applied sciences

MDPI

Article

A Convolutional Autoencoder Approach for Boosting the Specificity of Retinal Blood Vessels Segmentation

Natalia Nikoloulopoulou [1], Isidoros Perikos [1,2,*], Ioannis Daramouskas [1,2], Christos Makris [1], Povilas Treigys [3] and Ioannis Hatzilygeroudis [1]

[1] Computer Engineering and Informatics Department, University of Patras, 26504 Patras, Greece
[2] Computer Technology Institute and Press "Diophantus", 26504 Patras, Greece
[3] Institute of Data Science and Digital Technologies, Vilnius University, 01513 Vilnius, Lithuania
* Correspondence: perikos@ceid.upatras.gr

Abstract: Automated retina vessel segmentation of the human eye plays a vital role as it can significantly assist ophthalmologists in identifying many eye diseases, such as diabetes, stroke, arteriosclerosis, cardiovascular disease, and many other human illnesses. The fast, automatic and accurate retina vessel segmentation of the eyes is very desirable. This paper introduces a novel fully convolutional autoencoder for the retina vessel segmentation task. The proposed model consists of eight layers, each consisting of convolutional2D layers, MaxPooling layers, Batch Normalisation layers and more. Our model has been trained and evaluated on DRIVE and STARE datasets with 35 min of training time. The performance of the autoencoder model we introduce is assessed on two public datasets, the DRIVE and the STARE and achieved quite competitive results compared to the state-of-the-art methods in the literature. In particular, our model reached an accuracy of 95.73, an AUC_ROC of 97.49 on the DRIVE dataset, and an accuracy of 96.92 and an AUC ROC of 97.57 on the STARE dataset. Furthermore, our model has demonstrated the highest specificity among the methods in the literature, reporting a specificity of 98.57 on the DRIVE and 98.7 on the STARE dataset, respectively. The above statement can be noticed in the final blood vessel segmentation images produced by our convolutional autoencoder method since the segmentations are more accurate, sharp and noiseless than the result images of other proposed methods.

Keywords: deep learning; image segmentation; blood vessel detection; retinal fundus images; convolutional autoencoder; medical imaging

check for updates

Citation: Nikoloulopoulou, N.; Perikos, I.; Daramouskas, I.; Makris, C.; Treigys, P.; Hatzilygeroudis, I. A Convolutional Autoencoder Approach for Boosting the Specificity of Retinal Blood Vessels Segmentation. *Appl. Sci.* **2023**, *13*, 3255. https://doi.org/10.3390/app13053255

Academic Editor: Jan Egger

Received: 21 January 2023
Revised: 11 February 2023
Accepted: 17 February 2023
Published: 3 March 2023

1. Introduction

The eyes are very sensitive and numerous diseases are associated with them. Many critical human diseases can manifest in the retina and originate from the eye, brain, or cardiovascular system. First and foremost, cardiovascular or otherwise cardiological diseases concern a whole set that affects the heart and the blood vessels. According to the World Health Organization (WHO), over 17.1 million people die from cardiovascular diseases in 2019 [1] and the ones that can be studied and analysed through image representation of the eye are arteriosclerosis and hypertension. Arteriosclerosis is a disease in which fats, cholesterol and other substances are built up inside the walls of the arteries (the arteries become thick and stiff), resulting in "narrowing" or even entirely restricting blood flow [2]. Generally, about 2.2 billion people around the world suffer from eye and vision problems.

Hypertension constitutes a chronic condition in which the blood pressure in the arteries is elevated and the so-called Hypertensive Retinopathy (HR) constitutes another retina disease caused by high blood pressure levels. Moving on, another equally important disease is Diabetic etinopathy (DR), which constitutes a disease which affects the retinal vasculature, resulting in loss of vision. Diabetic retinopathy is the most common cause of blindness and vision loss in the western world in patients aged 20 to 65. It is caused by lesions in the vessels of the retina, as it occurs mainly in diabetic patients. Diabetic retinopathy caused by elevated blood sugar levels, is a complication of diabetes in which retinal blood vessels leak into the retina, accompanied by the swelling of the retinal vessels [3]. Diabetic retinopathy can cause the growth of new blood vessels [4]. Another disease that sthe visualisation of the retina can detect is a stroke, which is a condition where the blood supply stops in a part of the brain. As a result, the brain cells do not receive oxygen and die. Scientists have discovered the vessels of the 'eye's retina can help diagnose and treat stroke. In addition, many pathological changes in the retinal vessels constitute direct reflections of the fundus disease. An indicative example is glaucoma and age-related macular degeneration, a condition of macular degeneration that can cause the progressive loss of central vision. Lastly, glaucoma is caused by the high pressure of fluids in the interior of the eyes, causing gradual destruction of the human optic nerve and, as a result, the absence of the peripheral and the end of the 'patient's total vision. Thus, by analysing the length, width and branch structure of retinal vessels, doctors can detect the above diseases early and provide a proper cure for them.

The visualisation of the retina is now done with the help of fundus cameras. Gullstrand developed back in 1910 the notable fundus camera, which is the main concept still used today for image the retina [5]. With these cameras, there is a direct representation of the condition of the retina and therefore documented diagnostic access to the most common or rare diseases of the retina. Fundus cameras create a two-dimensional image from the three-dimensional surface of the eyes using a system that contains a low-energy microscope to which a CCD camera fits. The general procedure is as follows. First, the patient should sit with the chin supported and the forehead positioned properly towards a bar. At the same time, the device operator focuses and positions the camera correctly before pressing the button and activating the photo flash. The resulting photo is mainly an upright, enlarged photo of the retina with standard 30°, 45°, or even 60° imaging angles and magnification up to 2.5 times, depending on the system settings [6]. The resulting image of a fundus camera is illustrated in Figure 1.

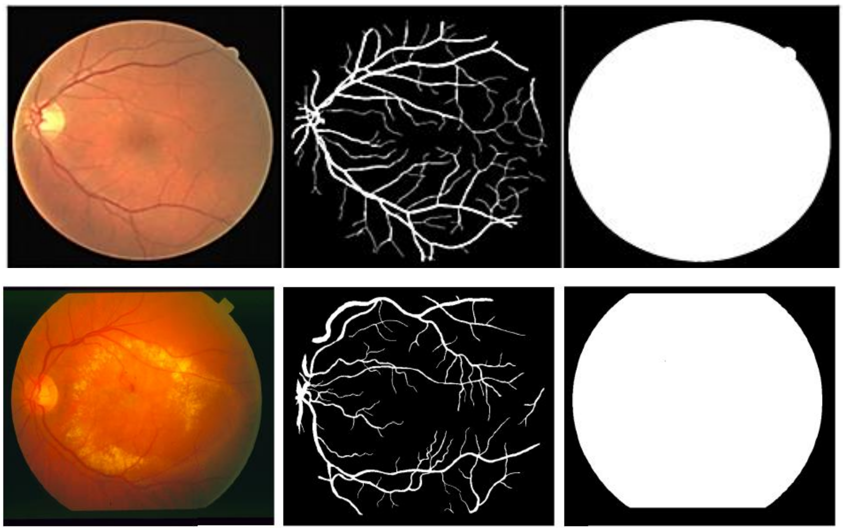

Figure 1. The original image (**right**), the corresponding ground-truth segmentation masks (**middle**), and the corresponding field-of-view masks (**left**). Example cases from the DRIVE (**top**) and from the STARE (**bottom**).

The retinal imaging procedure takes a digital picture of the back of the human eye. A detailed representation of the back of the human eye helps ophthalmologists detect many diseases, such as hypertension, diabetes, stroke and many other cardiovascular diseases. The fundus camera is the most widely used tool for photographing the eye's retina. Retina vessel segmentation is the primary step for the early detection and treatment of various eye diseases. More specifically, the evaluation of fundus images has been done manually and requires a highly skilled ophthalmologist. Through the morphological and topological changes of the retinal vessels, the latter can detect the existence of pathological situations.

Moreover, manual segmentation can be challenging due to the variety of morphological structures eye vessels can have [7]. Automatic segmentation of retinal vessels in fundus images is crucial since manual segmentation can be time and cost-demanding. All things considered, computer-aided detection systems for automatic vessel segmentation are in high demand.

The work of Matsui et al. was one of the first efforts in the literature to present a methodology for retinal image analysis, which is focused mainly on vessel segmentation [8]. Retinal imaging is now the primary way to care for patients with retinal and other systemic diseases [9]. Segmenting the vessels from eye fundus photos constitutes a tedious and demanding procedure in terms of time and carefulness that should be gotten and can require up to three days for all the observations to be gathered accurately. Blood vessel segmentation is a procedure performed manually by a specialist doctor and may be prone to errors. In addition, the daily costs associated with the expert decisions (e.g. ophthalmologist) on eye care and the augmenting number of retinal photos to be examined and analysed constitute the main reasons why an automatic vessel segmentation system should be adopted.

This article proposes a convolutional autoencoder model, a special stream of convolutional neural networks used to segment retina images. The remainder of the article is as follows. In Section 2, we present a complete review of the literature and examine recent related works in the area of eye blood vessel segmentation. After that, Section 3 presents our model, describes all the input data preprocessing steps, and illustrates the proposed architecture of the convolutional autoencoder we designed and developed. The experimental results of our study are presented briefly also in this section. Then, Section 4 explains the experimental study and the assessment of the proposed architecture on different public datasets. Furthermore, it provides a deep and complete comparison of our model with other recent works in the literature. Finally, Section 5 provides our work's main conclusions and draws the main directions for future work.

2. Related Work

Automated vessel segmentation is generally an understandable and well-known problem [10,11]. Basically, concerning the eye, the primary purpose is to separate the pixels of a fundus image into two categories: vessel pixels and none vessel pixels. Several research attempts have been made in the literature for accurate, automatic fundus image segmentation and evaluation. A detailed overview of methods, systems and approaches can be found in the works presented in [12,13].

The deep learning category mainly belongs to methods that solve the problem with classification algorithms. The pixel classification with specific characteristics is a well-known machine learning technique that classifies the pixels of an image into one or more classes. The classification of the pixels is usually performed using a Supervised Learning technique. Vessel segmentation with the help of supervised learning requires two main steps for making the algorithm work properly. In the first step, the algorithm learns statically to classify the pixels correctly from already known classifications. In the second step, which tests how well our algorithm performs, the algorithm classifies images that have never been examined. The first step concerns the training phase, and the second one concerns the testing. Then, for the correct evaluation of the classification algorithm's supervised functionality, the data used for training and the one for the evaluation must be completely different.

In the work presented in [14], the authors present an approach where they face the vessel detection task as a classification problem and develop a CNN (Convolutional Neural Network). Their network consists of two convolution layers, two pooling layers, one dropout layer and a loss layer and is formulated to automatically extract the features without any preprocessing steps. The proposed CNN achieves 91.99% accuracy and 96.52 AUC on the DRIVE dataset and 92.20% accuracy and 94.40 AUC value on the STARE data set, respectively.

Authors in the work presented in [15] present a fully convolutional neural network model used for the blood vessel segmentation task. Moreover, the authors performed five prepossessing steps on the RGB fundus images: extraction of the green channel, normalisation, gamma adjustment, and contrast-limited adaptive histogram equalisation. Finally, the reduction pixels value to the 0–1 range. Then the input given to the 1st convolutional layer is mainly a 1 × 28 × 28 patch extracted from the preprocessed fundus photo. Their model consists of 8 layers. The first two are convolutional layers with 32 filters, the third is a max-pooling layer, and after that, the fourth and the fifth are convolutional layers with 64 filters. The sixth one is an upsampling layer, and the seventh and the eighth are convolutional layers with the same size padding and 32 filters. Finally, the output dimensions are 1 × 28 × 28. The model reports high performance and significantly on the DRIVE dataset reported 95.33% accuracy and 97.4% AUC score.

Mostafiz et al. introduced two efficient methods for vessel segmentation in retinal images [16]. Their study approached the segmentation problem using a Fuzzy classifier and a U-net autoencoder with Residual blocks. The Fuzzy classifier method extracted features by considering a fundus image's mean and median properties, using a fuzzy interface to extract the vessels and post-processing with multi-level threshold and morphological operation. The second technique utilised an autoencoder model to construct masked versions of the retinal images, highlighting only the blood vessels. Both methods achieved state-of-the-art performance, with the Fuzzy system algorithm achieving 95.72% accuracy on the DRIVE test data and the autoencoder network achieving 96.75% accuracy. Their work performed various preprocessing steps on the retinal fundus images, including green channel extraction, complement operation, CLAHE to improve vessel contrast, Gaussian filter to reduce noise, and normalisation by subtracting the background image from the CLAHE applied to the image.

Another work was the construction of an ensemble of deep convolutional neural networks by Maji et al. [17]. More precisely authors developed a computational imaging framework for detecting blood vessels in fundus-coloured images using deep and ensemble learning. They used an ensemble of 12 deep convolutional neural networks to segment vessel and non-vessel areas of the image. Their work explained that ensemble learning involves using multiple models to solve an artificial intelligence problem. Their model consisted of three convolutional layers and two fully connected layers, and they trained it using randomly selected patches from the training images. They evaluated their model on the DRIVE dataset and achieved a maximum average accuracy of 94.7% and an area under the curve of 92.83% for vessel detection.

Moreover, Jin et. al. [18] introduced the Deformable U-Net (DUNet), which uses U-shape architecture to exploit local features of retinal vessels for end-to-end segmentation. They applied three preprocessing steps to the original images: normalisation, CLAHE operation, and gamma correction, and used 48x48 patches to reduce overfitting during training. The DUNet consists of an encoder, a decoder, and a framework, with deformable convolutional blocks, used to model vessels of various shapes and scales. The blocks consist of a convolution offset layer, a convolution layer, a batch normalisation layer, and an activation layer. The model was evaluated on four public datasets (DRIVE, STARE, CHASE_DB1, HRF), achieving a global accuracy of 95.66, 96.41, 96.10, and 96.51 and an AUC of 98.02, 98.32, 98.04, and 98.31 for vessel segmentation.

A noticeable related work in the field concerns the RV-GAN model introduced by Kamran et al. [19]. More specifically, the RV-GAN architecture has a new multi-scale generative architecture, which uses two generators and two multi-scale autoencoding discriminators for better micro-vessel localisation and segmentation. They used two generators since it produces high-quality domain-specific retinal image synthesis. The proposed generators and discriminators consist of both downsampling and upsampling blocks. The downsampling block comprises a convolution layer, a batch-norm layer and a Leaky-ReLU activation function consecutively. In contrast, the upsampling block consists of a transposed convolution layer, batch-norm, and Leaky-ReLU activation layer successively. To avoid the loss of fidelity, Kamran et al. introduced novel weighted loss, which incorporates and prioritises features from the ' ' discriminator's decoder over the encoder. By this, combined with the fact that the ' ' discriminator's decoder attempts to determine actual or fake images at the pixel level, it better preserves macro and microvascular structure. The evaluation metrics of RV-GAN are very promising for DRIVE, STARE and CHASE_DB1 datasets. The model achieves AUC of 98.87, 99.14, and 98.87 and global accuracy of 97.90, 96.97 and 97.54, respectively.

Another GAN architecture proposal was introduced in the work presented in [20], where authors introduced the M-GAN model. This new conditional generative adversarial network uses ACE preprocessing and a generator and discriminator to conduct retinal vessel segmentation. A preprocessing based on ACE is applied to the input fundus image. ACE mimics appropriate adaptive behaviours of the human visual system, such as colour constancy and lightness constancy [21]. The M-generator has deep residual blocks for robust segmentation, and the M-discriminator has a deeper network for efficient adversarial model training. A multi-kernel pooling block is added to support scale invariance, and the M-generator and M-discriminator both have downsampling layers to extract features. The M-generator also has upsampling layers to create segmented retinal blood vessel images, while the M-discriminator has a fully connected layer for decision-making. The performance of the M-GAN model was verified on DRIVE, STARE, CHASE_DB1 and HRF datasets and reported a global accuracy of 97.06, 98.76, 97.36, 97.61 and an AUC of 98.68, 98.73, 98.59, 98.52 on each dataset respectively.

Ultimately, Zhang et al. introduced a pyramid U-Net for the segmentation task of vessels task [22]. The structure of the encoder and decoder part of pyramid U-Net has pyramid-scale Aggregation blocks based on the widely used ResNet blocks. Two optimisations are applied to pyramid-scale aggregation blocks (PSABs) to enhance performance: pyramid inputs enhancement and deep pyramid supervision. In the encoder, scaled input images are added as extra inputs to PSABs, while in the decoder, scaled intermediate outputs are supervised by the scaled segmentation labels. To assess the performance of their approach, authors run experiments on the DRIVE and the CHASE_DB1 datasets. The performance of the pyramid model on the DRIVE dataset got a global accuracy of 96.15% and an AUC of 98.15, while on the CHASE_DBE dataset, the accuracy and the AUC were 96.39% and 98.32, respectively.

3. Methodology

3.1. Datasets

In the context of our work, we train and evaluate our auto-encoder with two publicly available datasets, the DRIVE [23] and the STARE [24]. The DRIVE dataset is the acronym for the Digital Retinal Images for Vessel Extraction and has been used for comparative studies on the segmentation of retinal blood vessels. The images that the DRIVE dataset consists of have been obtained from a diabetic retinopathy program in Holland. In total, 40 images have been selected; more specifically, 33 of those images do not show any sign of diabetic retinopathy, while 7 show some signs of diabetic retinopathy. Specifically, these images were captured using a Canon CR5 non-mydriatic 3CCD camera with a 45-degree FOV (Field of View). The plane resolution of DRIVE is 565 × 584 pixels and a 24-bit grey scale resolution. The dataset's images have been appropriately cropped around the Field of View and a mask image is also provided that delineates the Field of View of each image. The 40 images we used to create two sets, the training and the test set, and each one of those two sets has 20 images. Also, for images of the training set, there is available a single manual segmentation of the vasculature of each image. So, the testing set has 20 images, some masks, and manually labeled vessel structures. Specifically, for the testing set images, two manual segmentations are given; one is used as golden-standard, and the other one aims to assist in comparing the segmentations of the computer method to those of an independent expert. In addition, a mask image is also available for each one of the retinal images and indicates the same region of interest. An experienced ophthalmologist participated in the study to instruct all human observers to segment the vasculature manually. They were requested to mark the pixels for which they were confident for at least 70% that these pixels were vessels.

The STARE (Structured Analysis of the Retina) Project was created at the University of California in 1975. The project was supported by the U.S. National Institutes of Health [24]. Around thirty individuals from various backgrounds contributed to the project, including medicine, science, and engineering. The Shiley Eye Center at the University of California, San Diego, and the Veterans Administration Medical Center in San Diego provided the clinical data and images. The STARE dataset includes 20 colour fundus images with a resolution of 700 × 605 pixels, captured using a TopCon TRV-50 fundus camera. The dataset also contains the manually labeled vessel structure for each image, with two sets of annotations provided by two experts in the field. The first set of annotations is considered to be the ground truth. Half of the images in the STARE dataset depict healthy retinas, while the other half depict retinas with various diseases..

3.2. Image Preprocessing and Data Preparation

In this section, we explain the seven preprocessing steps we applied to our fundus images to improve the performance of our method. The first step concerns the conversion of the image of the eye retina to a greyscale image. This image conversion is suitable since it can produce detailed characteristics of the vessels. Retaining the optical characteristics in medical images to detect the most important features is essential. In the context of eye fundus images, examining blood vessels is crucial in diagnosing eye disorders. While the RGB images of the retina are sufficient for further analysis, converting them to grayscale images has shown more promising outcomes. Previous experiments have shown that single-channel images can produce better contrast between the vessels and background than RGB images [25]. It is essential to be noted that the original-coloured images have the dimensions of: (image_height, image_width, 3) due to the three channels- Red, Green, and Blue. In contrast, after the greyscale conversion, the images have the dimensions of: (image_height, image_width, 1).

After the greyscale conversion, our next step is to normalise our images. In statistics and statistical applications, normalisation can have many meanings. Generally, the normalisation of values refers to rescheduling them to a different scale. Normalising data is a crucial step in machine learning, as it ensures that each input, such as the pixels in each image in this case, has a similar distribution of data. Normalisation makes our model converge faster in the training phase. Data normalisation is performed by subtracting the average from each pixel and dividing the result by the standard deviation. This procedure will result in a centred Gaussian curve distribution around zero. The pixel values of our images must be positive, so we choose to normalise our data in the range [0, 255].

The third step of our proposed preprocessing is using the morphological operation Erosion. The natural effect of this operator is to erode the boundaries of regions with foreground pixels, in this context, the pixels representing the vessels. What we actually do with the Erose function is to enlarge the retinal blood vessels to make them more visible and emphasise the small vessels that are difficult to segment.

Histogram Equalization is a computer image processing technique to enhance image contrast. It is applied as the fourth step in preprocessing. This method typically increases the overall contrast in images when the data has similar contrast values. As a result, areas with low local contrast are given a higher contrast. This step significantly improves the performance of our model since, after this step, the blood vessels in the images are far more visible. So our model can recognise them much easier. So far, in the original fundus images, we have applied greyscale conversion, normalisation, morphological operation, and histogram equalisation [25]. An example case of the preprocessing steps is illustrated in Figure 2.

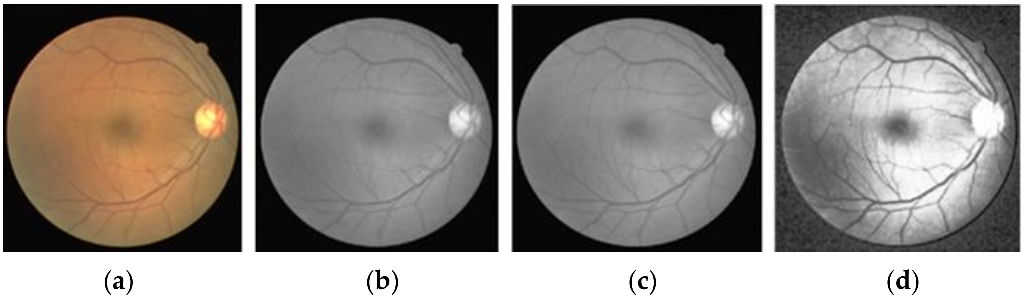

Figure 2. Main preprocessing step performed in the context of our methodology. (**a**) Original image; (**b**) Image after grayscale conversion; (**c**) Image after Erosion Morphological operation; (**d**) Image after Histogram Equalization operation.

Feature scaling is a method used to change the range of the data to another scale. As the range of the data values can vary widely, feature scaling is a necessary step in data preprocessing when using algorithms in machine learning. After the previous four preprocessing steps, the pixel values of the images have values in [0, 255], where a value of 0 represents a black pixel, and 255 represents a white one, respectively. It is essential to state that after this step, the pixels of our images are in the range [0, 1], where 0 represents a black pixel, and 1 represents a white one. The reason why we escalate pixel values to [0, 1] is that deep network learning usually shares many parameters, and if we do not scale our entry in a way that results in values fluctuating in similar scope, sharing them within the network will not be easy, because for example in a part of the image, the weight w will be huge and in another very small.

In the first five steps, we improved the quality of our fundus images to make the retinal blood vessels more discernible, especially the smaller ones, which are extremely difficult to segment. In the following two steps, we enlarge our database due to the pretty small original dataset (for example, the DRIVE dataset consists of only 20 images for our training phase). To do so, we create random patches from our images. We chose our patches to have the size of 48 × 48 and be cropped each time from the processed fundus images randomly. It must be noted that the corresponding patches are made in the manual segmentation of blood vessels in images since we will later use them as labels for the supervised training phase. The size of the patches was selected after experimentation. Due to the smaller size, it is more efficient to work in patches rather than work on the entire photo given to our model. In fact, in the training phase, our proposed model has better results in distinguishing the background of images from FOV (Field of View) since more attention is paid to details and small blood vessels, which are difficult to segment. After experimental evaluation, we found that the number of patches with the highest performance is around 100.000. In Figure 3, example of the patches are illustrated.

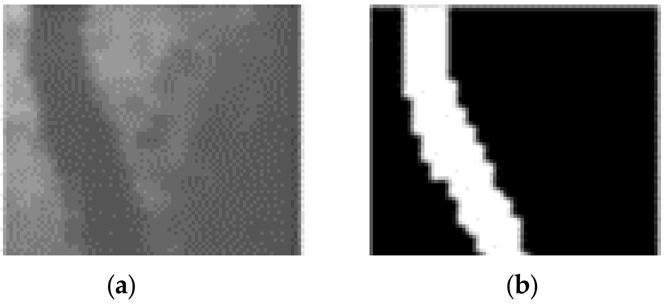

Figure 3. Random cropped patches. (**a**) Cropped patch from original fundus image after the preprocessing steps; (**b**) The corresponding cropped patch of the manual segmentation image.

The last step in our preprocessing phase is the data augmentation technique, which is used to create artificial variations on the existing images to augment the size of our data. To be more specific, data augmentation generates new and unique images from the existing dataset using transformation techniques such as zooming or rotating the existing images. Convolutional Neural Networks (CNNs) require a significant number of images to train the model effectively. Data augmentation helps our model to outperform and reduce the chance of overfitting. In the previous step, we created 100.000 random patches from the eye fundus images, and, in this step, we increased the total number of our dataset to 200.000 patches in total, which significantly improves the metrics that we use to evaluate the performance of our model, such as accuracy under the curve, global accuracy, specificity, precision and others. The size of the patches that we use is 48 × 48.

3.3. Methodology and Autoencoder Formulation

As we mentioned before, we approach vessel segmentation as a classification problem. Indeed, in the context of our work, we built a convolutional neural network, and more specifically, an autoencoder, which classifies the pixels of a given fundus image to be either vessel or non-vessel pixels. Our model was trained using supervised learning, meaning that the manually segmented images helped our network to learn how to detect the vessels more easily (see Figure 4 for an overview of the process). In the following two sections, we explain the theoretical background of this unique type of Neural network and present the layers of our proposed structure.

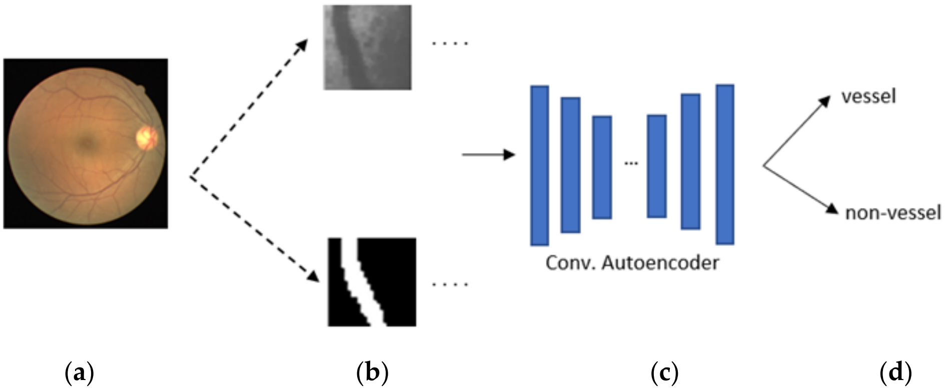

Figure 4. The training process: (**a**) Original image; (**b**) Random cropped patches with the corresponding cropped manual segmentation for the Supervised training; (**c**) Snapshot of the proposed autoencoder; (**d**) Classification of the image pixels.

3.3.1. Background

Autoencoder is a specific deep learning architecture and, more precisely, a specific type of feedforward neural network, where both the input and output data are the same size. With the help of its layers, this network compresses the given input data to a lower-dimension code and then reconstructs the output based on this representation. Autoencoder architecture consists of 3 components: the encoder, the bottleneck and the decoder. As we mentioned above, the encoder is responsible for compressing the input into a coded representation. This representation is called bottleneck and ' 'it's the layer where the input data has lower compression. Finally, in the decoding phase of the autoencoder, the model learns how to reconstruct the compressed data from the bottleneck layer so that the output has the exact dimensions as the input. There are many autoencoders, such as feedforward or LSTM networks. The type of encoder we will build is a fully convolutional autoencoder.

Modelling data that consists of images requires a particular approach in the world of neural networks. Autoencoders constitute a particular stream of neural networks whose input possesses the same dimension as the output. Since our input data is images of the eye retina, it makes sense to use Convolutional Neural Network (convnet) as the encoder and the decoder, respectively. The autoencoders used for images are large convolutional autoencoders due to their significantly better performance. We see a considerable loss of information when we are stacking our data. Instead of stacking the data, convolutional autoencoders preserve the dimensions of the input images and extract information gently and with the help of a layer called Convolutional. In convolutional autoencoders, the encoding part consists of hidden layers. The decoder has the same layers as the encoder but is mirrored. So, the encoder and the decoder are symmetrical with each other. This is not mandatory but is usually how we build our networks. We need to configure four parameters before continuing with the training phase. These are the number of nodes at the bottleneck layer (the smaller the number, the bigger the compression is), the number of hidden layers (it depends on how "deep" we want our network to be), the number of nodes in every dense layer as well as the loss function too. Below, we explain the types of layers that we used in our proposed structure.

3.3.2. Hidden Layers

As mentioned, CNNs are a particular network type used on two-dimensional image databases. The critical feature of convolutional neural networks and hence of convolutional autoencoders is the convolutional layer that gives the network its name. Convolution is the simple filter applied to an input that results in activation. Convolution is a linear operation that involves multiplying a set of weights with the input data using a two-dimensional array of weights called a filter or kernel. The filter is smaller than the input data, and the multiplication is performed using a dot product between the filter-sized patch of input and the filter. This systematic application of the same filter across an image allows the filter to detect a specific type of feature in the input, allowing it to discover that feature anywhere in the image. When a filter is multiplied with the input array, it produces a single value. A two-dimensional array of output values is obtained by repeatedly applying the same filter to different parts of the input, known as a feature map. The feature map represents a filtered version of the input [26]. The feature map implicitly depends on the learning model class used and on the input space where the data lies. Feature maps are produced using feature detectors or filters on either the input image or the feature map generated by the previous layers. These feature maps can provide useful information about the internal representations of the input for each Convolutional layer in the model. Visualising these feature maps can help gain insight into these representations. Convolutional layers also have a parameter which is called stride. The stride is the number of pixels the filter moves over the input array. When the step equals one, the filters are shifted by 1 pixel at a time.

When we build a neural network, we need an activation function that takes the linear neuron output as input and generates a non-linear output based on it. The activation function can be a step transfer function, a linear transfer function, a non-linear transfer function or a stochastic transfer function. ReLU is one of the most widely used activation functions in neural networks today. It is usually added to some layers in neural networks to add nonlinearity, which is required to handle ' 'today's complex and non-linear datasets. ReLU is more well-known than older activate functions, such as Sigmoid or Tanh, because it can be computed without a considerable cost, although it faces various problems when we use it. Its output is ReLU (x) = max (0, x). First, ReLU is not continuously differentiable. The gradient cannot be computed at x = 0, the breaking point between x and 0. Being unable to compute the gradient is not a big problem, but it can very lightly impact training performance. Second and graver, ReLU set all values < 0 to zero. This is beneficial regarding sparsity, as the network will adapt to ensure that the most critical neurons have values of >0. However, this is also a problem since the gradient of 0 is 0. Hence neurons arriving at large negative values cannot recover from being stuck at 0 [27]. What if we cause a small but significant leak of information to the left part of ReLU, i.e., where the output is always stuck to 0? The answer is the Leaky ReLU (rectified linear activation function), widely used in many machine learning applications. Specifically, it is an improvement of the traditional ReLU, and we recommend it be used more often. So, the activation function that we use is Leaky ReLU and is mathematically defined as f(x) = {0.01x if x < 0 or x otherwise}.

Deep learning neural networks will likely quickly overfit a training dataset with few examples. This phenomenon happens when the model fits very well in the training dataset. Therefore, it becomes difficult for the model to adapt to new examples that do not belong to the training dataset. To make it more understandable, our model can recognise specific images from the training dataset, not general patterns. Overfitting affects our model resulting in deficient performance when the model is evaluated on new data. Dropout layers can help us to prevent overfitting. The term "Dropout" refers to leaving out some nodes in the neural network. Using Dropout in a neural network makes the training process more turbulent, which compels nodes in a layer to randomly accept or reject responsibility for the input data [28]. In other words, the Dropout layer refers to ignoring a set of nodes during the training phase which are randomly selected. Therefore, the Dropout layer forces a neural network to learn more about the key features and on top of that, the training time of each epoch is shorter.

When we have features with values in the range 0–1 and some others in the range 1–100, we suggest normalising these values, so the training process of our model becomes faster. If this technique benefits the input layer, why do we not do the same for the values inside the dense layers of our convolutional autoencoder that constantly change? Batch normalisation layers reduce the overfitting effect and, similar to the Dropout layer, add a little noise to the activations of each hidden layer. Therefore, if we use batch normalisation layers, we will use fewer Dropout layers, which is good because we lose much information. However, we should not rely solely on the Batch normalisation layers as using a combination of Dropout layers is more efficient.

The convolutional autoencoder consists of the encoder and the decoder. In the decoding part, the model learns how to reconstruct data from the compressed encoder representation by having the same layers the encoder has but mirrored. As we explained before, the MaxPooling layer helps us compress the input image (Downsampling), so now it makes sense that we must restore the compressed image to its original dimensions. Here is where the Upsampling layer takes over action. The upsampling layer is a simple version of Unpooling (the opposite of the pooling layer), where it repeats the input's rows and columns.

Finally, the need for transposed convolutions generally arises from the desire to use a transformation going in the opposite direction of a standard convolution, i.e., from something that has the shape of the output of some convolution to something that has the shape of its input while maintaining a connectivity pattern that is compatible with said convolution. Hence, choosing a convolutional autoencoder would be a good idea in the decoding part of our model to use the Conv2Transpose layer. There are convolutional transpose layers for two and three dimensions; we chose the ones for two because our images have two dimensions. The Conv2DTranspose layers learn many filters, similar to the superficial Convolutional layer. We used multiple times the Conv2DTranspose layer in the decoder of our proposed model, reflecting the Convolutional layers of the encoder. We could use the superficial Convolutional layer in the Decode part as well, but the performance of our model was significantly lower.

3.3.3. Autoencoder

The proposed model consists of eight layers. Each comprises convolutional2D layers, MaxPooling layers, Batch Normalisation layers and more. Our autoencoder includes the encoder, the decoder and the bottleneck. The encoder consists of the first 4 big layers and the decoder of the rest 4. The immense layers in the network are the input and output layers located at the beginning and end of the network, respectively. The input of our model are the patches that we cropped in the preprocessing steps, so the input has (48,48,1) dimensions. As for the output, it has the exact dimensions as the input (definition of autoencoder). Also the second layer consists of 3 levels. The first level has a Convolution layer with a number of filters equals to 8 and a LeakyReLU layer, the second level has a convolutional layer which has 32 filters, a LeakyReLU and also a Batch Normalization layer, and the third level has a convolutional layer with 32 filters and parameter strides = (2,2) which act like a MaxPooling layer, a LeakyReLU and a Batch Normalisation layer. The compression our patches have so far is from their original to 24 × 24. Then the second layer consists of 3 levels. The first level has a convolutional layer with a number of filters equal to 256 and a LeakyReLU layer, the second level has the previous two layers again, but with the addition of a Dropout layer, and finally, the third level is a MaxPooling layer with size (2,2), which means it compresses our patches to 12 × 12.

The third layer has two levels: a convolutional layer with 512 filters, a LeakyReLU layer, and a Dropout layer. Then, in the second level of our third layer, we use a MaxPooling layer for further compression, and now our patches have the most significant compression sized as 6 × 6. Then, as we mentioned before, the decoding part of an autoencoder reconstructs the data and, more importantly, has the same layers as the encoder but mirrored. For example, the fourth layer has the same layers as the third, but we replace the MaxPooling layers with the Upsampling layers for the reconstruction. Also, it is important to mention that we replace all the Convolutional layers except the last one with the Conv2DTranspose layer in the decoding part. The architecture of our proposed method is presented in following in Figure 5.

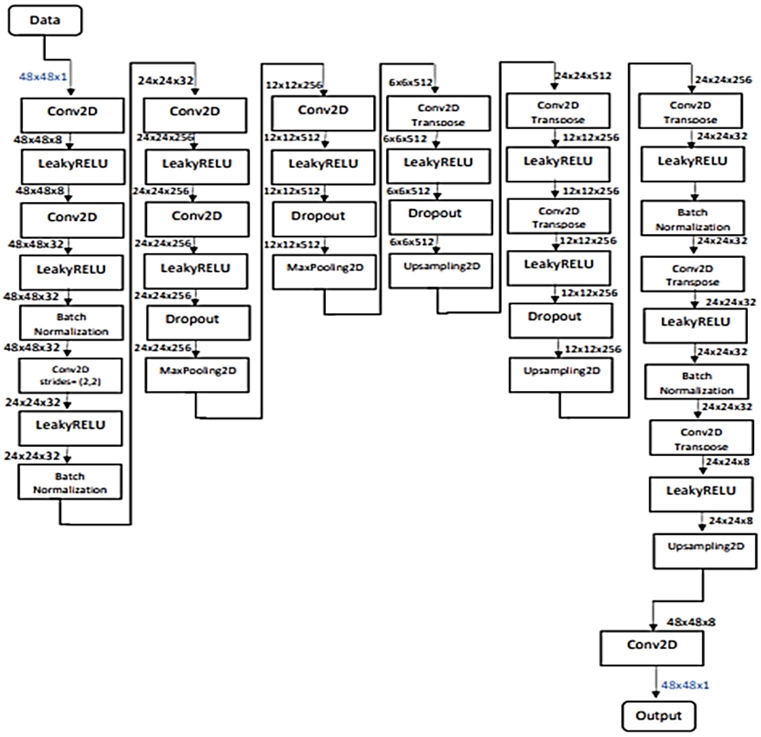

Figure 5. The architecture of the proposed autoencoder. Each column of hidden layers represents a bigger layer.

Our database's final size, the patches' dimensions, the number of epochs our model will be trained, and the batch size will be chosen after experimentation. We chose to crop 200.000 patches randomly from the original fundus images since we did not see any further improvement of our model in the training phase. Then, the most efficient combination of the parameters above is: patch size = (48, 48, 1) (the third dimension is one because our patches are greyscaled), number of epochs = 4 and batch size = 8.

4. Experimental Study

4.1. Performance Evaluation

For the evaluation procedure of our convolutional autoencoder, we used several metrics, which are the following: global accuracy, AUC_ROC, which is the Area-Under-Curve (AUC) of Receiver-Operating-Characteristic (ROC), specificity, f1-score, sensitivity and precision. We also represented the ROC curve for each dataset. These evaluation metrics were calculated based on the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) rates. Global accuracy is a metric used to measure the ratio of correctly classified pixels to the total number of pixels in the dataset. Specificity measures the proportion of the negatives correctly identified, while sensitivity, also known as recall, measures the proportion of the positives correctly identified. Finally, F1-score concerns the recall's harmonic mean (average) and precision [29]. Those metrics have the following mathematical definitions:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

$$\mathrm{Specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}$$

$$\mathrm{Sensitivity} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$\mathrm{F1Score} = \frac{2 \times (\mathrm{Recall} \times \mathrm{Precision})}{\mathrm{Recall} + \mathrm{Precision}}$$

The final results of our proposed model are promising. We tested our autoencoder on both the DRIVE and the STARE databases. First, the DRIVE dataset's metrics are AUC_ROC = 97.49, accuracy = 95.73, specificity = 98.57, f1-score = 81.0, precision = 88.0 and sensitivity = 75.0. On the STARE dataset, the metrics are AUC_ROC = 97.57, accuracy = 96.92, specificity = 98.7, f1-score = 79.0, precision = 82.0 and sensitivity = 75.45. Finally, the ROC curves of our introduced model on each of the databases mentioned above are presented in Figure 6. Also, an example segmentation case of our methodology is illustrated in Figure 7.

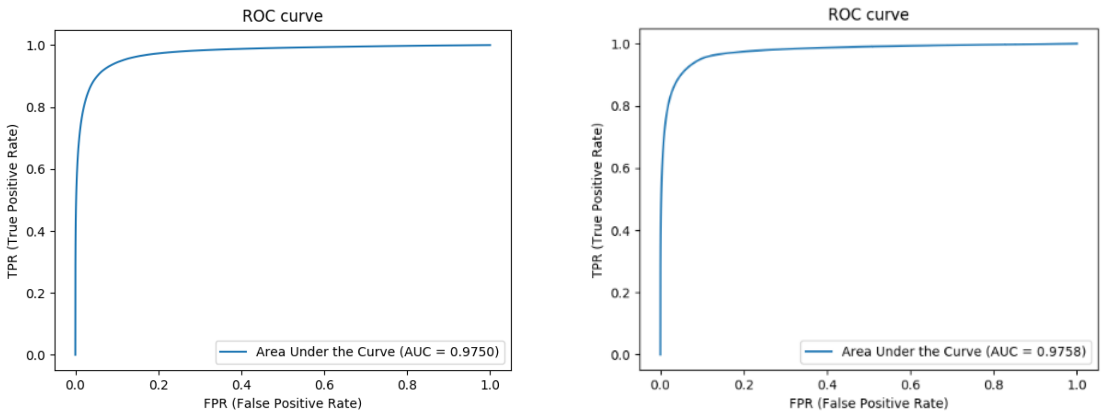

Figure 6. ROC curves of our model. On the left is the ROC curve for the DRIVE dataset and the ROC curve for the STARE on the right.

4.2. Comparison against Existing Methods

A deep and concrete comparative study has been performed to provide insight into the performance of our introduced method and compare its performance towards recent related works in the field. Comparing our autoencoder with other models (see Tables 1 and 2), we notice that our proposed autoencoder is trained in fewer epochs and performs quite well. Also, it has the highest specificity. It should be noted that the training process of other models lasts many hours since they are trained with 100–150 epochs. Specifically, the SA-UNet proposed by Guo et al. [30] is the newest model with the best overall performance trained in 150 epochs. On the other hand, our proposed convolutional autoencoder is trained significantly faster (within 35 min), and its results are very competitive, with the specificity surpassing any other model's. Specificity essentially measures the percentage of correct black pixels in evaluation images. This can be seen from the final result, after which the images are pretty "clean" without the noise we observe in other models. For example, an attempt at automatic segmentation of the blood vessels by the model of Fan et al. is shown in Figure 8. On the left, we can observe that the octave convolutional neural network adds enough noise in the final attempt to segment the retina image.

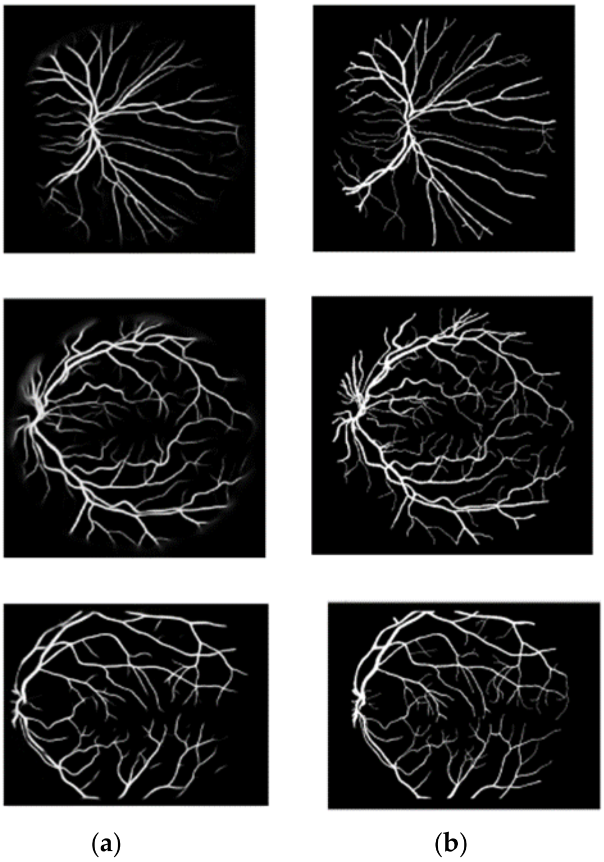

Figure 7. Segmentation results using the proposed model. (**a**) Results of automated segmentations by our model; (**b**) Corresponding manual segmentations.

On the contrary, our convolutional autoencoder has a more precise result without noise. These photos belong to the STARE database. It is important to emphasise that in the DRIVE database, each image's masks are available, allowing us to calculate the performance metrics only in the FOV (field of view) and not in the pixels of the background. In the STARE database, the masks are unavailable; therefore, the metric evaluation results include all the images' pixels.

Table 1. Performance comparison of the introduced convolutional autoencoder on the DRIVE dataset.

Model	**AUC_ROC**	**ACC**	**SPEC**	**F1 SCORE**	**PREC**	**SENS**
Guo et al. [14]	96.52	90.64	92.83	90.66	92.47	89.90
Jin et al. [18]	98.02	95.66				
Park et al. [21]	**98.68**	**97.06**	98.36	83.24		83.46
Zhang et al. [22]	98.15	96.15	98.07			82.13
Hou et al. [31]	96.10	94.10	96.90			73.50
Cheng et al. [32]	96.48	94.74	97.98			72.52
Zhao et al. [33]	84.80	94.40	97.80			71.60
Fu et al. [34]		95.23				76.03

Table 1. *Cont.*

Model	**AUC_ROC**	**ACC**	**SPEC**	**F1 SCORE**	**PREC**	**SENS**
Azad et al. [35]	97.88	95.59	97.84	82.22		80.12
Guo et al. [30]	98.64	96.98	98.40	82.63		82.12
Roychowdhury et al. [36]	96.20	95.20	98.30			72.50
OUR MODEL	97.49	95.73	**98.57**	81.27	88.00	78.40

Table 2. Performance comparison of the introduced convolutional autoencoder on the STARE dataset.

Model	**AUC_ROC**	**ACC**	**SPEC**	**F1 SCORE**	**PREC**	**SENS**
Guo et al. [30]	98.75	**97.13**	97.98	81.91		**86.64**
Alom et al. [37]	**99.14**	97.12	98.62	**84.75**		82.29
Mou et al. [38]	98.58	96.85	97.61			83.91
Lei et al. [39]	98.12	96.48	97.68			82.75
Tian et al. [40]		94.92	97.71			70.19
Yang et al. [41]		95.16	97.31			67.13
Shukla et al. [42]		95.73	98.63			70.23
Orujov et al. [43]		86.50	88.06			83.42
Mahapatra et al. [44]		96.01	98.02			68.46
OUR MODEL	97.57	96.92	**98.70**	79.00	82.00	75.45

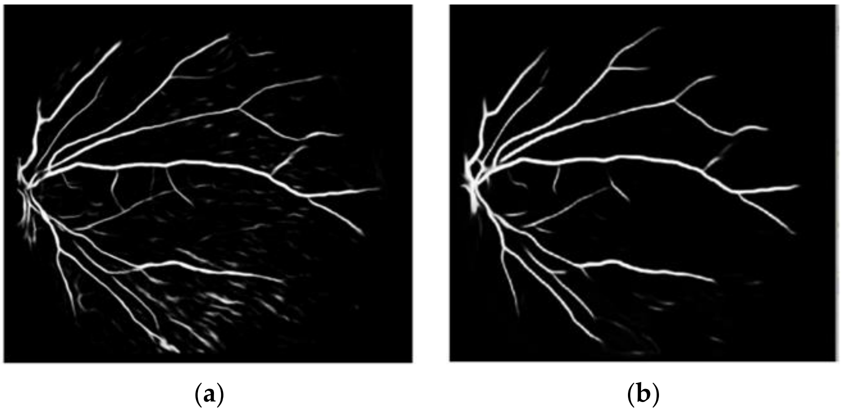

Figure 8. Comparison against existing method. (**a**) An attempt of automatic segmentation by the Octave Convolution Neural Network [45]. (**b**) The corresponding automated vessel segmentation of our proposed convolutional autoencoder.

Therefore, our initial idea was to build an automatic encoder by specifying a structure suitable for producing reconstructed images without noise, and it was very efficient. So, we built a convolutional autoencoder, a unique stream of a neural network, for the blood vessel segmentation task. Through the evaluation of our model, we saw that it is competitive with other proposals using past models and has the best specificity value. Another advantage of our autoencoder is the short time the training process takes. More specifically, our model takes up to 35 min to learn to automate segment fundus images. Practically, our model differs in the final results since the images produced are much cleaner without noise and without the creation of vessels that do not exist. In Tables 1 and 2, performance comparisons of our model towards works in the literature are presented. Finally, it is worth indicating that we designed, formulated and evaluated our model in a machine with the characteristics in Table 3.

Table 3. Hardware Characteristics.

Operating System	Ubuntu 16.04.7 LTS
CPU	Intel (R) Core (TM) i7-5960X 3.00 GHz
RAM	62 GB DDR4
GPUs	GPU0: G-FORCE GTX 1080 8 GB GPU1: G-FORCE GTX 1080 8 GB

The experimental results reveal pretty impressive findings. First, the results highlight the quite good performance of our model. Our autoencoder achieved quite good accuracy. Our model's quite good performance is the architecture we designed and the balanced number of layers it consists of. In addition, the results also point out that our model reports the best performance among the models in the literature in terms of specificity. Indeed, in this regard and to the best of our knowledge, the best performance is achieved by our model on both the STARE and DRIVE datasets. Finally, our model achieves a quite good F1 score while the sensitivity is on a good level compared to the related works in the literature. Last but not least, it is worth indicating that we designed, formulated and evaluated our model in a machine with the characteristics in Table 3.

5. Discussion and Conclusions

Through this research, we understood the vital role of bioinformatic applications in modern times. Fast, automatic and accurate vessel segmentation for diagnosis can even save lives. We approached the challenge of segmenting retinal blood vessels by treating it as a classification task. Since our work involves image processing, we chose the model of automatic encoders. For the construction of our auto-encoder, we chose convolutional layers. With its gradual construction, we understood what is more efficient for the model due to the variety of morphological structures eye vessels can have. The final convolutional auto-encoder, therefore, is trained on two datasets in a concise amount of time (35 min), having a competitive performance compared to other models that have been proposed in the past. The specificity metric has the highest value compared to all other models in both databases. This metric calculates the percentage of true negatives, and to be more understandable, it expresses how many pixels were correctly predicted as black, i.e., as non-vessels. The high specificity value can also be perceived practically through the images produced in the testing process. More specifically, as we have discussed in the comparison section, the corresponding images produced by our model are accurate, sharp and "cleaner" in the lines of the vessels and without excess noise.

Our research and the model we introduced could be applied in real situations since the proposed convolutional auto-encoder is efficient enough compared to other models. In particular, it would be possible to construct a system that would have as input the automatically segmented images of the retinal blood vessels and as an output the information regarding the patient and if they are suffering from a disease or not.

There are some directions that future work could examine. First, a more extensive scale evaluation could be designed, and additional datasets such as the High-resolution fundus and the CHASE-DB1 image databases to get an even better insight into the performance of our proposed method. Moreover, a deeper investigation-study of the layers could be the key to increasing the performance. Adding noise, such as Gaussian, could be examined to improve the model since it could help it better distinguish the vessels from the background. Also, another future work direction is the examination of techniques for creating the feature maps such as spatial pyramid networks. The examination of this direction concerns an essential aspect of our future work. Finally, the formulation of a web application with an interface to facilitate ophthalmologists in using our method in real-time easily constitutes another direction for future work..

Author Contributions: Conceptualization, I.P., P.T. and N.N.; methodology, I.P. and N.N.; software, N.N., I.P. and I.D.; validation, N.N. and I.P.; writing—original draft preparation, N.N., I.P. and I.H.; visualization, N.N. and I.D.; Writing—review & editing, NN, I.P. and P.T. supervision, I.H., P.T. and C.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported by the Institute of Data Science and Digital Technologies, Image and Signal Analysis Group, Faculty of Mathematics and Informatics, Vilnius University of Lithuania and the Computer Engineering and Informatics Department, University of Patras, Greece.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Han, Z.; Yin, Y.; Meng, X.; Yang, G.; Yan, X. Blood vessel segmentation in pathological retinal image. In Proceedings of the 2014 IEEE International Conference on Data Mining Workshop, IEEE, Shenzhen, China, 14–17 December 2014; pp. 960–967.
2. Cardiovascular Diseases (CVDs). World Health Organization. Available online: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds) (accessed on 27 February 2022).
3. Wikipedia, Arteriosclerosis. Available online: https://en.wikipedia.org/wiki/Arteriosclerosis (accessed on 2 February 2022).
4. Smart, T.J.; Richards, C.J.; Bhatnagar, R.; Pavesio, C.; Agrawal, R.; Jones, P.H. A study of red blood cell deformability in diabetic retinopathy using optical tweezers. In Proceedings of the Optical Trapping and Optical Micromanipulation XII, SPIE, San Diego, CA, USA, 9–12 August 2015; Volume 9548, pp. 342–348.
5. Laibacher, T.; Weyde, T.; Jalali, S. M2U-Net: Effective and efficient retinal vessel segmentation for resource-constrained environments. *arXiv* **2018**, arXiv:1811.07738.
6. Gullstrand, A. Neue methoden der reflexlosen ophthalmoskopie. *Ber. Dtsch. Ophthalmol. Ges.* **1910**, *36*, 326.
7. MacGillivray, T.J.; Trucco, E.; Cameron, J.R.; Dhillon, B.; Houston, J.G.; Van Beek, E.J.R. Retinal imaging as a source of biomarkers for diagnosis, characterization and prognosis of chronic illness or long-term conditions. *Br. J. Radiol.* **2014**, *87*, 20130832. [CrossRef] [PubMed]
8. Matsui, M.; Tashiro, T.; Matsumoto, K.; Yamamoto, S. A study on automatic and quantitative diagnosis of fundus photographs. I. Detection of contour line of retinal blood vessel images on color fundus photographs (author's transl). *Nippon. Ganka Gakkai Zasshi* **1973**, *77*, 907–918. [PubMed]
9. Abràmoff, M.D.; Garvin, M.K.; Sonka, M. Retinal imaging and image analysis. *IEEE Rev. Biomed. Eng.* **2010**, *3*, 169. [CrossRef]
10. Khandouzi, A.; Ariafar, A.; Mashayekhpour, Z.; Pazira, M.; Baleghi, Y. Retinal vessel segmentation, a review of classic and deep methods. *Ann. Biomed. Eng.* **2022**, *50*, 1292–1314. [CrossRef]
11. Ciecholewski, M.; Kassjański, M. Computational methods for liver vessel segmentation in medical imaging: A review. *Sensors* **2021**, *21*, 2027. [CrossRef]
12. Moccia, S.; De Momi, E.; El Hadji, S.; Mattos, L.S. Blood vessel segmentation algorithms—Review of methods, datasets and evaluation metrics. *Comput. Methods Programs Biomed.* **2018**, *158*, 71. [CrossRef]
13. Mookiah, M.R.; Hogg, S.; MacGillivray, T.J.; Prathiba, V.; Pradeepa, R.; Mohan, V.; Anjana, R.M.; Doney, A.S.; Palmer, C.N.; Trucco, E. A review of machine learning methods for retinal blood vessel segmentation and artery/vein classification. *Med. Image Anal.* **2021**, *68*, 101905. [CrossRef]
14. Guo, Y.; Budak, Ü.; Vespa, L.J.; Khorasani, E.; Şengür, A. A retinal vessel detection approach using convolution neural network with. *Measurement* **2018**, *125*, 586–591. [CrossRef]
15. Dasgupta, A.; Singh, S. A fully convolutional neural network based structured prediction approach towards the retinal vessel segmentation. In Proceedings of the 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017), IEEE, Melbourne, Australia, 18–21 April 2017; pp. 248–251.
16. Mostafiz, T.; Jarin, I.; Fattah, S.A.; Shahnaz, C. Retinal blood vessel segmentation using residual block incorporated U-Net architecture and fuzzy inference system. In Proceedings of the 2018 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), IEEE, Chonburi, Thailand, 14–16 December 2018; pp. 106–109.
17. Maji, D.; Santara, A.; Mitra, P.; Sheet, D. Ensemble of deep convolutional neural networks for learning to detect retinal vessels in fundus images. *arXiv* **2016**, arXiv:1603.04833.
18. Jin, Q.; Meng, Z.; Pham, T.D.; Chen, Q.; Wei, L.; Su, R. DUNet: A deformable network for retinal vessel segmentation. *Knowl.-Based Syst.* **2019**, *178*, 149. [CrossRef]
19. Kamran, S.A.; Hossain, K.F.; Tavakkoli, A.; Zuckerbrod, S.L.; Sanders, K.M.; Baker, S.A. RV-GAN: Segmenting retinal vascular structure in fundus photographs using a novel multi-scale generative adversarial network. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2021; pp. 34–44.
20. Park, K.B.; Choi, S.H.; Lee, J.Y. M-gan: Retinal blood vessel segmentation by balancing losses through stacked deep fully convolutional networks. *IEEE Access* **2020**, *8*, 146308. [CrossRef]
21. Rizzi, A.; Gatta, C.; Marini, D. A new algorithm for unsupervised global and local color correction. *Pattern Recognit. Lett.* **2003**, *24*, 1663–1677. [CrossRef]

22. Zhang, J.; Zhang, Y.; Xu, X. Pyramid u-net for retinal vessel segmentation. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Toronto, ON, Canada, 6–11 June 2021; pp. 1125–1129.
23. RIVE, Digital Retinal Images for Vessel. 2004. Available online: https://drive.grand-challenge.org/ (accessed on 20 January 2023).
24. Goldbaum, M. STructured Analysis of the Retina, STARE Dataset. Available online: https://cecas.clemson.edu/~ahoover/stare/ (accessed on 20 January 2023).
25. Pizer, S.M.; Amburn, E.P.; Austin, J.D.; Cromartie, R.; Geselowitz, A.; Greer, T.; ter Haar Romeny, B.; Zimmerman, J.B.; Zuiderveld, K. Adaptive histogram equalization and its variations. *Comput. Vis. Graph. Image Process.* **1987**, *39*, 355–368. [CrossRef]
26. Brownlee, J. How Do Convolutional Layers Work in Deep Learning Neural Networks? *Machine Learning Mastery, 17 April 2019.* Available online: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/ (accessed on 20 January 2023).
27. Chris, Leaky ReLU: Improving Traditional ReLU, Machine Curne, 15 October 2019. Available online: https://www.machinecurve.com/index.php/2019/10/15/leaky-relu-improving-traditional-relu/ (accessed on 20 January 2023).
28. Brownlee, J. A Gentle Introduction to Dropout for Regularizing Deep Neural Networks, Machine Learning Mastery, 3 December 2018. Available online: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/ (accessed on 20 January 2023).
29. Ghoneim, S. Accuracy, Recall, Precision, F-Score & Specificity, Which to Optimize on? Towards Data Science, 2 April 2019. Available online: https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124 (accessed on 14 March 2022).
30. Guo, C.; Szemenyei, M.; Yi, Y.; Wang, W.; Chen, B.; Fan, C. Sa-unet: Spatial attention u-net for retinal vessel segmentation. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, Milan, Italy, 10–15 January 2021; pp. 1236–1242.
31. Hou, Y. Automatic segmentation of retinal blood vessels based on improved multiscale line detection. *J. Comput. Sci. Eng.* **2014**, *8*, 119–128. [CrossRef]
32. Cheng, E.; Du, L.; Wu, Y.; Zhu, Y.J.; Megalooikonomou, V.; Ling, H. Discriminative vessel segmentation in retinal images by fusing context-aware hybrid features. *Mach. Vis. Appl.* **2014**, *25*, 1779–1792. [CrossRef]
33. Zhao, Y.; Rada, L.; Chen, K.; Harding, S.P.; Zheng, Y. Automated vessel segmentation using infinite perimeter active contour model with hybrid region information with application to retinal images. *IEEE Trans. Med. Imaging* **2015**, *34*, 1797–1807. [CrossRef]
34. Fu, H.; Xu, Y.; Lin, S.; Kee Wong, D.W.; Liu, J. Deep vessel: Retinal vessel segmentation via deep learning and conditional random field. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention; Springer: Cham, Switzerland, 2016; pp. 132–139.
35. Azad, R.; Asadi-Aghbolaghi, M.; Fathy, M.; Escalera, S. Bi-directional ConvLSTM U-Net with densley connected convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27 October–2 November 2019.
36. Roychowdhury, S.; Koozekanani, D.D.; Parhi, K.K. Iterative vessel segmentation of fundus images. *IEEE Trans. Biomed. Eng.* **2015**, *62*, 1738–1749. [CrossRef]
37. Alom, M.Z.; Yakopcic, C.; Hasan, M.; Taha, T.M.; Asari, V.K. Recurrent residual U-Net for medical image segmentation. *J. Med. Imaging* **2019**, *6*, 014006. [CrossRef]
38. Mou, L.; Chen, L.; Cheng, J.; Gu, Z.; Zhao, Y.; Liu, J. Dense dilated network with probability regularized walk for vessel detection. *IEEE Trans. Med. Imaging* **2019**, *39*, 1392–1403. [CrossRef] [PubMed]
39. Wu, H.; Wang, W.; Zhong, J.; Lei, B.; Wen, Z.; Qin, J. Scs-net: A scale and context sensitive network for retinal vessel segmentation. *Med. Image Anal.* **2021**, *70*, 102025. [CrossRef] [PubMed]
40. Tian, F.; Li, Y.; Wang, J.; Chen, W. Blood vessel segmentation of fundus retinal images based on improved frangi and mathematical morphology. *Comput. Math. Methods Med.* **2021**, *2021*, 4761517. [CrossRef] [PubMed]
41. Yang, J.; Huang, M.; Fu, J.; Lou, C.; Feng, C. Frangi based multi-scale level sets for retinal vascular segmentation. *Comput. Methods Programs Biomed.* **2020**, *197*, 105752. [CrossRef] [PubMed]
42. Shukla, A.K.; Pandey, R.K.; Pachori, R.B. A fractional filter based efficient algorithm for retinal blood vessel segmentation. *Biomed. Signal Process. Control.* **2020**, *59*, 101883. [CrossRef]
43. Orujov, F.; Maskeliūnas, R.; Damaševičius, R.; Wei, W. Fuzzy based image edge detection algorithm for blood vessel detection in retinal images. *Appl. Soft Comput.* **2020**, *94*, 106452. [CrossRef]
44. Mahapatra, S.; Agrawal, S.; Mishro, P.K.; Pachori, R.B. A novel framework for retinal vessel segmentation using optimal improved frangi filter and adaptive weighted spatial FCM. *Comput. Biol. Med.* **2022**, *147*, 105770. [CrossRef]
45. Fan, Z.; Mo, J.; Qiu, B.; Li, W.; Zhu, G.; Li, C.; Hu, J.; Rong, Y.; Chen, X. Accurate retinal vessel segmentation via octave convolution neural network. *arXiv* **2020**, arXiv:1906.12193.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.