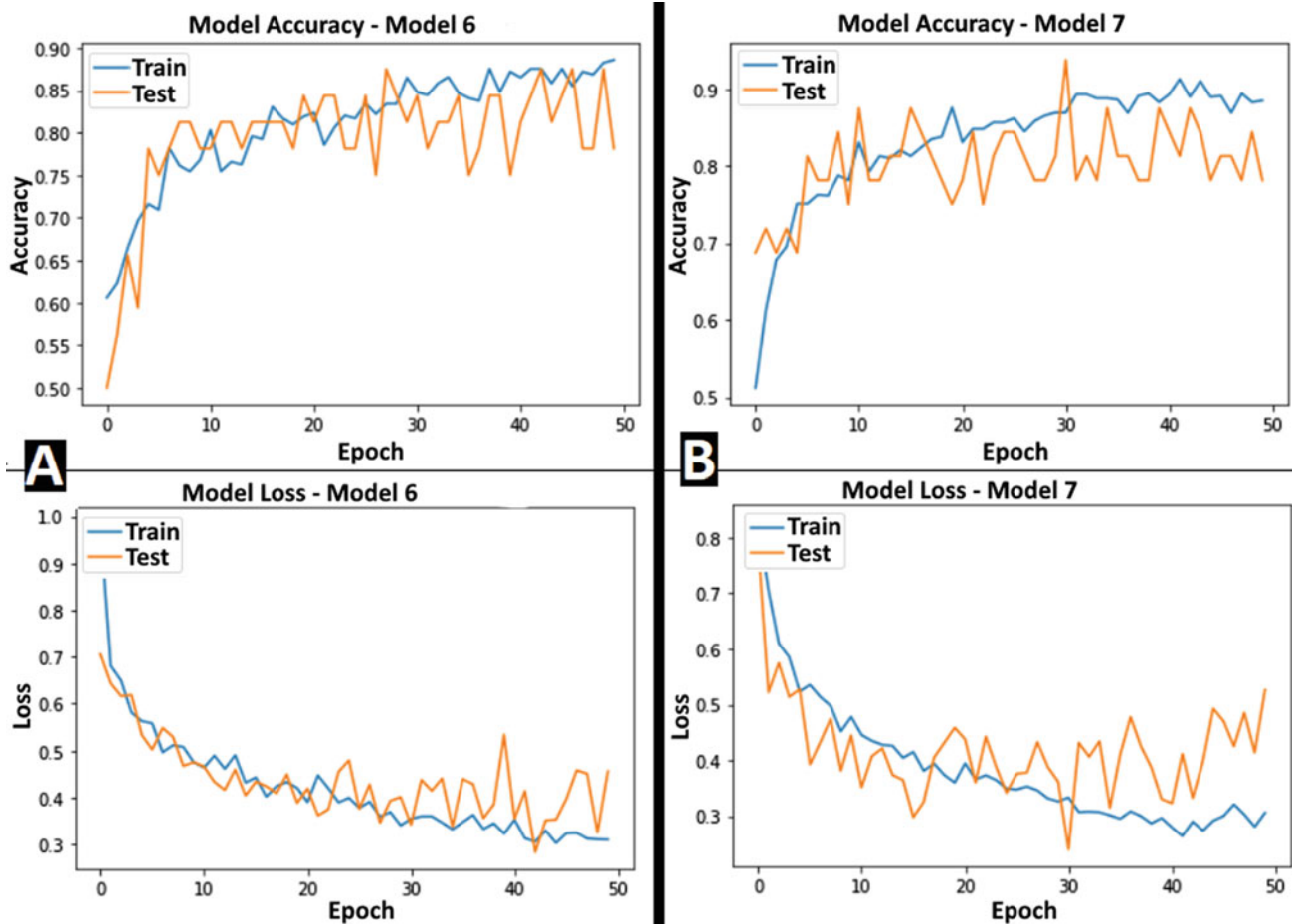


Fig. 27.5 (a) On the left, training and testing of model 6 at 50 epochs, (b) on the right, training and testing of model 7 at 50 epochs

Table 27.3 Parameterization in 30 epochs of the DenseNet, ResNet and VGG16 networks

Model	Filter	Network	Activation function	Loss train	Loss Val	Train Acu. (%)	Test Acu. (%)	Var. Loss	Var. Acu. (%)
1	GLCM	VGG16	SOFTMAX	0.3	0.5	89.4	78.1	0.01	31.92
2	GLCM	VGG16	SIGMOID	0.7	0.7	58.8	53.1	0.00	8.12
3	GLCM	VGG16	ReLU	Nan	Nan	45.3	53.1	Nan	15.21
4	GLCM	DenseNET	ReLU	0.0	1.7	99.0	62.5	0.67	333.06
5	GLCM	ResNet	ReLU	0.1	0.9	97.3	46.9	0.18	635.04
6	Malachite	VGG16	SOFTMAX	0.4	0.4	82.8	84.4	0.00	0.64
7	Malachite	VGG16	SIGMOID	0.4	0.4	84.0	84.4	0.00	0.04
8	Malachite	VGG16	ReLU	Nan	Nan	45.7	50.0	Nan	4.62
9	Malachite	DenseNet	ReLU	0.0	0.7	99.3	84.4	0.10	55.50
10	Malachite	ResNet	ReLU	0.1	1.3	95.2	59.4	0.33	320.41
11	RGB	VGG16	SOFTMAX	3.43	3.00	85.5	93.8	0.04	17.14
12	RGB	VGG16	SIGMOID	3.81	3.61	81.7	87.5	0.01	8.53
13	RGB	VGG16	ReLU	Nan	Nan	43.0	50.0	Nan	12.25
14	RGB	DenseNET	ReLU	0.49	7.00	98.4	78.1	10.60	102.82
15	RGB	ResNet	ReLU	1.14	16.35	95.9	65.6	57.88	229.83

27.3.2 Analysis of Results with 30 Seasons

In this section, we adopted the forced stopping criterion with 30 epochs for the purpose of comparing the training

generated in Sect. 27.3.1. Initially, we trained the 15 models considering only the change in the number of epochs and maintaining all other hyperparameters defined in Table 27.2 from Sect. 27.3.1. Therefore, for an in-depth analysis, we

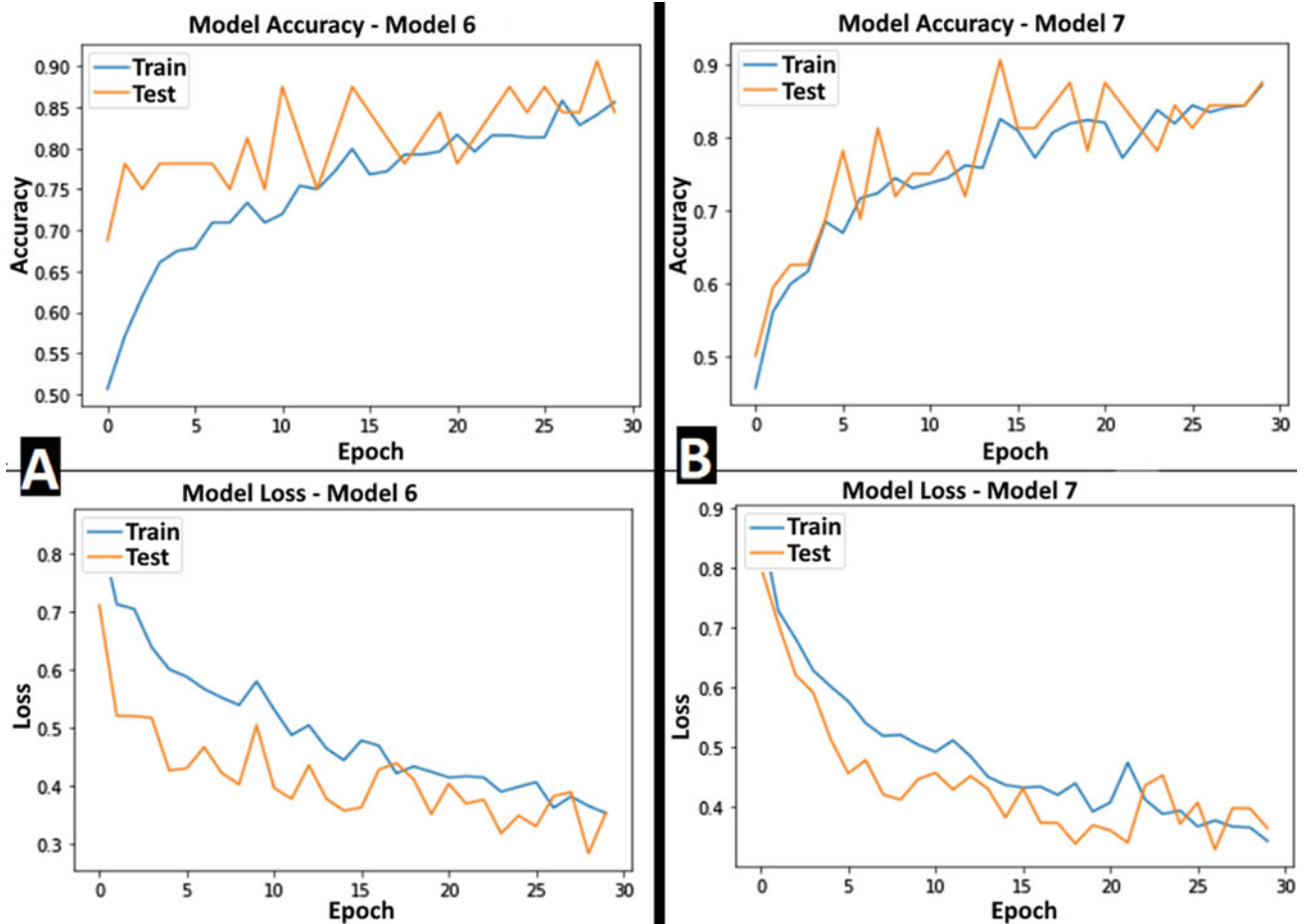


Fig. 27.6 (a) On left, training and testing of model 6 at 30 epochs with the Malachite filter, (b) on right, training and testing of model 7 at 30 epochs with the Malachite filter

generated the Table 27.3, presenting the results according to their respective network architectures and activation functions. As shown in Table 27.3, models 6, 7, and 11 obtained the best results among the 15 training run developed. Both the variance of the loss and the accuracy decreased compared with their respective models at 50 epochs.

Figure 27.6 is intended to show the models 6 and 7 performance during the execution with respect to training and testing performance for accuracy and loss analysis.

We analyzed the variation between training and testing accuracy in Fig. 27.6. The accuracy variances of models 6 and 7 obtained the lowest results among the models presented in Table 27.3, showing the superiority of the Malachite filter over the GLCM filter during the analysis over 30 epochs. Thus, both model 6 and model 7 required analysis by the AUC metric to identify the best model. Table 27.3 represented in bold the best models trained with 30 epochs.

27.3.3 Final Results

For the final evaluation of the results, we adopted an AUC metric, in which the closer the value reaches 100%, the better the effectiveness of the model in making correct predictions. In Fig. 27.7 was possible to compare the best related models at epochs 50 and 30.

As illustrated in Fig. 27.7, we achieved the best AUC performance during training for 50 epochs with model 11, developed with VGG16 architecture, Softmax activation function and RGB image. Model 11 had a good efficiency in predicting the images, finding a correlation between the label and the model classification. Consequently, we observed in the final confusion matrix, shown in Fig. 27.8a, that the classification hit rate was 82%. As per the AUC results shown in Fig. 27.7, the Malachite filter showed the second place in performance all parameterizations and epochs addressed here, both in VGG16 architecture, Sigmoid and Softmax

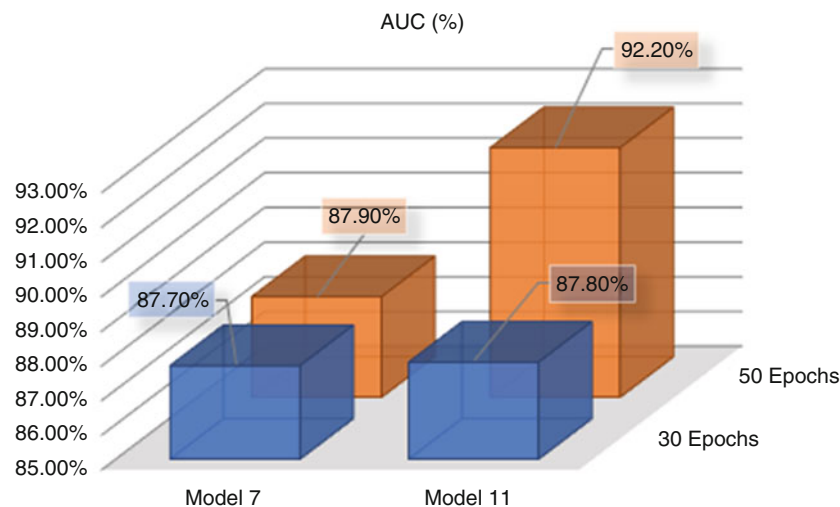


Fig. 27.7 The best models in this study

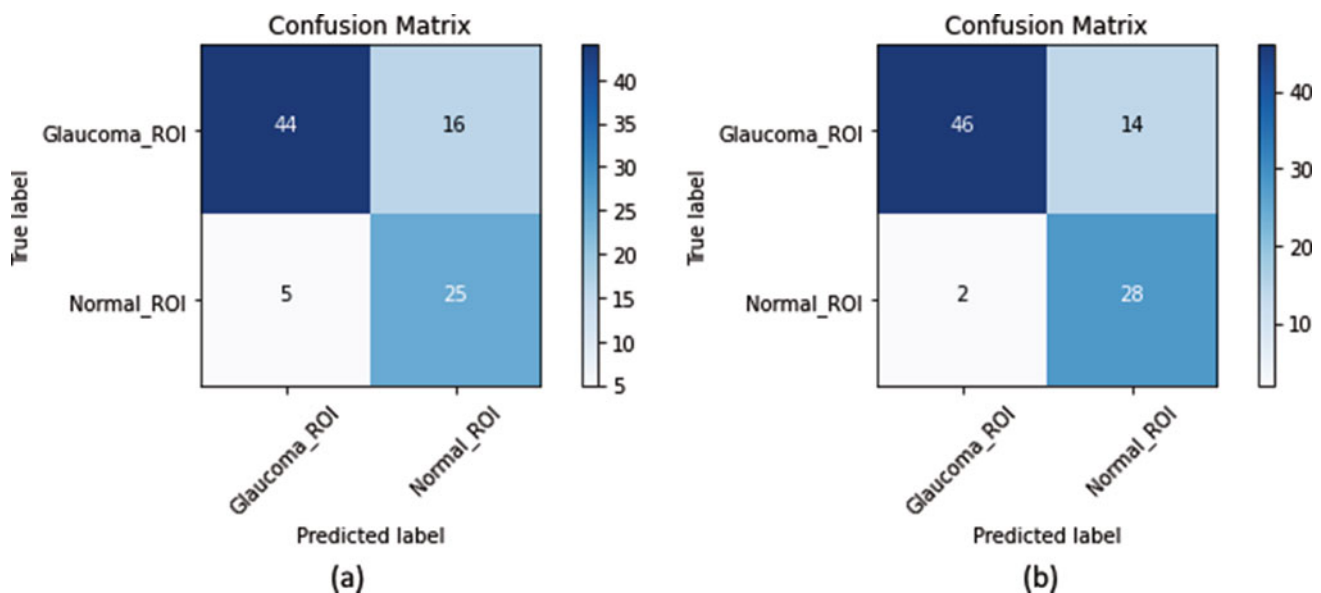


Fig. 27.8 Model 11 – Confusion Matrix for (a) 50 epochs and (b) 30 epochs

activation function, presenting an AUC of 87.9%. Figure 27.9 represents the confusion matrix of model 7 during training over 50 epochs.

Considering overall classification among the training developed during the previous sections, Table 27.4 shows the best results of this study. Therefore, after the result analysis, we concluded that Softmax and Sigmoid activation functions presented the best performances, while the networks with the ReLU function performed below expectations.

In this study, model 11 at epoch 30 achieved the third best performance over all results presented in Fig. 27.7, with an AUC of 87.8%. As presented in Fig. 27.8b, it presented a classification hit rate of 76.7

27.4 Conclusion

In this work, we applied concepts of Digital Image Processing (DIP) and Convolutional Neural Networks (CNN) for experimental studies and analysis in the identification of glaucoma optic neuropathy. The studies performed sought to compare different DIP approaches and CNN models during the training and validation phases of the networks, seeking to evaluate the model by the metric accuracy, confusion matrix and AUC.

From the results obtained, we observed that although the ReLU activation function is best suited for networks with outputs 0 and 1, they achieved the worst results on the test-

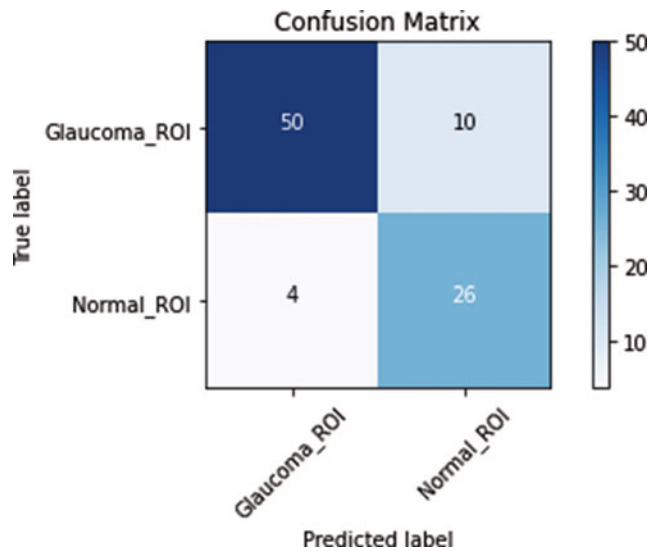


Fig. 27.9 Model 7 – Confusion Matrix for 50 epochs

Table 27.4 Metrics of the best performing networks

Ranking	Filter	Netwrk	Func. activation	Acu. (%)	AUC (%)	Epochs
1	RGB	VGG16	Softmax	82.2	92.2	50
2	Malachite	VGG16	Sigmoid	84.4	87.9	50
3	RGB	VGG16	Softmax	76.7	87.8	30

ing clusters performed. We found the best results using the VGG16 network, with Softmax activation function, where it achieved a satisfactory AUC among the experiments. The Malachite network, created by the authors specifically for this study, showed a high performance during the AUC metric evaluation, when used with the Sigmoid activation function. However, when applying the Softmax activation function, the RGB image had a better AUC compared to the other models, as it obtained a higher accuracy in classifying the images.

Therefore, considering the results obtained in this study, we recommend the developed tool to help ophthalmologists during the process of glaucoma pattern detection. For future works, we suggested the expansion of the database, in addition to readjustments of hyperparameters to reduce the cost function during training and validation of network

models. Furthermore, we suggest the development of the best models with the Retinographic device, currently used to help ophthalmologists to detect glaucoma.

References

1. M. Ávila, J. Ottaiano, C. Umbelino, et al., *Censo oftalmológico: As condições de saúde ocular no Brasil [Internet]* (CBO, São Paulo, 2019)
2. G. Kande, P. Mittapalli, Segmentation of optic disk and optic cup from digital fundus images for the assessment of glaucoma. *Biomed. Signal Process. Control* **24**, 34–46 (2016)
3. M. Ponti, G. da Costa, Como funciona o deep learning, arXiv preprint arXiv:1806.07908 (2018)
4. S. Jian, H. Kaiming, R. Shaoqing, Z. Xiangyu, Deep residual learning for image recognition, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2016), pp. 770–778
5. G. Huang, Z. Liu, L. Maaten, K. Weinberger, Densely connected convolutional networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017), pp. 4700–4708
6. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556 (2014)
7. M. Goldbaum, *Structured Analysis of the Retina (stare)* (2000)
8. E. Decenciere, D. Etienne, Z. Xiwei, et al., Feedback on a publicly distributed image database: The Messidor database. *Image Anal. Stereol.* **33**(3), 231–234 (2014)
9. A. Chakravarty, G. Joshi, S. Krishnadas, J. Sivaswamy, A. Tabish, A comprehensive retinal image dataset for the assessment of glaucoma from the optic nerve head analysis. *JSM Biomed. Imaging Data Papers* **2**(1), 1004 (2015)
10. T. Barrier, S. Brahnam, S. Ghidoni, E. Menegatti, L. Nanni, Different approaches for extracting information from the co-occurrence matrix. *PloS One* **8**(12), e83554 (2013)
11. M. Marengoni, S. Stringhini, Tutorial: Introdução à visão computacional usando opencv. *Revista de Informática Teórica e Aplicada* **16**(1), 125–160 (2009)
12. A. Géron., *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow* (O’Reilly Media, Inc., 2022)
13. D. Kingm, J. Ba, Adam: A method for stochastic optimization, in *International Conference on Learning Representations*, (2015), pp. 1–13
14. T. Fawcett, An introduction to ROC analysis. *Pattern Recognit. Lett.* **27**(8), 861–874 (2006)
15. D. Hand, W. Krzanowski, *ROC curves for continuous data* (Chapman and Hall/CRC, 2009)
16. I. Teixeira, Deep learning aplicado à automação de balanças comerciais de hortifrutis (2018)
17. J. Richards, J. Richards, *Remote Sensing Digital Image Analysis: An Introduction*, vol 3 (Springer, 1999), pp. 10–38.1