Computer Science & Engineering: An International Journal (CSEIJ), Vol 12, No 6, December 2022

CONVOLUTIONAL NEURAL NETWORK BASED RETINAL VESSEL SEGMENTATION

Savithadevi M

Research Scholar, National Institute of Technology, Tiruchirappalli

ABSTRACT

In human eye, the state of the blood vessel is a crucial diagnostic factor. The segmentation of blood vessel from the fundus image is difficult due to the spatial complexity, adjacency, overlapping and variability of blood vessel. The detection of ophthalmic pathologies like hypertensive disorders, diabetic retinopathy and cardiovascular diseases are remain challenging task due to the wide-ranging distribution of blood vessels. In this paper, Stacked Autoencoder and CNN (Convolutional Neural Network) technique is proposed to extract the blood vessel from the fundus image. Based on the experiments conducted using the Stacked Autoencoder and Convolutional Neural Network gives 90% & 95% accuracy for segmentation.

KEYWORDS

Stacked Autoencoder, Convolutional Neural Network (CNN), retinopathy, blood vessel, supervised learning method.

1. INTRODUCTION

The primary goals of supervised learning methods are to retrieve typically important features from labeled data, identify and eliminate input redundancies, and preserve only the most important aspects of the information in robust and exclusionary representations. Many scientific and commercial applications have also routinely utilized supervised methods. [1]. For developing a robust screening system for diabetic retinopathy, the automatic analysis of retinal vessel topography is used [2]. A Convolutional Neural Network (CNN) is a standard multi-layer neural network that consists of one or even more convolution layers, a pooling layer, and optionally one or more fully connected layers. Another advantage of CNN is it consists of a small number of units compared to the other deep networks which consist of many hidden layers. Autoencoder is a supervised technique where the number of inputs and targets might be given. It is trained to copy its input to its output and it also consists of hidden layers. Autoencoder is a network that mainly consists of two parts namely an encoder function and a decoder function. The main application of the autoencoder is dimensionality reduction, classification, and information retrieval.

2. LITERATURE REVIEW

In 2010, quinmu et al, presented a method to locate the vessel's central lines using the radial projection method. For the extraction of the major structure of blood vessels, the supervised classification method is used. Marin et al (2011), proposed a supervised neural network-based technique. For training and classification, a neural network with multilayer feed-forward is utilized. In different conditions with multiple images, this method improves the robustness. Holbura.et.al (2012), presented a new method by combining Support Vector Machine (SVM) and neural network technique over the same feature set. The weighted decision fusion is used to improve classification accuracy. In 2014 Mehrotra et al, presented a method in which they use morphological methods such as top and bottom hat transformations to highlight the blood vessels in the retinal image.Tan.et.al., (2017) proposed a method of supervised learning that occurs spontaneously, segments, and effusion the hemorrhages and micro-aneurysms, which uses ground truth data for segmenting vessels. Lin.et.al., (2018) present a work in which a deep learning method with conditional random field and holistically-nested edge detection is added to perform the vessel segmentation in the retinal image.

Literature says that still there is a hope for improving accuracy.

3. PROPOSED METHOD

In this work, propose Convolutional Neural Network and a Stacked Autoencoder based supervised learning method to extract the blood vessel from the fundus image. In this let's consider several pre-processing techniques to accurately recognize the blood artery in the fundus image. The proposed method can outperform the existing methods based on the accuracy of classification. Figure 1 depicts the suggested method's architecture.

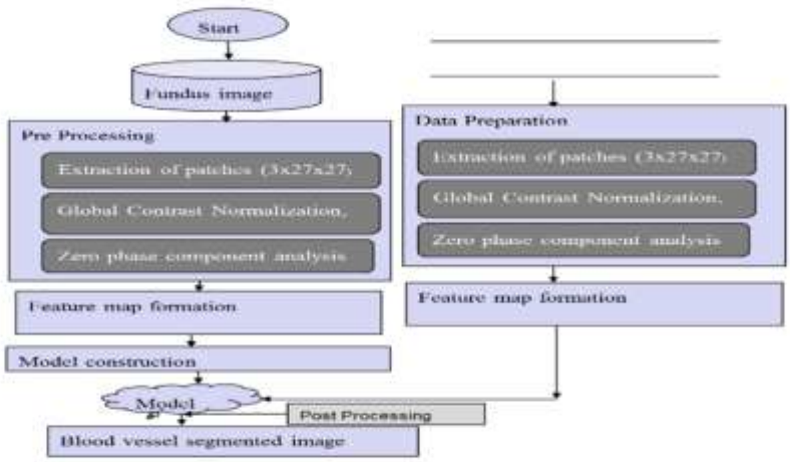

Fig.1. Architecture of the overall proposed system

3.1. Dataset Description

The images for the DRIVE database come from a screening test in the Netherlands for diabetes mellitus. 400 diabetic patients aged 25 to 90 make up the screening population. A total of 40 images were chosen at random, from which 33 showed no symptoms of diabetes mellitus and 7 showed mild initial symptoms. JPEG has been employed to compress each image.

3.2. Patch Extraction

From the original image (Fig 2), the patch size of b x b (where b=27) is being extracted to process the image. The patch can be extracted in random and sequential ways in the DRIVE database. For training, it generates randomly 900 patches from the training image. While for testing, 400 patches from the test image are generated sequentially. Using binary classification, every pixel is assigned with a positive value of 1 (for vessel pixel) and a negative value (for non-vessel pixel). The pixel centered can be identified and based on that center the pixel around it can be extracted from the fundus image of size 27X27. Since it is an RGB image it has 3 channels. The same process is performed in all the channels in the image and extracts 27X27X3 size of the patch.

3.3. Global Contrast Normalization

It can be seen that brightness varies amongst the images in the collection by looking at the image in the DRIVE database. Global contrast normalization is employed to get around problem. This means that the standard deviation of each patch's components is divided by the mean, which isthen removed from the result.

3.4. Zero phase Component Analysis (zca whitening)

Input is typically "whitened" to reduce repetition. The neighboring pixel in the input image are almost correlated to each other. So, in order to remove the redundancy in the input image, ZCA whitening is applied in the image. In this an epsilon value of 0.001 is assigned in order to avoid the data elements to be divided by zero.

Let the centered pixel is stored in the data matrix A with data points in row and features in columns. Then A has the same number of rows as samples and columns as 3 X 27 X 27=2187 is equal to features. The covariance matrix $\Sigma = \frac{1}{b}\sum_{j=1}^{b} A\,A^T$ the diagonal of Λ has eigen values and the columns of N has eigenvectors, so that is represented as,

$\Sigma = N\Lambda N^T$. Then N be an orthogonal matrix (rotation/reflection) and N^T provides a rotation necessary to decorrelate the data.
The general PCA whitening is given by

$$X_{PCA}=\Lambda^{-1/2}N^T \quad \text{-------------------} \ (1)$$

The ZCA whitening is $X_{PCA} = N\, X_{PCA}$ then it can be represented as

$$X_{ZCA}=N\,\Lambda^{-1/2}N^T = \Sigma^{-1/2} \quad \text{--------------------} \ (2)$$

3.5. Model Construction

3.5.1. Convolutional Neural Network (cnn)

The convolutional neural network is divided into two stages with various layers that generate low and generally high features. In contrast to traditional convolutional neural networks, layer-skipping in proposed network. (i.e., the classifier is inputted by the output of the two stages), This allows the classifier to use both high-level global and low-level local features. The relatively high features generate holistic explanations of the blood vessel, while the low-level features aim to accurately recognize the blood vessel. Both of them are advantageous for Retinal image classification.

Semi-supervised convolutional neural networks are proposed. Sparse Laplacian filter studying is used to learn the network's filters with such a great number of unsupervised patches in order to obtain deep as well as distinct data on blood vessels. The output layer is the soft-max classifier layer, which was trained using multi-task learning with a small number of labelled patches. The network can automatically learn good features to classify the blood vessel type for a given retinal image. By choosing the label with greatest chance, the category of patch could be predicted.

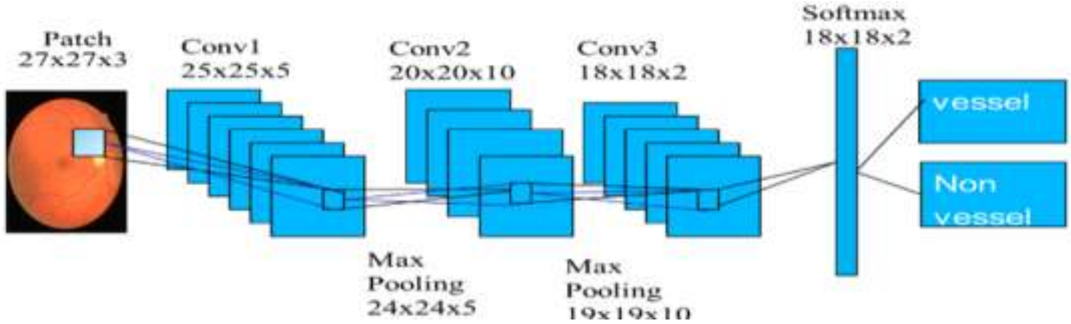

Fig.2. CNN Architecture

3.5.2. CNN architecture for vessel identification

Convolution is a mathematical term that describes the method of repeatedly applying one function across the output of another function. It means applying a 'filter' to that of an image throughout all feasible offsets in this context. A filter is composed of a layer of weight vectors, with the input resembling a length of 2 patches and thus the output resembling a single unit. Since this filter is used repeatedly, the probable results in connectivity look like a sequence of interlinking receptive fields that map to a matrix of filter outputs.

3.5.3. Local Connectivity

It is impractical to connect neurons in the previous quantity to all neurons once dealing with multi-dimensional inputs such as images. Rather than, each neuron is only attached to a subset of input volume. The spatial extent of this connectivity is a hyperparameter known as the neuron's receptive field.

3.5.4. Max Pooling Layer

The stride size, pool size, adding, and layer name contain the maximum pooling layer. Down

sampling is carried out by max pooling layer by splitting the input into rectangular pooling areas and estimating the maximum of every region.

3.5.5. Fully Connected Layer

All neurons connected to the fully connected layer connect all neurons in the previous layer (whether pooling, fully connected, or convolutional) to each neuron it has. The class scores will be calculated by the fully-connected layer, resulting in values assigned to a class score.

3.5.6. Softmax Layer

The output layer of the convolutional neural network is the softmax classifier. The softmax classification layer receives the feature vector obtained by the previous layers as input and then outputs the based-on probability vector. The final feature vector is fed to the softmax layer, which differentiates between vessel and non-vessel patches.

ALGORITHM FOR CNN-BACK PROPAGATION

Inputs: network, a multilayer network
Examples, a set of input/output pairs
α, the learning rate
Repeat
For each e in examples do
/*compute the output for this example*/
O – Run-Network (network,I^e)
/* compute the error and Δ for units in the output layer */
$$\text{Error}^e \leftarrow T^e - O$$
/* update the weights leading to the output layer */
$$W_{j,i} = W_{j,i} + \alpha * \text{Error}_i^e * g'(in_i)$$
For each subsequent layer in network do
/* Compute the error at each node */
$$\Delta_j \leftarrow g'(in_j) \sum_i W_{j,i} \Delta_i$$
/* Update the weights leading into the layer */
$$W_{k,j} = W_{k,j} + \alpha * I_k * \Delta_j$$
End
End
Until network has converged
Return network

ALGORITHM: AUTOENCODER

Input: patch dataset extracted from drive database
Output: Trained autoencoder, classified dataset
Begin
Set Hidden layer of size = 7
L2 weight regularizer = 0.001
Sparsity regularizer = 4
Sparsity proportion = 0.05
Train encoder
$f(z) = \frac{1}{1+e^{-z}}$ (Logsig function)
Select linear transfer function as activation function for the decoder.
$$f(z) = Z$$
Extract the features in the hidden layer.
Train a softmax layer for classification using the features, from the second autoencoder.
Stack the encoders and the softmax layer to form a deep network.
The cross entropy is given by
$$E = \frac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{k} t_{ij} \ln y_{ij} + (1 - t_{ij})\ln(1 - y_{ij})$$
Back propagate the error and update the weight matrix.
Use the model to identify the vessel or non-vessel types.
End

Function BACK-PROPAGATION-UPDATE (Network(N), examples, a) returns a network with modified weights

Autoencoder

3.5.7. Training Single Layer Autoencoder

From the DRIVE database, 900 patches are extracted from each training image of 20 images. Thus train an autoencoder with 18,000 patches of size 27 x 27 x 3 given to the input layer of an autoencoder as a row vector. The encoding and decoding function is used to rebuild the input image with minimized reconstruction error which can be calculated by cross entropy using Stochastic Gradient Descent (SGD). For encoding, the logistic sigmoid function is used and for decoding, purelin function is used. In autoencoder the input and target value must be the same.

If an autoencoder obtains a vector $x \in R^{D_x}$, as input, the encoder routes the vector x with another vector $Z \in R^{D^{(1)}}$ as follows:

$Z^{(1)} = h^{(1)}(W^{(1)}x + b^{(1)})$ ------------------------------ (3)

The first layer is noted by the superscript (1). $h^{(1)} : R^{D^{(1)}} \rightarrow R^{D^{(1)}}$ is a transfer function for the encoder, $W^{(1)} \in R^{D^{(1)} X\, D(x)}$ is a weight matrix, and $b^{(1)} \in R^{D^{(1)}}$ is a bias vector. The decoder then maps the encoded representation z back into an approximate of the original image vector, x, as follows:

$\hat{x} = h^{(2)}(w^{(2)}x + b^{(2)})$ ------------------------------------- (4)

The superscript (2) denotes the second layer

$h^{(2)}$: $R^{Dx} \rightarrow R^{Dx}$ is the decoder's transfer function,
$W^{(1)} \in R^{Dx} \times D^{(1)}$ is a weight matrix, and
$b^{(2)} \in R^{Dx}$ is a bias vector.

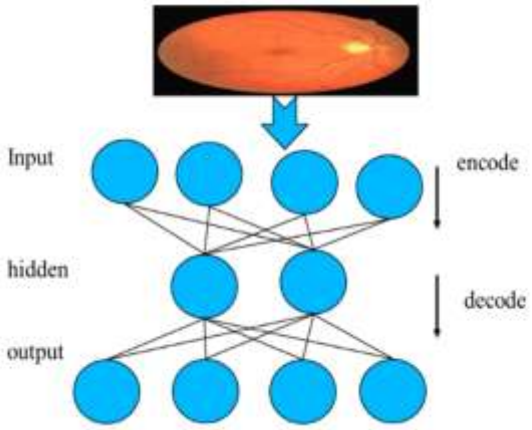

Fig.3. Stacked Autoencoder

3.5.8. Training the Second Layer Autoencoder

The stacked autoencoder can be obtained by serially arranging the layers such as the first layer's output is fed into the second layer's input, and so on. Thus, the latent characteristics from the initial autoencoder are fed through into second layer as input. This process will go on for the subsequent hidden layers. Finally, the last layer, latent feature is considered as a final feature vector for the input image.

Testing

Following DRIVE database training, the test image is given as an input to the autoencoder network for testing and then the accuracy of correct prediction of the vessel and non-vessel patches from the test image is found. The accuracy rate can be predicted using a confusion matrix.

The test image is divided into patches. The image can be divided into patches sequentially in order for testing. The tested image result is constructed as a matrix. A value 1 for is assigned for the patch that is correctly classified as vessel and a value 0 for the patch that is not classified as vessel patch. Likewise, a 20 X 20 matrix for the whole image is obtained. After this construct a mask for the value 1 and value 0. For the value 0, multiply the corresponding image patch with zeros and for value 1, multiply the corresponding image patch with ones.

4. Performance Evaluation

The performance can be measured by using

True positive: A true positive test result is one that detects a condition when it exists.
True negative: A true negative test result is one that does not detect the condition when it is not present.

False positive: A false positive test result is one that detects a condition that does not exist.
False negative: A false negative test result is one that fails to detect a condition when it exists.
The performance of networks tested on benchmark-specific test sets in terms of area under accuracy. (Acc), sensitivity (Sens), specificity (Spec), defined as

$$\text{Acc}=\frac{TP+TN}{TP+TN+FP+FN} \quad \text{----------------} \; (5)$$

$$\text{Sens}=\frac{TP}{TP+FN} \quad \text{-----------------} \; (6)$$

$$\text{Spec}=\frac{TN}{TN+FP} \quad \text{------------------} \; (7)$$

4.1. Experimental Results and Discussion

In matlab 2016a, experiments are conducted by implementing the algorithm using the following functions softmax, conv, pool, encoder and decoder functions.
A sample image from DRIVE database in which patch extraction, GCN, ZCA is performed is shown in Fig.4.

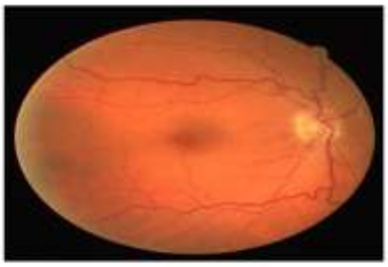

Fig.4. Sample image from DRIVE database

Fig.5. Training patches extracted from the DRIVE Database

Patches of size 27 x 27 extracted from the sample image in DRIVE database is shown in the Fig.5.

Fig.6. Training patches after applying GCN transformation

Fig.7. Training patches after applying ZCN whitening Transformation

Experiments are carried out by altering the hidden units during training, with the results summarized in Table 1. The results demonstrate that raising the hidden units size increases the accuracy to a certain level, but after enhancing the hidden unit's size to 10, the accuracy decreases. So, the hidden layer size is fixed as 7.

Table 1. Performance of training and testing with various hidden layers using Autoencode

Hidden layer	**Best training epoch**	**Training data classification accuracy**	**Testing data classification accuracy**
HL=5	Epoch=203	Correctly classified =70.7%	Correctly classified =55.0%
HL=7	Epoch=520	Correctly classified =90.7%	Correctly classified =90.0%
HL=10	Epoch=520	Correctly classified =90.7%	Correctly classified =57.0%
HL=15	Epoch=388	Correctly classified =90.7%	Correctly classified =70.0%

Experiments are carried out in the network by changing the number of both training and testing patches, and the results are presented in Table 2. The accuracy will increase when the training samples is increased and the accuracy will decrease when the training patches get decreased.

Table 2. Performance of training and testing with various patches using Autoencoder

Training images represented in patches	**Best training epoch**	**Training data classification accuracy**	**Testing data classification accuracy**
70% -training 30%-testing	Epoch=96	Correctly classified =99.0%	Correctly classified =81.8%
60%-training 40%-testing	Epoch=92	Correctly classified =99.0%	Correctly classified =53.0%
80%-training 20%-testing	Epoch=1000	Correctly classified =90.7%	Correctly classified =75.0%

Table 3. CNN architecture

Name	**Size**	**Kernel size**
convolution	25x25x5	(3,3,3,5)
Max Pooling	24x24x5	2 x 2
Convolution	20x20x10	(5,5,5,10)
Max Pooling	19x19x10	2 x 2
Convolution	18x18x2	(2,2,10,2)
Softmax	18x18x2	------

Table 4. Performance of training and testing with various patches using CNN

No. of training images	**Minimum error epoch**	**Training data classification accuracy**	**Testing data classification accuracy**
70% - training30%-testing	Epoch=70	Correctly classified =99.0%	Correctly classified =88.8%
60%-training40%-testing	Epoch=70	Correctly classified =99.0%	Correctly classified =73.0%
80%-training20%-testing	Epoch=70	Correctly classified =99.0%	Correctly classified =95.0%

The proposed method performance is compared with existing literature and it is tabulated in Table 5.

After constructing model test images are divided into patches and vessel patches are identified for model construction.
Then, the mask is multiplied with test image and the vessel are extracted from it. Finally, the extracted vessel from the original image can be obtained and shown in Fig 8 & 9.

Autoencoder model is constructed using 1800 patches and tested with 400 patches. Accuracy obtained is 90% by varying the hidden layers. Convolutional Neural Network model is also constructed using 1800 patches and tested with 400 patches. Accuracy obtained is 95% by varying the training patches. In future more, number of patches can be trained to improve accuracy.

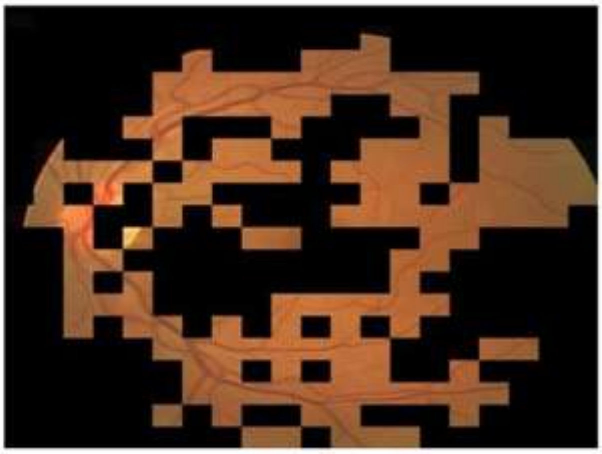

Fig.8. Segmented blood vessel from the test image using Autoencoder

Fig.9. Segmented blood vessel from the test image using CNN

5. CONCLUSION AND FUTURE WORK

The segmentation of blood vessels from image data is a difficult issue in medical imaging. The learned features of the network are discriminative enough just to perform well even in complex background. It should be observed that the proposed model outperformed other methods despite the absence of blood vessels. Basically, CNN requires higher processing systems since it extracts a greater number of features which is very useful in image classification, whereas Autoencoder does not require such high-end systems. The performance of extraction task was examined using Convolutional Neural Network technique and stacked auto encoder technique and achieved the accuracy of 95% & 90%. Young radiologist can use this as a tool for cross reference their initial prediction. In future several deep networks concept can be applied to improve the accuracy.

REFERENCES

[1] Jonathan Masci, Ueli Meier, Dan Cire¸san, and J¨urgen Schmidhuber," Stacked Convolutional Auto- Encoders for Hierarchical Feature Extraction", Springer, LNCS 6791, pp. 52–59, 2011.

[2] Studisull'Intelligenza Artificiale (IDSIA) Lugano, Switzerland {jonathan,ueli,dan,juergen} @idsia.ch

[3] QinmuPeng, Guangxi Peng, DuanquanXu, Xinge You, Baochuan Pang, "A Fast Approach to Retinal Vessel Segmentation", in: Chinese Conf. Patt. Recog. –CCPR, pp. 1 – 5,2010.

[4] Marin, A. Aquino, M.E. Gegundez-Arias, J.M. Bravo, "A New Supervised Method for Blood Vessel Segmentation in Retinal Images by using Gray-Level and Moment Invariants Based Features",IEEE Trans. Med. Imag., Vol. 30, pp 146–158,2011.

[5] Holbura, M. Gordan, A. Vlaicu, I. Stoian, D. Capatana, "Retinal vessels segmentation using supervised classifiers decisions fusion", IEEE Int. Conf. on Automation Quality and Testing Robotics, pp. 185 – 190,2012.

[6] Mehrotra, S. Tripathi, K. K. Singh, P. Khandelwal, "Blood Vessel Extraction for retinal images using morphological operator and KCN clustering", IEEE Int. Adv. Comput. Conf. – IACC'2014, pp. 1142 – 1146,2014.

[7] Paweł Liskowski, Krzyszt of Krawiec, "Segmenting Retinal Blood Vessels with Deep Neural Networks", IEEE Transaction on Medical Imaging 2016.