IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 June 2022, accepted 23 August 2022, date of publication 29 August 2022, date of current version 8 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3202890

RESEARCH ARTICLE

VG-DropDNet a Robust Architecture for Blood Vessels Segmentation on Retinal Image

ANITA DESIANI[1], ERWIN[2], (Member, IEEE), BAMBANG SUPRIHATIN[1], FILDA EFRILIYANTI[3], MUHAMMAD ARHAMI[4], AND EMY SETYANINGSIH[5]

[1]Mathematics Department, Mathematics and Natural Science Faculty, Universitas Sriwijaya, Indralaya 30662, Indonesia
[2]Computer Engineering Department, Computer Science Faculty, Universitas Sriwijaya, Indralaya 30662, Indonesia
[3]Computation Laboratory, Mathematics and Natural Science Faculty, Universitas Sriwijaya, Indralaya 30662, Indonesia
[4]Informatics Engineering Department, Politeknik Negeri Lhokseumawe, Lhokseumawe 24375, Indonesia
[5]Computer Systems Department, Institut Sains dan Teknologi Akprind, Yogyakarta 55222, Indonesia

Corresponding author: Erwin (erwin@unsri.ac.id)

This work was supported by the Computation Laboratory, Mathematic and Natural Science Faculty, Universitas Sriwijaya.

ABSTRACT Additional layers to the U-Net architecture leads to additional parameters and network complexity. The Visual Geometry Group (VGG) architecture with 16 backbones can overcome the problem with small convolutions. Dense Connected (DenseNet) can be used to avoid excessive feature learning in VGG by directly connecting each layer using input from the previous feature map. Adding a Dropout layer can protect DenseNet from Overfitting problems. This study proposes a VG-DropDNet architecture that combines VGG, DenseNet, and U-Net with a dropout layer in blood vessels retinal segmentation. VG-DropDNet is applied to Digital Retina Image for Vessel Extraction (DRIVE) and Retina Structured Analysis (STARE) datasets. The results on DRIVE give great accuracy of 95.36%, sensitivity of 79.74% and specificity of 97.61%. The F1-score on DRIVE of 0.8144 indicates that VG-DropDNet has great precision and recall. The IoU result is 68.70. It concludes that the resulting image of VG-DropDNet has a great resemblance to its ground truth. The results on STARE are excellent for accuracy of 98.56%, sensitivity of 91.24%, specificity of 92.99% and IoU of 86.90%. The results of the VGG-DropDNet on STARE show that the proposed method is excellent and robust for blood vessels retinal segmentation. The Cohen's Kappa coefficient obtained by VG-DropDNet at DRIVe is 0.8386 and at STARE is 0.98, it explains that the VG-DropDNet results are consistent and precise in both datasets. The results on various datasets indicate that VG-DropDnet is effective, robust and stable in retinal image blood vessel segmentation.

INDEX TERMS Blood vessels, DenseNet, retinal image, segmentation, U-Net, VG-DropDNet.

I. INTRODUCTION

Segmentation is a technique used in image processing that divides an image into several distinct areas, including the object being studied and the background. Segmentation is widely used in a variety of fields, including road detection using satellite images and disease diagnosis using medical images [1]. Segmentation of medical images is a critical processing step because it enables disease diagnosis and treatment planning [2]. Manual medical image segmentation performed by humans requires expert knowledge and high accuracy, while human capabilities are limited and humans have laborious factors that can cause the segmentation process to take longer [3], [4], [5]. The retina is a standard medical image that is segmented. Through the retinal blood vessels, retinal images can be used to diagnose, treat, and monitor for a variety of eye diseases, including glaucoma, myopia, and diabetic retinopathy (DR) [6].

Convolutional Neural Networks (CNN) have made significant progress in medical image analysis in recent years [7], [8]. The U-Net architecture is the most frequently [8], [9] used CNN architecture for segmentation in the medical field [8], [9]. U-Net consists of encode and decode that connected by a bridge [8], [9]. All input images must go through an encode path to capture information in the form of features and a decode path to return the same size as the

The associate editor coordinating the review of this manuscript and approving it for publication was Cristian A. Linte.

input in the result [9]. The bridge is used as a connecting path encode and decode [10]. The U-Net bridge usually consists of several blocks [9]. U-Net accurately predicts each pixel in image and is widely used for retinal blood vessels segmentation [8]. Numerous studies have demonstrated successful segmentation on retinal blood vessels [8], [10], [11]. Laibacher *et al.* obtained an accuracy value of 96.35% and an F1-score of 80.91% by combining the U-Net and MobileNet V2 architectures, but they did not calculate the sensitivity, specificity, or Intersection over Union (IoU) values [11]. Popat *et al.* used the U-Net architecture in conjunction with standard genetic algorithms and resulted in 95% for accuracy, 98.5% for specificity, and 80.83% for F1-score but the sensitivity was still low at 75.06% [8]. This study did not calculate performance values based on IoU. Al-masni and Kim combined U-Net with Inversion Recovery and Contextual Multi-Scale Multi-Level Network (CMM-NET) and resulted accuracy of 96.64%, specificity of 98.39%, and F1-score of 80.27%, but the sensitivity and IoU were low at 78.59% and 67.08%, respectively [10].

U-Net architecture is quite slow because the network must be run separately for each patch. U-Net has a lot of redundancy due to overlapping patches [12]. However, deeper networks can be much more efficient in terms of computation and number of parameters used. In addition, Deeper Networks are able to create deep representations, at each layer and are able to explore new, more features [13]. Unfortunately, adding layers to the U-Net architecture introduces additional parameters and increases network complexity [2], [12]. Visual Geometry Group (VGG) is one of the CNN architectures that has a deeper network [7]. Although this architecture has a deep network, it has no effect on network complexity because it uses a small convolution filter on large amounts of data [14], [15]. Unfortunately, VGG is not suitable for small datasets as it causes the network to acquire an excessive number of features [15], [16].

VGG is more commonly used in classification than segmentation [17], [18]. Khan *et al.* classified DR disease using VGG architecture and spatial pyramid pooling layers and obtained high accuracy and specificity of 85% and 91%, respectively, but the F1-score was still low at 59.6% [17]. Kaur and Gandhi used 16 layers of VGG layers in brain image classification. The results of this study obtained 100% for accuracy, sensitivity, and specificity, however, this study did not address F1-score or IoU [16]. Mateen *et al.* combined the VGG architecture with single decomposition analysis and principal component analysis in the DR disease classification. This study yielded a high accuracy of greater than 92%, but this study did not evaluate other performances [19].

Dense Connected (DenseNet) is an architecture that alleviates the problem of learning redundant features by combining feature maps from previous layers [17], [20]. The results of the feature maps from the previous layer are used as input for the next layer. The outputs of the feature maps in the next layer have the same number of filters (depth) as the previous feature maps. It can reduce a total number of parameters on the DenseNet architecture. Reusing feature maps on DenseNet can prevent the network from overlearning features [12], [17], [20]. DenseNet designs dense blocks using concatenate operations to reuse feature maps from previous layers [20], [21]. The dense block consists of three sequential operations, namely batch normalization, ReLU activation function and convolution [20]. Wang *et al.* used DenseNet for retinal blood vessel segmentation and obtained high accuracy and specificity values of 96.09% and 99.04%, respectively, but the sensitivity was still low, namely 75.39% [20]. Cheng *et al.* combined DenseNet and U-Net architectures on retinal blood vessel segmentation and obtained high accuracy and specificity values of 95.59% and 98.34%, but it gave low sensitivity [22]. Unfortunately, these studies did not assess F1-score and IoU performance.

VGG and DenseNet architectures are deep neural networks that have many parameters [23]. The complex models and long computational times are the results of an architecture that has many parameters [24]. This can lead to many new problems, including overfitting. Overfitting occurs when the model or method used has a low error in the training data but a large error in the data that has never been used (testing data) [25]. The prediction graph on overfitting means that the model predicts too accurately and memorizes the pattern in the training data, but fails to predict in the test data due to high error, thus failing to capture the overall trend of the connection [25]. To overcome complex models and overfitting problems, the dropout techniques can be used [25], [26]. Dropout is a technique to avoid overfitting while accelerating the learning process [25]. Dropout is a regularization strategy for neural networks in which certain neurons are randomly selected and not used during training [27]. These neurons can be discarded at any time. This means that the contributions of deleted neurons will be temporarily suspended, and no new weights will be applied while the neurons are training [25]. Removing a neuron from an existing network means deleting it temporarily. Several studies have shown that adding dropouts to DenseNet can improve performance in image segmentation. Lee and Lee, used DenseNet Fully Convolutional (FC) and dropout on semantic segmentation. The study obtained a high accuracy of 91.5%, although the IoU value remained low at 66.9% [26]. Wang *et al.* used the FC-DenseNet, dropout, and scSE modules for Pneumothorax (PTX) segmentation and could be able to achieve high accuracy, sensitivity, and specificity, and an F1-score greater than 88%, but this study did not discuss the result of IoU [21]. Bui *et al.* used 3D-skipDenseSeg with dropout on the convolution layer for MRI of the infant brain segmentation and produced a high F1-score above 90% although it did not measure any other performance [28].

U-Net architecture is widely used because it is suitable for image segmentation [9]. However, it has a weakness in the complexity of the network in adding layers and parameters [12]. This study proposes a new architecture, namely VG-DropDNet which combines the advantages of U-Net, VGG net and Densenet architectures. VG-DropDNet applied different architectures. In the encoded path of VG-DropDNet, the VGG architecture is used to overcome the complexity of the network and the large number of parameters. VGG has deep layers and a large number of parameters. Large parameters in VGG makes the extracted features more complex, but the computation time is faster because it uses a small convolution filter. DenseNet is used In the bridge path of VG-DropDNet because it would reduce the parameters number of VGG architecture without losing important feature information. On DenseNet, these features would be combined by concatenate operation. However, it can cause the model to be overfitting. To overcome overfitting on DenseNet, a dropout layer is inserted in each dense block. The decoded path from VG-DropDNet still uses the U-Net part because it should return the patch sizes in training to the same original image size as the input image. The combination architecture proposed in this study is evaluated using various performance measures, namely accuracy, sensitivity, specificity, F1-score, IoU, G-mean, Matthews Correlation Coefficient (MCC), and Cohen's Kappa coefficient to see how reliable, robust and valid the proposed architecture is in blood vessels segmentation on retinal images.

II. METHODOLOGY

A. DATASETS DRIVE AND STARE

The DRIVE (Digital Retinal Images for Vessel Extraction) and STARE (Structured Analysis of the Retina) datasets are used in this study. They are obtained from the https://drive.grandchallenge.org/ for DRIVE dataset website and http: www.//ces. clemson.edu/ahoover/star/ for STARE dataset website. The DRIVE dataset contains publicly available fundus camera data from 400 diabetics in the Netherlands aged 25–90 years [29]. The STARE dataset is the result of a publicly available digital fundus camera from the University of California, San Diego [29]. The DRIVE dataset is divided into two parts, namely test data and training data. Test data and training data DRIVE contained twenty images with a resolution of 565 pixels × 584 pixels. The STARE dataset contains twenty images with 700 pixels × 605 pixels resolution.

B. ENHANCEMENT OF IMAGE

Prior to beginning the segmentation process, it is necessary to perform image enhancement. Image enhancement consists of several stages, including image quality enhancement, image contrast enhancement, and noise filtering.

1) GAMMA CORRECTION

Gamma correction is used to adjust and control the brightness of individual pixels in an image; otherwise, the image appeared dark. Gamma correction is determined using the power-law transformation described in the equation (1) [30].

$$O = \left(\frac{I}{255}\right)^{\gamma} \tag{1}$$

where I and O are input and output images, respectively. The original brightness values I and O are mapped into [0, 1]. γ is the gamma value, if the value $\gamma < 1$ (encoding gamma) then the output image will look bright while if $\gamma > 1$ (decoding gamma) then the output image will look dark [31].

2) CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION

By utilizing a clip limit that acts as a contrast limiter, Contrast Limited Adaptive Histogram Equalization (CLAHE) is used to improve image quality. CLAHE divided the input image into several local areas called tiles. Clip limit is used to limit the maximum value of the histogram to a specified value. The truncated pixels would be redistributed in local regions throughout the histogram [32]. The number of pixels in each gray level would be divided equally, as explained in (2) [33]:

$$N_{avg} = \frac{N_{CR-XP} \times N_{CR-YP}}{N_{gray}} \tag{2}$$

where, N_{avg} is the average number of pixels. N_{gray} is the number of gray levels in the tile. N_{CR-XP} is the number of pixels on the X-axis of the tile. N_{CR-YP} is the number of pixels on the Y-axis of the tile. Equation (3) is used to calculate the clip limit [33].

$$N_{CL} = N_{CLIP} - N_{avg} \tag{3}$$

where N_{CL} is the clip limit and N_{CLIP} is the clip limit input value with a range of 0 to 1.

3) BOTTOM HAT TRANSFORMATION

The bottom hat transformation combines the operations of closing and image reduction [34]. By connecting adjacent pixel fragments and removing small holes in the image's center, the closing operation refines objects. This operation is carried out in two stages, namely dilation, and erosion. The bottom hat transform is used to eliminate dark objects against a light background. The equation for this transformation used equation (4) [34]:

$$A_{bot} = A \bullet B \tag{4}$$

where A_{bot} is the bottom hat transformation's image. $\bullet$ is the closing operation. A denotes the initial image. B is a structuring element in the form of an operator matrix, which take the shape of lines, disks, or diamonds.

4) IMAGE PATCHING

To overcome the need for large training data on CNN architectures, this study uses a circular random patching technique. The goal is to get the blood vessels that are inside the retinal circle. The circular random patching technique worked by generating a point $P(x_i, y_j)$ and calculating the distance from that point to the center point $P(x_0, y_0)$ of the retinal circle. If d is the diameter of the circle, r is the circle radius, the resulting $\overrightarrow{\mathbf{OP}}$ distance should be smaller than r, so the image patch generated at point has to contain retinal image blood vessels. In P the testing stage, an ordered patching technique is divided the retinal image into equal sizes according to the desired size and number sequentially starting from the pixels at the starting point of the image. The patching technique in the training stage can be seen in Fig. 1.

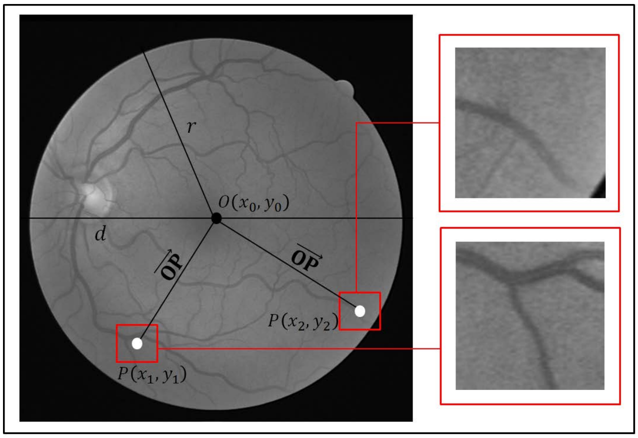


FIGURE 1. Illustration of the patching technique in the VG-DropDNet architecture for blood vessel segmentation in retinal images.

C. SEGMENTATION OF IMAGES

1) CONVOLUTION LAYER

The convolution layer is a feature extraction layer [35]. Numerous parameters are used in the convolution layer, including the filter, kernel, input depth, number of filters, stride, and padding [36]. Determination of the size of the feature map uses equation (5) [37].

$$n_{out} = \left\lfloor \frac{n_{in} + 2P - K}{S} \right\rfloor + 1 \tag{5}$$

where, n_{out} is the feature map size. n_{in} is the input height. P is padding. K is the kernel height. S is stride.

It uses equation (6), (7), (8), and (9) to calculate the outputs of the convolution layer [38], [39]:

$$C_q^p = O_q^p + b_q \tag{6}$$

where

$$O_q^p = A_p * K_q \tag{7}$$

where $*$ indicates that a convolution operation has been performed. Equation (7) is used to determine the value of each matrix entry O_q^p.

$$o_{i,j} = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} a_{u+1,v+j} \cdot k_{u+1,v+1} \tag{8}$$

Therefore,

$$c_{i,j} = \left(\sum_{u=0}^{n-1} \sum_{v=0}^{n-1} a_{u+1,v+j} \cdot k_{u+1,v+1} \right) + b_q \tag{9}$$

where $u = 0, 1, 2, \ldots, n-1$, and $v = 0, 1, 2, \ldots, n-1$. The i denotes the row, j denotes the column, n denotes the height of the kernel matrix, A denotes the input matrix, a is entry of input matrix A, k is entry of kernel matrix, b denotes the kernel bias, O denotes the convolution matrix, C denotes the convolution matrix, $q = 1, 2, 3, \ldots, Q, p = 0, 2, 3, \ldots, I, Q$ is the number of kernels used, I is the number of input matrix.

2) FUNCTION OF ACTIVATION

CNN contains a non-linear component is called the activation function. CNN's performance is enhanced through the use of the activation function. The activation function is a function used in artificial neurons to determine the output based on the input. This activation function is used in the hidden layer and output layer. In the hidden layers, the most widely used activation function is ReLU [40], [41]. The ReLU is simple and is able to determine which neurons are active so that not all neurons are used. it can reduce the number of parameters too much. Another advantage of ReLU is to reduce the likelihood of gradient loss during training. ReLU has a function to make all negative image pixel values become zero [42]. Equation (10) illustrated the ReLU activation function.

$$\mathrm{f(c)} = \max(\mathrm{c}, 0) \tag{10}$$

where f(c) is the activation function's result and c is input for ReLU.

In output layer is used sigmoid activation function. The sigmoid function is widely used in outputs that has 2 classes (binary) outputs. the sigmoid function when it returns a value in the region of negative infinite $(-\infty)$ and positive infinity $(+\infty)$ as in a linear function, it will return the value in the range $(0, 1)$.The sigmoid activation function occupies only the range from 0 to 1 and is asymptomatic in both values. it is very useful for binary classification with 0 and 1 as output [39], [43]. Equation (11) illustrates the Sigmoid function [44].

$$\mathrm{g(z)} = \frac{1}{1 + e^{-z}} \tag{11}$$

where g(z) is the activation function's result, z is input for ReLU, $\mathrm{g(z)} \in [0, 1]$ and $z \in [-\infty, \infty]$.

3) BATCH NORMALIZATION

When the CNN network is trained, the input will change on each layer. The process works in a long training time [45]. This problem can be solved by adding a layer of batch normalization [46]. Batch Normalization uses a mini-batch stack to normalize the input on each layer [45]. The batch normalization procedure begins by calculating the average of each mini-batch (μ_j) using equation (12) [45]:

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_{i,j} \tag{12}$$

For $i = 1, 2, 3, \ldots, m$, and $j = 1, 2, 3, \ldots, n$. Where x is entry matrix input, m is the number of mini-batch data (rows), and n is the number of mini-batch batches (columns). Then, the variance value is calculated in one mini batch $\left(\sigma_j^2\right)$ using equation (13) [45]:

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_{i,j} - \mu_j\right)^2 \tag{13}$$

The final normalization is calculated of input in one mini Batch $\left(\hat{x}_{i,j}\right)$ using equation (14) [45]:

$$\hat{x}_{i,j} = \frac{x_{i,j} - \mu_j}{\sqrt{\sigma_j^2 + \varepsilon}} \tag{14}$$

where, ε is a small positive constant to improve numerical stability.

4) MAX POOLING

Pooling layers has identical stride filter sizes. The benefit of using the pooling layer is the computations are performed more quickly. The most frequently used pooling layer method is max pooling [47]. Max pooling uses the filter's maximum input value to generate output with a smaller size [48].

5) DROPOUT

To handle overfitting in this study, it uses dropout. Overfitting is a frequent occurrence in CNN architecture. During training, dropout will randomly remove some neurons based on the dropout rate used [49]. The omitted neuron's value will be set to 0. In this study, a dropout layer with a rate of 50% is used.

6) CONVOLUTIONAL TRANSPOSED

In deep networks, the up-sampling layer is used to increase the dimension of features. Numerous techniques, including transposed convolution, can be used as an up-sampling layer on a CNN. Transposed convolution enlarges the feature map by returning the convolution input value [50]. The steps of the transposed convolution process are to filled the position between each matrix entry with 0, where the number of positions is the step value minus 1 [51]. All entries of the input matrix are incremented by 0 according to the padding rule. The number of increments of 0 is the size of the kernel matrix minus 1. The first and last rows, as well as the first and last columns of the input matrix are removed according to the padding used. Then, the input matrix is carried out with a convolution operation on the input matrix with the kernel matrix rotated 180 degrees.

7) CONCATENATE

Concatenate is used to combine two input matrices from two feature maps into a single larger input matrix [9].

8) BINARY CROSS-ENTROPY

A loss function can be used to maximize and study the network quickly and accurately in CNN. Binary cross-entropy is a loss function that is frequently used for classification and segmentation purposes due to its effectiveness [60]. In retinal blood vessel segmentation, the loss function is used to determine the error difference between the segmented image and the ground truth [53].

D. TRAINING

The purpose of this stage is to produce the best weights that would be used in the testing stage. The data used are 20 retinal images from DRIVE dataset and 20 retinal images from the STARE dataset for training data. The training dta will be divided to training data and validation data for each dataset. The process that occured in the training stage can be seen in Fig.2. Based on Fig.2, the original image is improved by using gamma correction, CLAHE, and bottom hat transformation. The circular random patching technique is applied to reproduce the data. The patching technique divides each pre-processed image into 10000 small pieces measuring each 64 pixels × 64 pixels, so that a total of 200000 data is obtained for STARE dataset and DRIVE training data. The total data is divided into 64% for the training data, 16% for validation data and 20% for testing data for each dataset.

E. TESTING

The weight results from the training process will be used in the testing process to see how far the architecture has succeeded for blood vessel segmentation in retinal image with new data input from the STARE dataset and the DRIVE test data. New data is data that has never been used at all in the training process. For each dataset, 20 images are used. As illustration of the process carried out in the proposed method could be seen in Fig.3.

Based on Fig.3, the input image is first performed with the ordered patching technique and enhancements as in the training process. After that, the weights from the training results are used for the testing process. The results of the testing process are still in the form of images of small pieces so that reconstruction is carried out. The results of blood vessel segmentation are compared with ground truth in both of datasets to measure the performance of the proposed architecture VG-DropDNet.

F. EVALUATION

In blood vessel segmentation, each pixel is classified into two categories, namely blood vessels and non-vessels (background). In the confusion matrix, the following four results can be found, namely TP, FN, FP and TN. This study evaluates the performance of the proposed architecture with a confusion matrix is included accuracy (Acc), sensitivity (Se), specificity (Sp), IoU, F1-score (F1), G-mean, MCC, and Cohen's Kappa coefficient (K). These performance are defined in equations (15), (16), (17), (18), (19), (20), (21), (22) [54], [55], [56].

$$Acc = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \tag{15}$$

$$\mathrm{Se} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{16}$$

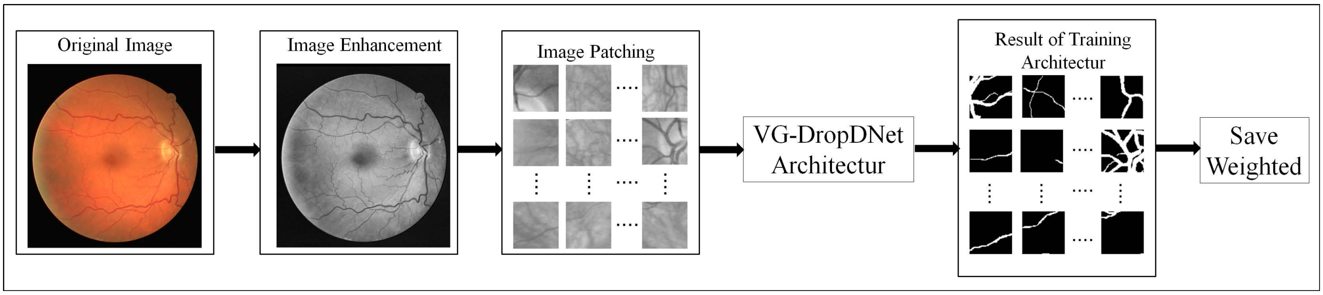


FIGURE 2. Illustration of the traning process in the VG-DropDNet architecture for blood vessel segmentation in retinal images.

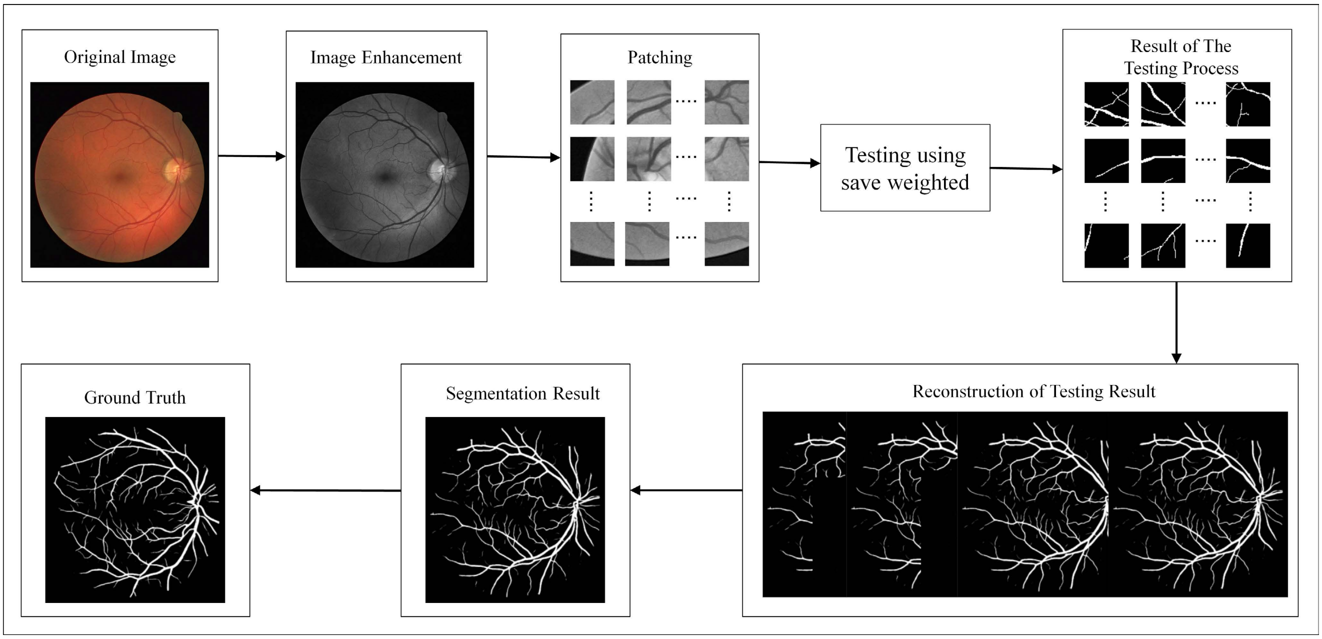

Original Image
Image Enhancement
Patching
Testing using save weighted
Result of The Testing Process
Reconstruction of Testing Result
Segmentation Result
Ground Truth

FIGURE 3. Illustration of the testing process in VG-DropDNet architecture for blood vessels segmentation in retinal image.

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \tag{17}$$

$$IoU = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \tag{18}$$

$$F1 = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}} \tag{19}$$

$$G-mean = \sqrt{Se \times Sp} \tag{20}$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{(TP \times FP)\,(TP \times FN)\,(TN \times FP)(TN \times FN)} \tag{21}$$

$$K = \frac{P_o - P_c}{1 - P_c} \tag{22}$$

The true positive (TP) is a properly classified blood vessels pixel. The false negative (FN) is a blood vessels pixel that is incorrectly classified as a background pixel. The true negative (TN) is a correctly classified background pixel. The false positive (FP) is a background pixel that are incorrectly classified as blood vessels. P_c is the increase in the number of possible changes in the number of blood vessels and background pixels. P_o is the percentage of pixel counts that are consistent on the segmentation result

The accuracy is used to calculate the accuracy of an architecture [66]. However, accuracy is not always suitable for measuring model performance, especially on unbalanced data. Class imbalance between blood vessel pixels and background pixels in the dataset requires another performance measure [56]. Sensitivity assesses performance in predicting blood vessels pixels [66]. Specificity assesses performance in predicting background pixels [56]. IoU uses to measure the similarity between the predicted segmentation image and the ground truth image [55]. Performance measures that is suitable for used in unbalanced pixel classes are F1-score, G-Mean, and Matthews Correlation Coefficient (MCC) [75]. F1-score is assessed performance on unbalanced datasets and widely used in segmentation [56]. G-Mean is a statistical measure used to measure the balance between blood vessel pixels with a background [59]. MCC is the correlation between predicted pixels and ground truth [75]. The Cohen's Kappa coefficient measures the intensity of the agreement

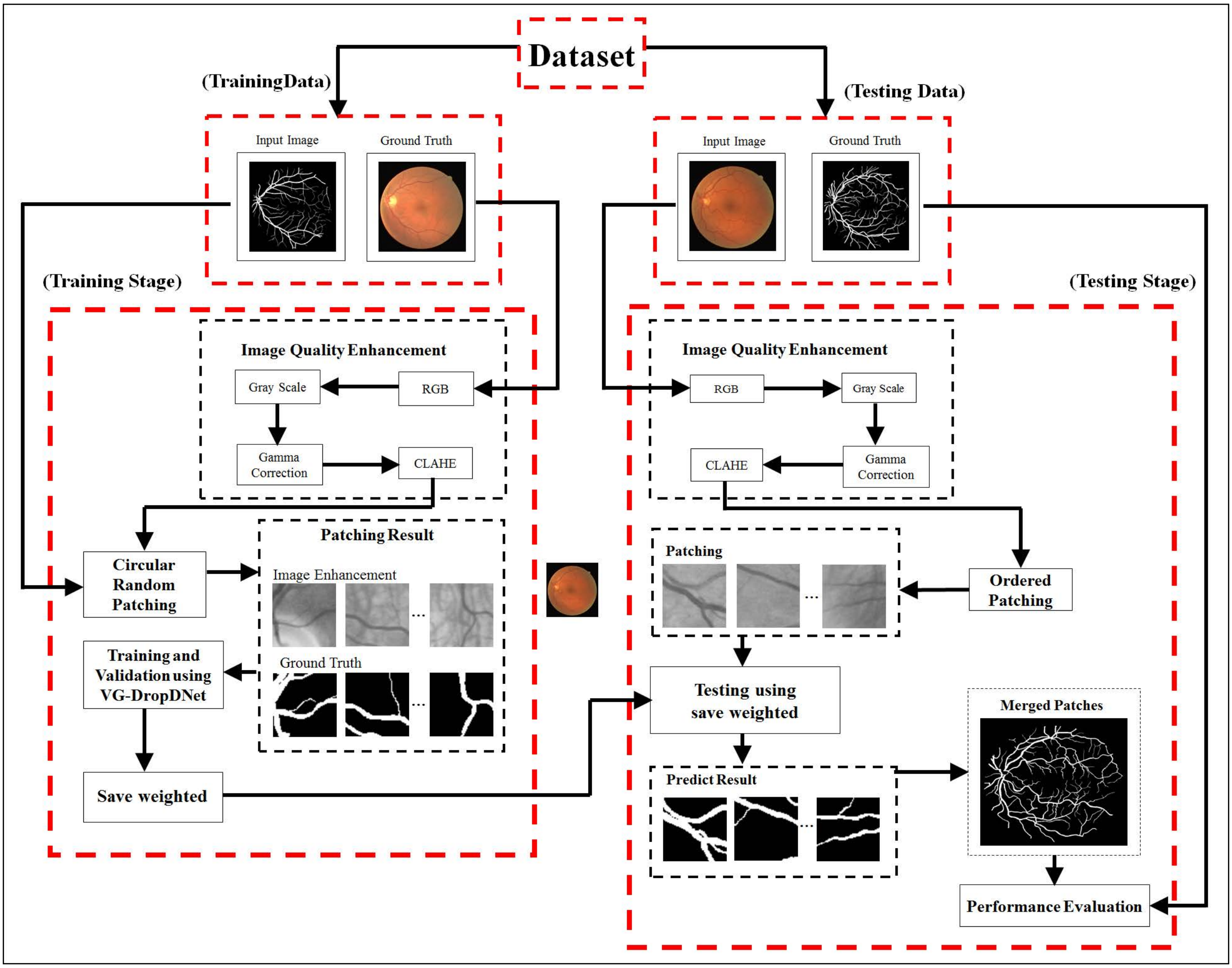


FIGURE 4. Proposed method stages of retinal image blood vessel segmentation used the VG-DropDNet architecture.

between the results of the proposed segmentation architecture and those manually performed by experts (ground truth) [54].

III. RESULT

In this study, there are 2 stages carries out, namely training and testing. The stages in this research as a whole could be seen in Fig.4. the training process begins with image quality enhancement, Circular random patching, training and validation using VG-DropDNet. For the testing process from Fig.4, the steps carried out are image quality enhancement, Ordered Patching Technique, testing using the latest input dataset, Reconstruction of segmentation results, performance evaluation using Acc, Se, Sp, IoU, F1, G-mean, MCC, and Cohen's Kappa coefficient. The stages of Image quality enhancement consist of RGB, Grayscale, Gamma Correction, and CLAHE.

A. IMAGE ENHANCEMENT

The data used in this study are BGR images from the DRIVE and STARE datasets which have dimensions of 565 pixels × 584 pixels for DRIVE and 700 pixels × 605 pixels for STARE. The image enhancement process can be seen in Fig.5. In Fig.5 it can be seen that the input data is in the form of a BGR image, therefore it is necessary to convert it to RGB image. Furthermore, to simplify the image processing process, it is necessary to convert the image to grayscale format. After that, the image contrast is improved so that the retinal blood vessel network looked clearer using gamma correction and CLAHE. The final image from the contrast enhancement process still contains noise that can interfere with segmentation. To overcome this, a bottom hat transformation operation is used.

B. THE VG-DROPDNET ARCHITECTURE

The VG-DropDNet architecture is proposed to obtain a new architecture that combined the advantages of VGG, DenseNet, and dropout architectures to overcome the limitations of U-Net so the proposed architecture is robust and had stronger performances for blood vessels segmentation in

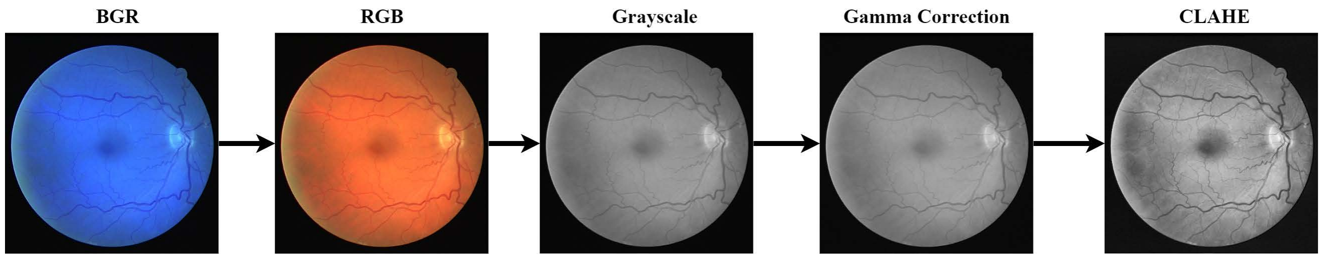


FIGURE 5. Stages of retinal image enhancement quality for the input image on the VG-DropDNet architecture.

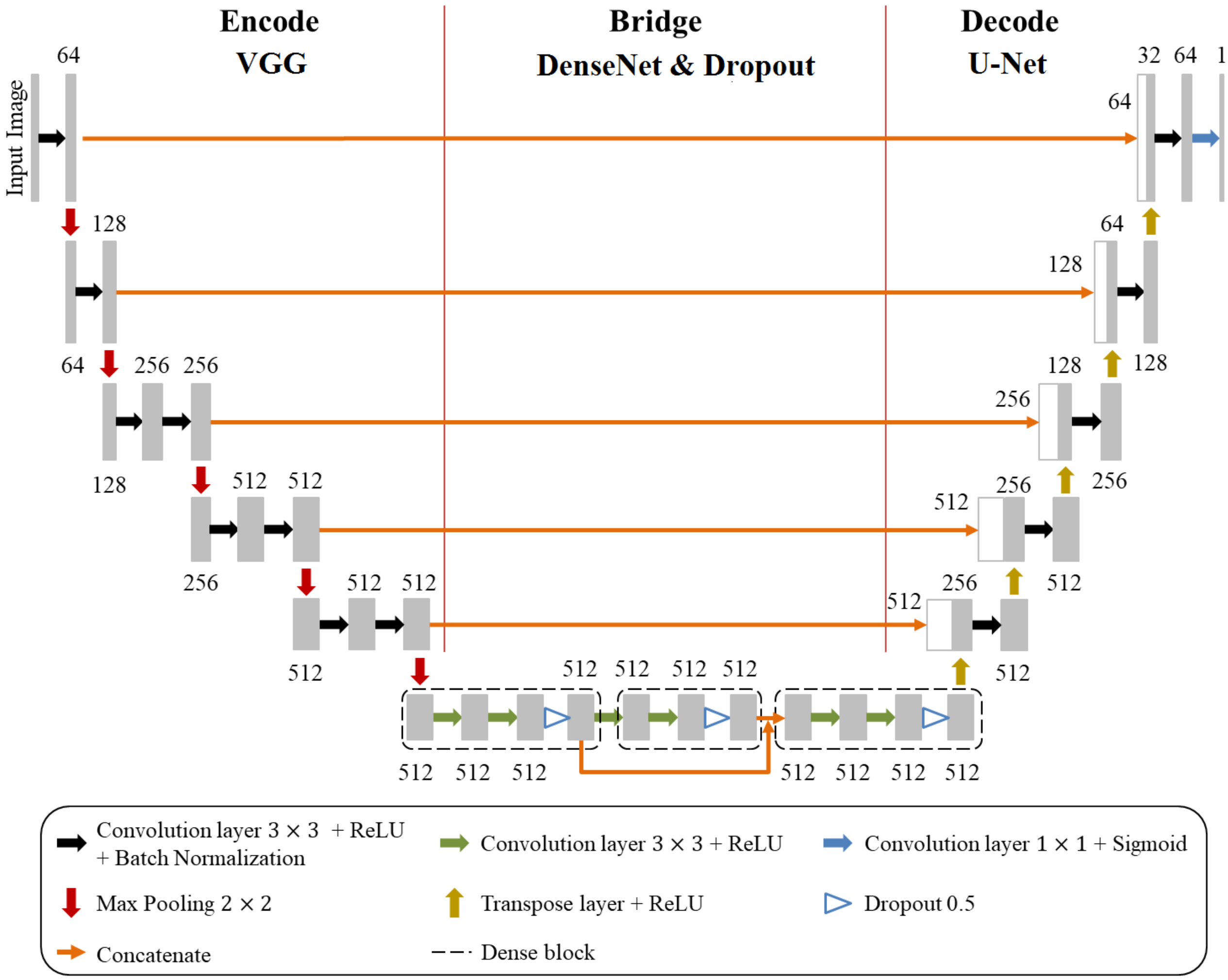
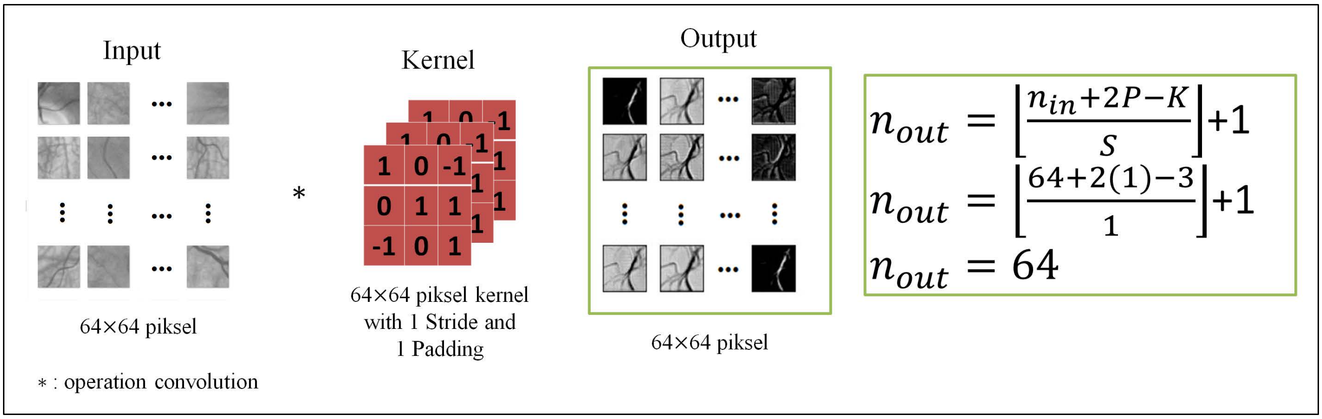


FIGURE 6. Model architecture of VG-DropDNet that contained of 3 parts for blood vessels segmentation on retinal image.

retinal images. The basic architecture used in VG-DropDNet is a modified U-Net. The modifications make the proposed architecture consisted of 3 parts, namely the encoder, bridge and decoder parts. On the encoder part, it used the VGG architecture with the aim of getting more features with deep networking. On the bridge, it used DenseNet to be able to call information on the previous layer which is combined with dropout to overcome overfitting on DenseNet caused by excess features of the concatenate operation. On the decoder part, it still used the U-Net architecture to make it easier to restore the image patched size to its original image size, namely the size used by the input image. A model of VG-DropDNet architecture can be seen in Fig.6.

In Fig.6, it can be seen that the architecture added a connecting line between the encoder line and the decoder line. The steps in the first layer convolution block for the encoder


Input
Kernel
Output
64×64 piksel
64×64 piksel kernel with 1 Stride and 1 Padding
64×64 piksel
∗ : operation convolution

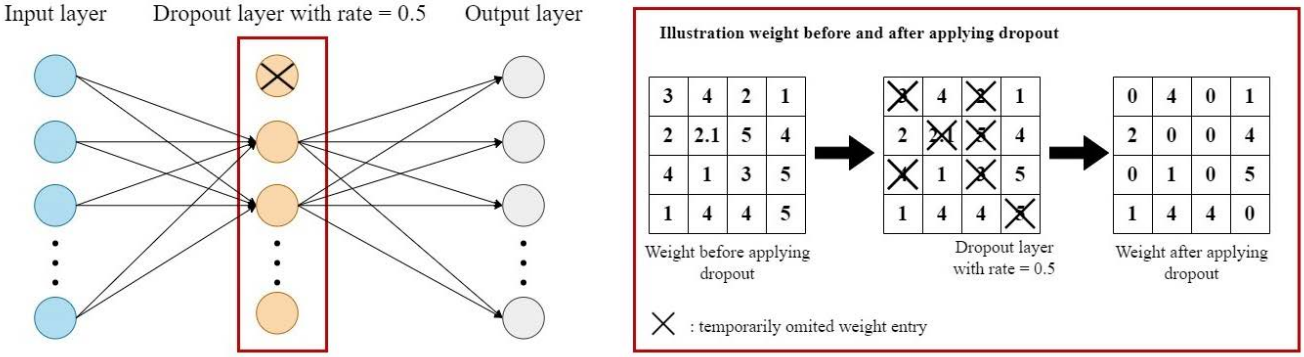
1	0	-1
0	1	1
-1	0	1

$$n_{out} = \left\lfloor \frac{n_{in}+2P-K}{S} \right\rfloor + 1$$

$$n_{out} = \left\lfloor \frac{64+2(1)-3}{1} \right\rfloor + 1$$

$$n_{out} = 64$$

FIGURE 7. Sample of some data in the first convolution layer with 3 × 3 of kernel size and 64 × 64 of a feature map.


Input layer
Dropout layer with rate = 0.5
Output layer
Illustration weight before and after applying dropout
Weight before applying dropout
Dropout layer with rate = 0.5
Weight after applying dropout
: temporarily omited weight entry

3	4	2	1
2	2.1	5	4
4	1	3	5
1	4	4	5

$\cancel{3}$	4	$\cancel{2}$	1
2	$\cancel{2.1}$	$\cancel{5}$	4
$\cancel{4}$	1	$\cancel{3}$	5
1	4	4	$\cancel{5}$

0	4	0	1
2	0	0	4
0	1	0	5
1	4	4	0

FIGURE 8. Example of a simple calculation on the dropout layer on the proposed architec-ture VG-DropDNet.

path, namely the pre-processed image is used as input. Then a convolution operation is performed using a 3 × 3, 1 same padding, 1 stride and the number of filters used is 64. The convolution operation is carried out simultaneously with the ReLU activation function. After that, normalization is carried out using batch normalization. An example of the result of a feature map on the first convolution layer block can be seen in Fig.7. From Fig.7, it can be seen that in the convolution layer block using input measuring 64 pixels × 64 pixels ($n_{in} = 64$), 3 × 3 kernel ($K = 3$), 1 stride ($S = 1$), and padding same ($P = 1$), the output feature size is 64 pixels × 64 pixels ($n_{out} = 64$).

Furthermore, the dimension map of the feature map is reduced using a max pooling size of 2 × 2 with 2 strides. In the second to fifth convolution layer blocks, the same steps are carried out as the first layer convolution block using the number of filters, namely 128, 256, 512, and 512. The size of the input image used in the encoder line is 2^n because in the max polling process the features would be divided by 2. The number of max polls used in this architecture is 5 so that the input image should have size of 64 pixels × 64 pixels. 64 pixels × 64 pixels image size is used because at each layer max poll the image size would be reduced as much as 2^{n-1}. If the 32 pixels × 32 pixels image input size is used, then in the last layer the feature size would be 0, so the feature could not be used as input for the connecting line.

The bridge path is based on the DenseNet architecture using three dense blocks. It hoped more features would be obtained. However, DenseNet's concatenate function would cause model overfitting. To overcome overfitting, it added a dropout layer on each dense block. In the first dense block and the second dense block, the convolution operation and the ReLU activation function are performed twice using a 3 × 3 kernel matrix, the same padding, and 512 filters. After that, the convoluted feature map would enter the dropout layer with a drop rate of 0.5 so that 50% of the feature map would be omitted temporarily by the computer. The value of the omitted feature map would be set to 0 to avoid overfitting during training. For the third dense block, inputs from the first and second dense blocks are used which are combined using concatenate. An example of a simple calculation on the dropout layer could be seen in Fig.8.

Based on Fig.8. supposed the weight or kernel in the dropout layer is 4 × 4. The dropout rate used is 0.5, which means it would randomly remove 50% of the weighted entries from the total weighted entries. From Fig.8, it could be seen that the total weighted entries are 16 so that 8 weight entries would be eliminated. The weight entries to be omitted are marked with a red cross in figure 4. The value of the omitted weight entries would be filled with a value of 0, so there are 8 entries with a value of 0. Some examples of feature map results on the bridge path of the VG-DropDNet architecture can be seen in Fig.9.

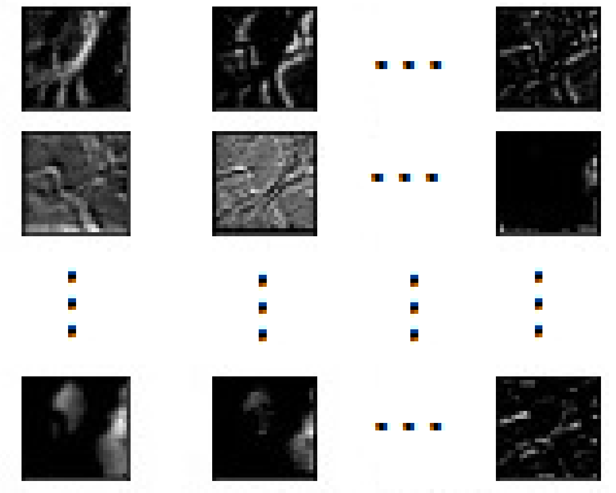


FIGURE 9. Example of the result of a feature map on the bridge path of the VG-Drop Net architecture.

Based on Fig.6 for the decoder path, it still used U-Net Architecture because it returned the same size as the input on the output. The steps in the first transposed convolution block for the decode path, namely the feature map resulting from the third dense block, are used as input. Then transposed convolution is performed using a 3 × 3 kernel matrix, the same padding, 2 stride and 256 filters. The output of the transposed convolution is substituted into the ReLU activation function. Some samples of feature map results in the first transposed convolution block can be seen in Fig.10. The feature map merging operation of the encode and decode paths is performed using the concatenate operation. The next step is to perform a convolution operation again using a 3 × 3 matrix, the same padding, 1 stride, and 512 filters. The results of the convolution operation are normalized using batch normalization. In the second, third, fourth and fifth transposed convolution layers, the same steps are carried out as the first block using the number of filters for each block, namely 256, 128, 64, and 64. The last step in the decode path is a convolution operation using a 1 × 1 kernel matrix and substitution with Sigmoid activation function.

C. TRAINING STAGE

In the training stage, it is carried out to produce weights that would be used in the testing stage. This research used 50 epochs, 8 batch sizes, and 20000 iterations. Small pieces of the image resulting from the patch technique would be used as input to the CNN architectural model. In the CNN architectural model, the first step is to initialize the weight value of each epoch. For each epoch the loss (error) value should be calculated. The weight should be saved if the error value in the validation data is smaller than the previous epoch, otherwise, the weights should be updated for the next epoch to produce the best weight. The line graph of the results of the test data training process and validation for the error (loss) and accuracy values could be seen in figure 0.

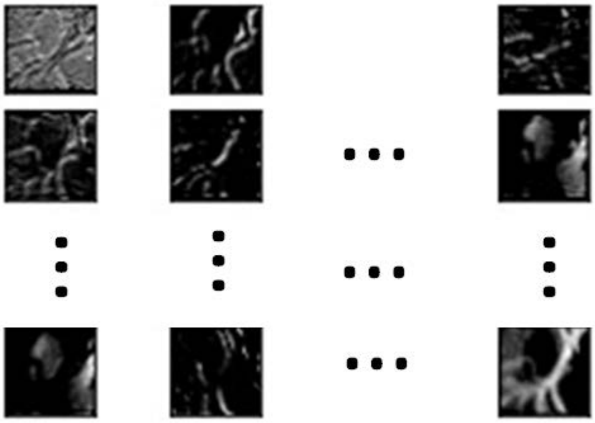


FIGURE 10. Example of the result on the first transposed convolution block feature map in the VG-DropDNet architecture.

In figure 11(a) it can be seen that the line graph of the error (loss) value on the VG-DropDNet architecture for training data and validation data on the DRIVE dataset always decreases until the 50th epoch with the loss value closed to 0. The results of the loss graph in Fig.11(a) shows no overfitting in the DRIVE dataset. Loss values of the training data and validation data has a similar distribution. It indicates that the model generated on the DRIVE dataset is able to recognize other data patterns well even though the data has never been trained before. On the STARE dataset from figure 11(b) it can also be seen that the line graph of the error (loss) value on the VG-DropDNet architecture for training data and validation data always decreased until the 50^{th} epoch. The training loss value on STARE is closed to 0.2. Training on the STARE dataset also did not experience overfitting between training data and validation data. This indicated that the model used in the STARE dataset is able to predict well for other image data. The graph of the results of the test data training process and validation for the accuracy value could be seen in Fig.12.

Graphs of accuracy on training data and accuracy on validation data for each DRIVE and STARE dataset can be seen in Fig.12(a) and Fig.12(b). From Fig.12 it can be seen that the resulting accuracy increases with the number of epochs up to the 50th epoch. The accuracy graphs on the DRIVE and STARE datasets does not experience overfitting because they have a similar distribution in both accuracy on training data or accuracy on validation data. The model obtained from

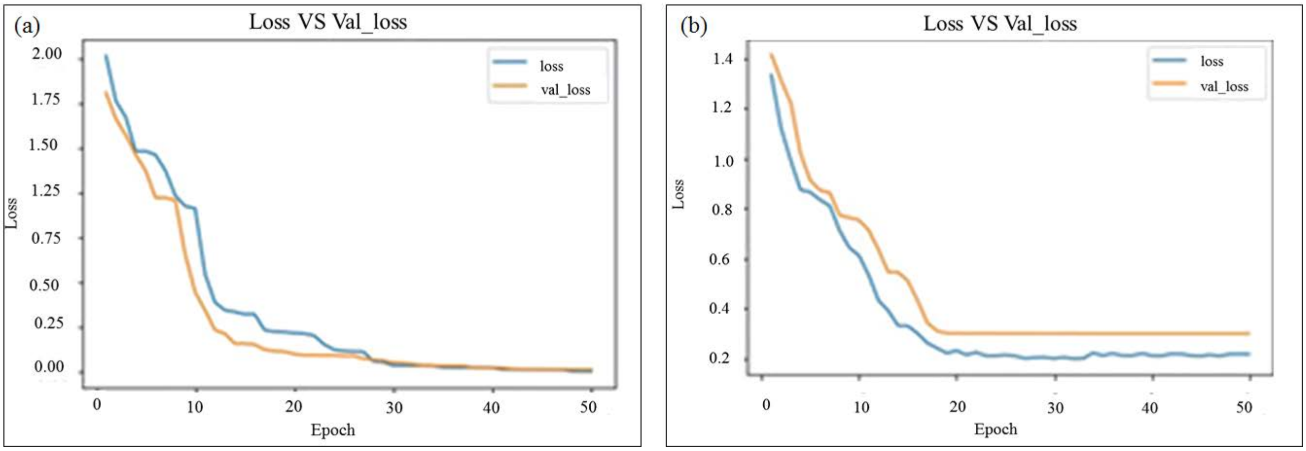


FIGURE 11. Graph of the results of the training process error value (loss) of training data and validation using VG-DropDNet on the dataset (a) DRIVE (b) STARE.

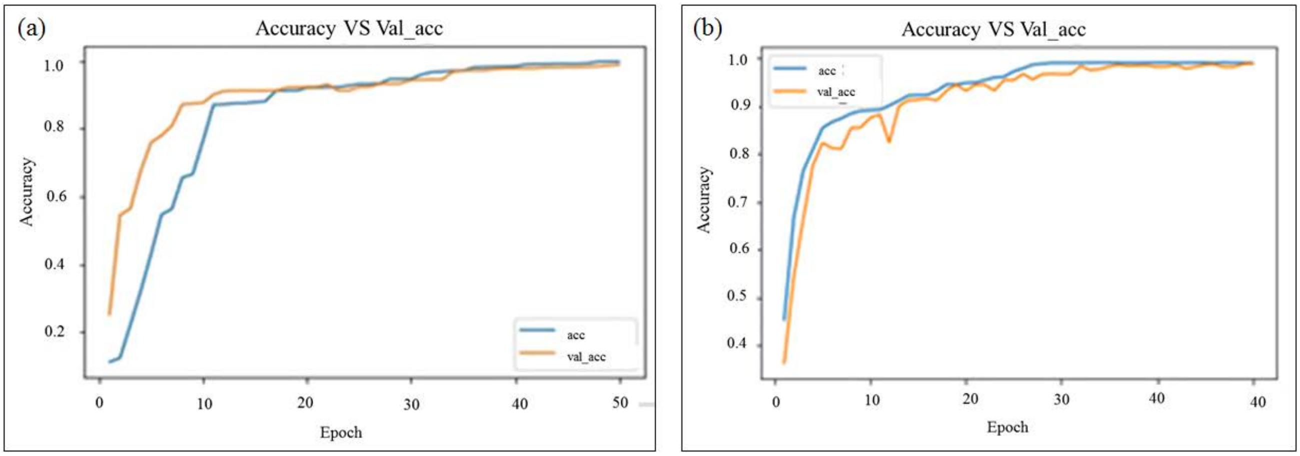


FIGURE 12. Graph of the results of the training process, the value of training data accuracy and validation using VG-DropDNet on the dataset (a) DRIVE (b) STARE.

the training process also does not experience a vanishing or exploded gradient. Based on Fig.12(a) and Fig.12(b), it can be seen that the proposed architecture has excellent work because the accuracy values are above 99% for the DRIVE and STARE datasets. It means that the model obtained from the VG-DropDNet architecture has excellent level of accuracy for blood vessels segmentation in retinal images.

D. TESTING STAGE

At the testing stage, testing is carried out using new data to see how successful the VG-DropDNet architecture is used for blood vessels segmentation on retinal image. At this stage, the weights obtain from the training results are applied to segment retinal blood vessels in the testing data. The image from the testing will be compared with the ground truth image in each dataset. The comparison of the results of the test stage and ground truth can be seen in Table 1 for DRIVE dataset and Table 2 for STARE dataset.

Table 1 and Table 2 compared the results on segmentation of the VG-DropDNet architecture with the available ground truth. It can be seen from the proposed that the blood vessel segmentation in the retinal image is successfully compared to the ground truth image, but some retinal thin vessels in the ground truth are not detected in the segmented image as in files Test_07, Im0324, Test_03, Im0003, and Im0004.

The performance value could be done by comparing the segmentation results in testing stage with ground truth image using a confusion matrix table for each data test. A Cohen's Kappa coefficient value showed that the segmentation results from the VG-DropDNet architecture are very suitable and in accordance with the ground truth results carried out by experts on the STARE dataset. It explains that the VG-DropDNet architecture has a very good performance on the STARE dataset compared to the DRIVE dataset.

From the confusion matrix obtained, the performance of the VG-DropDNet architecture on DRIVE can be measured based on the values of accuracy, sensitivity, specificity, F1-score and IoU. The performance results on the DRIVE dataset are 95.36% for accuracy, 79.94% for sensitivity, 97.61% for specificity, and 68.70% for IoU. The IoU value

TABLE 1. Example comparison between the images obtained in the testing stage with the ground truth images on the drive dataset.

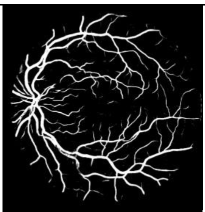
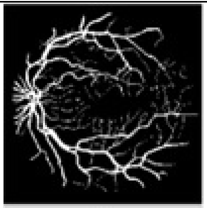
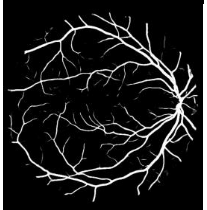
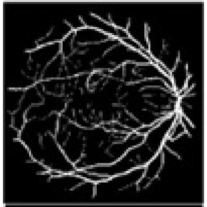
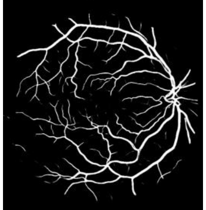
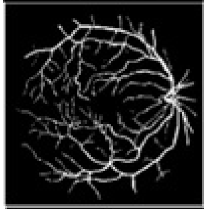
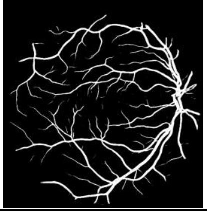
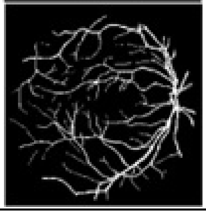
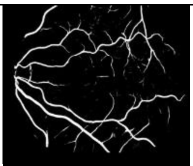
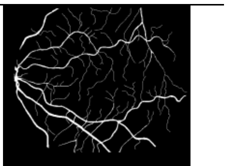
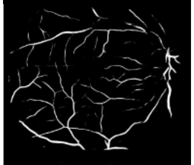
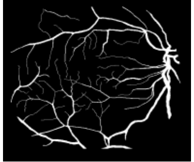
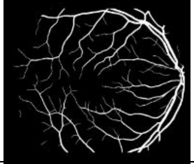
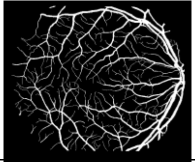
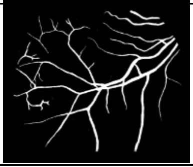
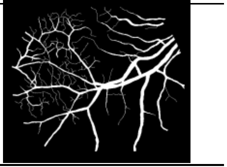
Nama File	Segmentation	Ground Truth
Test_03		
Test_07		
Test_17		
Test_18		

TABLE 2. Example comparison between the images obtained in the testing stage with the ground truth images on the stare dataset.

Nama File	Segmentation	Ground Truth
Im0003		
Im0004		
Im0082		
Im0324		

is still low, but the Cohen's kappa coefficient obtains for the DRIVE dataset is very good, namely 0.8386.

From the Cohen's kappa coefficient value obtained, it is concluded that the results of the VG-DropDNet architecture had a very strong intensity of agreement with the ground truth carried out by experts for the DRIVE dataset.

The results of the VG-DropDNet architecture performance on the STARE dataset are 98.56% for accuracy, 91.24 for sensitivity, 99.41% for specificity, and 86.90% for IoU. All performance results on the STARE dataset are surprisingly excellent compared to the results on the DRIVE dataset. A Cohen's Kappa coefficient obtained for the STARE dataset is also excellent at 0.98. This value is higher than the Cohen's Kappa coefficient in the DRIVE dataset. The Cohen's Kappa coefficient shows that the segmentation results from the VG-DropDNet architecture are very suitable and in accordance with the ground truth results carried out by experts on the STARE dataset. It explains that the VG-DropDNet architecture had a very good performance on the STARE dataset compared to the DRIVE dataset.

Another performance is Receiver Operating Characteristics (ROC) Curves. ROC is a probability that summarizes the performance of the confusion matrix on all threshold values. Area Under the Curve (AUC) converts the ROC curve to numeric to measure performance in binary classification by taking values between 0 and 1. The closer to one AUC value, the better the model works. Fig.13 shows the ROC curve and AUC values for DRIVE and STARE. AUC is calculated by the ROC. From Fig.13(a) it can be seen that the ROC graph in the DRIVE dataset is closer to the random classifier diagonal line than Fig.13(b) in the STARE dataset. This causes the AUC value in the DRIVE (0.9309) dataset to be smaller than the STARE (0.9696). The AUC value on STARE is better than on DRIVE, meaning that the model works better on the STARE dataset. STARE has a larger image size. the patching process performed on STARE provides a greater opportunity to obtain more and clearer features of blood vessel in retinal images. This can affect the performance of STARE which is better than on DRIVE. Fig.13(a) describes about the ROC and AUC on DRIVE dataset by VG-DropDNet. Fig.13(b) describe about the ROC and AUC on STARE dataset by G-DropDNet.

IV. DISCUSSION

The VG-DropDNet architecture is used to segment retinal blood vessels in this work. Confusion matrix can be used to calculate architectural performance based on accuracy, sensitivity, specificity, and IoU. The comparison of the segmentation results of this study with other studies on the DRIVE dataset is shown in Table 3.

Table 3 contains some results from the last 5 years of research using the DRIVE dataset for blood vessels

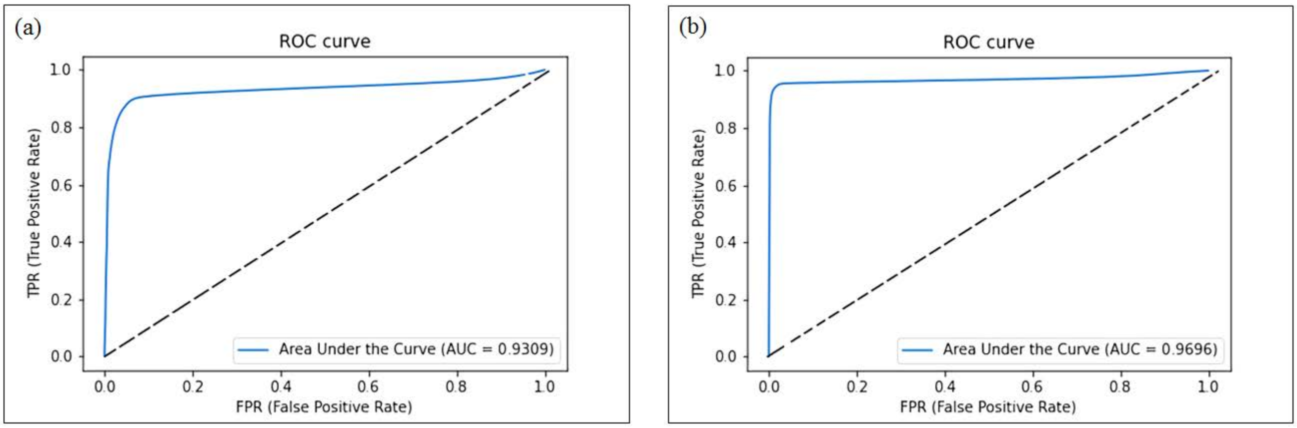


FIGURE 13. Graph of the receiver operating characteristic (ROC) curve obtained during the testing process on VG-DropDNet for the (a) DRIVE (b) STARE dataset.

segmentation on retinal image. In Table 3 the highest accuracy is obtained by Al-masni and Kim [10], but the sensitivity value obtained is still below the sensitivity value of the VG-DropDNet architecture. The highest specificity value is obtained on the architecture carried out by Wang *et al.* [20], but the study does not measure IoU values. The architecture proposes by Zhang *et al.* [60] provided the highest sensitivity among other studies. Unfortunately, this study does not measure IoU. The VG-DropDNet architecture provided the highest IoU, but the IoU value still needed to be improve. Although the values of accuracy, sensitivity and specificity are below the results of other studies, the VG-DropDNet has a good ability to segment blood vessels and other features as background because the accuracy, sensitivity and specificity values obtained are above 75%. The comparison of the segmentation results of this study with other studies on the STARE dataset is shown in Table 4.

Table 4 is a comparison of the results of the last 4 years of studies using the STARE dataset for blood vessels segmentation on the retinal image. As shown in Table 4, the proposed method achieves the highest accuracy. In terms of sensitivity, and specificity, the proposed method also outperforms other researchers by more than 90%. The highest IoU value is obtained from the researcher Guo *et al.* [58], but the IoU of the proposed method has also achieved good results exceeding 80%.

According to the comparison results in Table 3 and Table 4, the VG-DropDNet architecture for blood vessels segmentation on the retinal image has a good performance on DRIVE dataset. On the STARE dataset, the VG-DropDNet architecture is excellent and robust for blood vessels segmentation on the retinal images. The ability of VG-DropDNet to detect the bounding box between blood vessels and background in DRIVE dataset should be improved because the IoU is still under 70%. In STARE dataset, the IoU is excellent at detecting the bounding box indicates by the IoU value greater than 80%.

TABLE 3. Performance comparison of VG-dropdnet for blood vessels segmentation on retinal image in drive.

Method	Year	Acc (%)	Se (%)	Sp (%)	IoU (%)
CNN 9 Layers[61]	2019	94.90	78.27	97.60	67.16
ResNet[62]	2019	95.56	77.26	98.13	-
U-Net and MobileNetV2[11]	2019	96.35	-	-	-
DenseNet and Patching Techniques [20]	2019	96.90	75.39	**99.04**	**-**
DenseNet dan U-Net[22]	2020	95	76.72	95	-
U-Net and Algoritma Genetika[8]	2020	95	75.06	98.5	-
DG-Unet[63]	2020	96.04	76.41	98.37	-
CMM-Net[10]	2021	**96.64**	78.59	98.39	67.08
PyramidU-Net[60]	2021	96.22	**82.13**	99.32	-
Bridge-Net[64]	2022	95.65	**78.5**	96.18	
VG-DropDNet	**2022**	95.36	79.74	97.61	**68.70**

TABLE 4. Performance comparison of VG-DropDNet for blood vessels segmentation on retinal image in STARE.

Method	Year	Acc (%)	Se (%)	Sp (%)	IoU (%)
Residual Block Incorporated U-Net [65]	2018	95.37	55.82	98.62	47.80
Dilated Multi-Scale CNN[29]	2019	97.81	82.49	99.04	73.79
DenseNet and Patching Techniques [20]	2019	97.04	79.14	97.22	-
Modied Residual U-Net[66]	2019	-	81.01	97.93	68.73
SD-U-Net[58]	2019	98.50	75.41	98.99	**97.63**
IterNet[67]	2020	96.41	77.15	98.86	-
BSEResU-Net[68]	2021	95.43	74.97	98.42	-
PCAT-UNet[69]	2022	97.96	87.03	99.37	-
Bridge-Net[64]	2022	96.68	80.02	98.64	-
EDLFDPRS Frame Work[70]	2022	97.33	84.27	98.57	-
VG-DropDNet	**2022**	**98.56**	**91.24**	**99.41**	86.90

The number of Blood Vessels and non-blood vessels (background) pixels in the images is not balanced. The statistic

TABLE 5. Comparison of MCC, G-mean, and F1-score results for the DRIVE Dataset on The VG-DropDNet architecture with other studies.

Method	Year	MCC	G-mean	F1
FC-CRF[71]	2017	0.7556	0.8741	0.7857
CRF[72]	2017	0.7556	0.883	0.7942
BTS-DSN[73]	2019	0.7964	0.8795	0.8249
SIDNet[74]	2020	0.7717	0.8807	0.801
Hybrid Unet[75]	2021	**0.8685**	0.8622	**0.8285**
Bridge-Net[64]	2022	0.7982	0.8775	0.82
VG-DropDNet	**2022**	0.7881	**0.8836**	0.8144

TABLE 6. Comparison of MCC, G-mean, and F1-score results for the STARE dataset on The VG-DropDNet architecture with other studies.

Method	Year	MCC	G-mean	F1
FC-CRF[71]	2017	0.7417	0.8628	0.764
CRF[72]	2017	0.783	0.8859	0.8017
BTS-DSN[73]	2019	0.8221	0.899	0.8421
SIDNet[74]	2020	0.7738	0.88	0.7866
M-Gan[76]	2020	0.8306	0.9095	0.8324
DBFU-Net[77]	2022	0.8332	0.9198	0.8521
Bridge-Net[64]	2022	0.8147	0.8868	0.8289
VG-DropDNet	**2022**	**0.9222**	**0.9239**	**0.9299**

performances for unbalanced data are MCC, G-mean and F1-score. MCC is used to measure the correlation between two classes. Geometric Average (G-mean) is a metric that measures the balance between majority and minority data. The F1-Score is used to measure whether the classification results have good precision and recall. The results sought to maximize the accuracy of each class to balance. The results obtain in the DRIVE dataset are 0.7881 for MCC, 0.8833 for G-mean and 0.814 for F1-score. The results in the STARE dataset are 0.922 for MCC, 0.9239 for G-mean and 0.9299 for F1-score. The results of the MCC, G-mean and F1-score are all close to 1. To see how well the results of the MCC, G-mean and F1-score produced by VG-DropDNet, these results are compared with the results of other studies as in Table 5 and Table 6.

In table 5, the highest MCC and F1-score for DRIVE are obtained by Nagdeote and Prabhu [75]. However, the highest G-mean is obtained by the VG-DropDNet architecture. The MCC on the VG-DropDNet architecture is higher than the results of Zhuo *et al.* [72] and Orlando *et al.* [71]. Although the MCC is lower than the results of Nagdeote and Prabhu [75], and Guo *et al.* [73], the VG-DropDNet architecture on DRIVE is able to have a good balance for blood vessels and non-blood vessel (background) segmentation on retinal images. It is indicated by the results of the MCC, G-mean, and F1-score which are all close to 1 on DRIVE. In table 6, the results on the STARE dataset are excellent. VG-DropDNet give the highest MCC, G-mean, and F1-score results compared to other research results in table 6. The VG-DropDNet has good precision and recall in blood vessels segmentation in retinal images especially on STARE. From the results on both datasets, it is concluded that the proposed method, VG-DropDNet is robust, valid and has great balance model for blood vessels segmentation on retinal images. The Vg-DropDNet provides the accurate valid blood vessels on retinal images.

V. CONCLUSION

Based on the results and discussion it can be concluded that the VG-DropDNet architecture provides excellent performance results for blood vessels segmentation in retinal image, with the accuracy is more than 90%. VG-DropDNet can be the recommended architecture for blood vessel segmentation. The results of the VG-DropDNet architecture performance on the DRIVE dataset are great even though the IoU value obtained is still below 70%, but IoU in other studies are still rarely measured. The comparison of studies results in Table 3 shows that the IoU on DRIVE is the highest result. On the STARE dataset, the results of the VG-DropDNet architecture performance give excellent results. All performance results on the STARE dataset are above 86%. The Cohen's Kappa coefficient obtained by the VG-DropDNet architecture on both the DRIVE and STARE datasets has given great results above 0.8. It explains that the results of blood vessel segmentation on retina images generated by the VG-DropDNet architecture has a strong intensity of agreement with the ground truth images that have been provided by experts in the DRIVE and STARE datasets. The VG-DropDNet architecture is also able to provide a good balance for the segmentation of blood vessels in retinal images seen from the MCC and G-mean values close to 1. The segmentation image generated by VG-DropDNet is accurate and valid based on the performance results of accuracy, sensitivity, specificity, IoU, MCC, G-mean and F1-score. The segmented images from Vg-DropDNet are binary images containing only retinal blood vessels. These images can be used for the classification of retina disorders detection that requires examination of abnormalities in the retinal blood vessels. Image results from Vg-DropDNet can be used in medical applications to help detect retina disorders automatically. For future works, there are two focuses. The first is on improving and merging the VG-DropDNet architecture with other deep learning architectures and different stages of pre-processing data to improve performance results that are not optimal yet. The second is implementing the segmentation image from VG-DropDNet on the classification of retina disorders based on blood vessels retinal disorders so that this research can be applied in real terms in the medical field. Despite the success of VG-DropDNet in separating lesions from blood vessels, there are cases where this architecture does not perform well. This is especially true for retinal images with severe diabetic retinopathy with very large lesions or high vascular tortuosity. To overcome the limitations of VG-DropDNet, the use of morphological operating parameters can be used instead of predefined transformations to separate different components of the retinal image at the post-processing stage. By studying the morphological operations of the image, complex components of retinal images can be captured well.

REFERENCES

[1] A. Bali and S. N. Singh, ‘‘A review on the strategies and techniques of image segmentation,’’ in *Proc. 5th Int. Conf. Adv. Comput. Commun. Technol.*, Feb. 2015, pp. 113–120, doi: 10.1109/ACCT.2015.63.

[2] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, ‘‘Drinet for medical image segmentation,’’ *IEEE Trans. Med. Imag.*, vol. 37, no. 11, pp. 2453–2462, Nov. 2018, doi: 10.1109/TMI.2018.2835303.

[3] D. D. Patil, S. G. Deore, and S. Bhusawal, ‘‘Medical image segmentation: A review,’’ *Int. J. Comput. Sci. Mobile Comput.*, vol. 2, no. 1, pp. 22–27, 2013.

[4] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. De Vries, M. J. N. L. Benders, and I. Išgum, ‘‘Automatic segmentation of MR brain images with a convolutional neural network,’’ *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1252–1261, May 2016, doi: 10.1109/TMI.2016.2548501.

[5] M. Han, Y. Bao, Z. Sun, S. Wen, L. Xia, J. Zhao, J. Du, and Z. Yan, ‘‘Automatic segmentation of human placenta images with U-Net,’’ *IEEE Access*, vol. 7, pp. 180083–180092, 2019, doi: 10.1109/ACCESS.2019.2958133.

[6] E. Imani, M. Javidi, and H.-R. Pourreza, ‘‘Improvement of retinal blood vessel detection using morphological component analysis,’’ *Comput Methods Programs Biomed.*, vol. 118, no. 3, pp. 263–279, Mar. 2015, doi: 10.1016/j.cmpb.2015.01.004.

[7] D. R. Sarvamangala and R. V. Kulkarni, ‘‘Convolutional neural networks in medical image understanding: A survey,’’ *Evol. Intell.*, vol. 15, no. 1, Mar. 2021, Art. no. 0123456789, doi: 10.1007/s12065-020-00540-3.

[8] V. Popat, M. Mahdinejad, O. Cedeño, E. Naredo, and C. Ryan, ‘‘GA-based U-Net architecture optimization applied to retina blood vessel segmentation,’’ in *Proc. 12th Int. Joint Conf. Comput. Intell.*, 2020, pp. 192–199, doi: 10.5220/0010112201920199.

[9] O. Ronneberger, P. Fischer, and T. Brox, ‘‘U-Net: Convolutional networks for biomedical image segmentation,’’ in *Proc. Med. Image Comput. Comput.-Assisted Intervent*, vol. 9351, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[10] M. A. Al-masni and D.-H. Kim, ‘‘CMM-Net: Contextual multi-scale multi-level network for efficient biomedical image segmentation,’’ *Sci. Rep.*, vol. 11, p. 10191, Dec. 2021, doi: 10.1038/s41598-021-89686-3.

[11] T. Laibacher, T. Weyde, and S. Jalali, ‘‘M2U-Net: Effective and efficient retinal vessel segmentation for real-world applications,’’ in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 115–124, doi: 10.1109/CVPRW.2019.00020.

[12] Z. Zhang, C. Wu, S. Coleman, and D. Kerr, ‘‘DENSE-INception U-Net for medical image segmentation,’’ *Comput. Methods Programs Biomed.*, vol. 192, Aug. 2020, Art. no. 105395, doi: 10.1016/j.cmpb.2020.105395.

[13] S. S. Mousavi, M. Schukat, and E. Howley, ‘‘Deep reinforcement learning: An overview,’’ in *Proc. SAI Intell. Syst. Conf.*, in Lecture Notes in Networks and Systems, vol. 16. London, U.K., 2018, pp. 426–440, doi: 10.1007/978-3-319-56991-8_32.

[14] K. Simonyan and A. Zisserman, ‘‘Very deep convolutional networks for large-scale image recognition,’’ 2015, *arXiv:1409.1556*.

[15] K. Vignesh, G. Yadav, and A. Sethi, ‘‘Abnormal event detection on BMTT-PETS 2017 surveillance challenge,’’ in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 2161–2168, doi: 10.1109/CVPRW.2017.268.

[16] T. Kaur and T. K. Gandhi, ‘‘Automated brain image classification based on VGG-16 and transfer learning,’’ in *Proc. Int. Conf. Inf. Technol. (ICIT)*, 2019, pp. 94–98, doi: 10.1109/ICIT48102.2019.00023.

[17] Z. Khan, F. G. Khan, A. Khan, Z. U. Rehman, S. Shah, S. Qummar, F. Ali, and S. Pack, ‘‘Diabetic retinopathy detection using VGG-NIN a deep learning architecture,’’ *IEEE Access*, vol. 9, pp. 61408–61416, 2021, doi: 10.1109/ACCESS.2021.3074422.

[18] S.-H. Wang, M. A. Khan, and Y.-D. Zhang, ‘‘VISPNN: VGG-inspired stochastic pooling neural network,’’ *Comput., Mater. Continua*, vol. 70, no. 2, pp. 3081–3097, 2022, doi: 10.32604/cmc.2022.019447.

[19] M. Mateen, J. Wen, Nasrullah, S. Song, and Z. Huang, ‘‘Fundus image classification using VGG-19 architecture with PCA and SVD,’’ *Symmetry*, vol. 11, no. 1, p. 1, Dec. 2019, doi: 10.3390/sym11010001.

[20] C. Wang, Z. Zhao, Q. Ren, Y. Xu, and Y. Yu, ‘‘Dense U-Net based on patch-based learning for retinal vessel segmentation,’’ *Entropy*, vol. 21, no. 2, pp. 1–15, 2019, doi: 10.3390/e21020168.

[21] Q. Wang, Q. Liu, G. Luo, Z. Liu, J. Huang, Y. Zhou, Y. Zhou, W. Xu, and J.-Z. Cheng, ‘‘Automated segmentation and diagnosis of pneumothorax on chest X-rays with fully convolutional multi-scale ScSE-DenseNet: A retrospective study,’’ *BMC Med. Informat. Decis. Making*, vol. 20, no. S14, pp. 1–13, Dec. 2020, doi: 10.1186/s12911-020-01325-5.

[22] Y. Cheng, M. Ma, L. Zhang, C. J. Jin, L. Ma, and Y. Zhou, ‘‘Retinal blood vessel segmentation based on densely connected U-Net,’’ *Math. Biosci. Eng.*, vol. 17, no. 4, pp. 3088–3108, 2020, doi: 10.3934/MBE.2020175.

[23] P. Chen, L. Gao, X. Shi, K. Allen, and L. Yang, ‘‘Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss,’’ *Computerized Med. Imag. Graph.*, vol. 75, pp. 84–92, Jul. 2019, doi: 10.1016/j.compmedimag.2019.06.002.

[24] S. Singaravel, J. Suykens, and P. Geyer, ‘‘Deep-learning neural-network architectures and methods: Using component-based models in building-design energy prediction,’’ *Adv. Eng. Inform.*, vol. 38, pp. 81–90, Oct. 2018.

[25] H. Salehinejad and S. Valaee, ‘‘Ising-dropout: A regularization method for training and compression of deep neural networks,’’ in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 3602–3606, doi: 10.1109/ICASSP.2019.8682914.

[26] S. Lee and C. Lee, ‘‘Revisiting spatial dropout for regularizing convolutional neural networks,’’ *Multimedia Tools Appl.*, vol. 79, nos. 45–46, pp. 34195–34207, Dec. 2020, doi: 10.1007/s11042-020-09054-7.

[27] J. M. Karnuta, S. M. Navarro, H. S. Haeberle, J. M. Helm, A. F. Kamath, J. L. Schaffer, V. E. Krebs, and P. N. Ramkumar, ‘‘Predicting inpatient payments prior to lower extremity arthroplasty using deep learning: Which model architecture is best?’’ *J. Arthroplasty*, vol. 34, no. 10, pp. 2235–2241, 2019, doi: 10.1016/j.arth.2019.05.048.

[28] T. D. Bui, J. Shin, and T. Moon, ‘‘Skip-connected 3D DenseNet for volumetric infant brain MRI segmentation,’’ *Biomed. Signal Process. Control*, vol. 54, Sep. 2019, Art. no. 101613.

[29] Y. Jiang, N. Tan, T. Peng, and H. Zhang, ‘‘Retinal vessels segmentation based on dilated multi-scale convolutional neural network,’’ *IEEE Access*, vol. 7, pp. 76342–76352, 2019.

[30] X. Tang, Z. Li, and Y. Chen, ‘‘A night image enhancement algorithm based on guided filtering,’’ in *Advanced Graphic Communications and Media Technologies* (Lecture Notes in Electrical Engineering), vol. 417. New York, NY, USA: Springer-Verlag, 2017, pp. 283–288.

[31] R. C. Gonzales and R. E. Woods, *Digital Image Processing*, 3rd ed. Hoboken, NJ, USA: Pearson-Hall, 2008.

[32] R. S. C. Boss, K. Thangavel, and D. A. P. Daniel, ‘‘Automatic mammogram image breast region extraction and removal of pectoral muscle,’’ 2013, *arXiv:1307.7474*.

[33] G. Yadav, S. Maheshwari, and A. Agarwal, ‘‘Contrast limited adaptive histogram equalization based enhancement for real time video system,’’ in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2014, pp. 2392–2397, doi: 10.1109/ICACCI.2014.6968381.

[34] R. Kushol, M. H. Kabir, M. S. Salekin, and A. B. M. A. Rahman, ‘‘Contrast enhancement by top-hat and bottom-hat transform with optimal structuring element: Application to retinal vessel segmentation,’’ in *Image Analysis and Recognition* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2017, pp. 533–540.

[35] S. Wang, Y. Yin, G. Cao, B. Wei, Y. Zheng, and G. Yang, ‘‘Hierarchical retinal blood vessel segmentation based on feature and ensemble learning,’’ *Neurocomputing*, vol. 149, pp. 708–717, Feb. 2015, doi: https://doi.org/10.1016/j.neucom.2014.07.059.

[36] J. Shi, J. Dang, M. Cui, R. Zuo, K. Shimizu, A. Tsunoda, and Y. Suzuki, ‘‘Improvement of damage segmentation based on pixel-level data balance using VGG-UNet,’’ *Appl. Sci.*, vol. 11, no. 2, pp. 1–17, 2021, doi: 10.3390/app11020518.

[37] S. Albawi, T. A. M. Mohammed, and S. Alzawi, ‘‘Layers of a convolutional neural network,’’ in *Proc. Int. Conf. Eng. Technol. (ICET)*, 2017, pp. 1–6.

[38] A. Desiani, B. Suprihatin, S. Yahdin, A. I. Putri, and F. R. Husein, ‘‘Bi-path architecture of CNN segmentation and classification method for cervical cancer disorders based on Pap-smear images,’’ *IAENG Int. J. Comput. Sci.*, vol. 48, no. 3, pp. 782–791, 2021. [Online]. Available: http://www.iaeng.org/IJCS/issues_v48/issue_3/IJCS_48_3_37.pdf

[39] A. Desiani *et al.*, ‘‘R-peak detection of beat segmentation and convolution neural network for arrhythmia classification,’’ *J. Eng. Sci. Technol.*, vol. 17, no. 2, pp. 1231–1246, 2022.

[40] N. M. Aboelenein, P. Songhao, A. Koubaa, A. Noor, and A. Afifi, "HTTU-Net: Hybrid two track U-Net for automatic brain tumor segmentation," *IEEE Access*, vol. 8, pp. 101406–101415, 2020, doi: 10.1109/ACCESS.2020.2998601.

[41] X. Sun, L. Liu, C. Li, J. Yin, J. Zhao, and W. Si, "Classification for remote sensing data with improved CNN-SVM method," *IEEE Access*, vol. 7, pp. 164507–164516, 2019, doi: 10.1109/ACCESS.2019.2952946.

[42] J. Shi, J. Dang, M. Cui, R. Zuo, K. Shimizu, A. Tsunoda, and Y. Suzuki, "Improvement of damage segmentation based on pixel-level data balance using VGG-Unet," *Appl. Sci.*, vol. 11, no. 2, pp. 1–17, 2021, doi: 10.3390/app11020518.

[43] R. Zhang, Z. Zhou, W. Wu, C.-C. Lin, P.-H. Tsui, and S. Wu, "An improved fuzzy connectedness method for automatic three-dimensional liver vessel segmentation in CT images," *J. Healthcare Eng.*, vol. 2018, pp. 1–18, Oct. 2018, doi: 10.1155/2018/2376317.

[44] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Aug. 2017, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.

[45] S.-H. Wang, C. Tang, J. Sun, J. Yang, C. Huang, P. Phillips, and Y. D. Zhang, "Multiple sclerosis identification by 14-layer convolutional neural network with batch normalization, dropout, and stochastic pooling," *Frontiers Neurosci.*, vol. 12, p. 818, Nov. 2018, doi: 10.3389/fnins.2018.00818.

[46] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 448–456.

[47] L. Pauly, H. Peel, S. Luo, D. Hogg, and R. Fuentes, "Deeper networks for pavement crack detection," in *Proc. Int. Symp. Autom. Robot. Construction (IAARC)*, Jul. 2017, pp. 479–485, doi: 10.22260/ISARC2017/0066.

[48] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, "Learning pooling for convolutional neural network," *Neurocomputing*, vol. 224, pp. 96–104, Feb. 2017.

[49] T. A. Soomro, A. J. Afifi, L. Zheng, S. Soomro, J. Gao, O. Hellwich, and M. Paul, "Deep learning models for retinal blood vessels segmentation: A review," *IEEE Access*, vol. 7, pp. 71696–71717, 2019, doi: 10.1109/ACCESS.2019.2920616.

[50] D. Im, D. Han, S. Choi, S. Kang, and H.-J. Yoo, "DT-CNN: An energy-efficient dilated and transposed convolutional neural network processor for region of interest based image segmentation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 10, pp. 3471–3483, Oct. 2020, doi: 10.1109/TCSI.2020.2991189.

[51] Y. Zhang, X. Zhao, and P. Liu, "Multi-point displacement monitoring based on full convolutional neural network and smartphone," *IEEE Access*, vol. 7, pp. 139628–139634, 2019, doi: 10.1109/ACCESS.2019.2943599.

[52] S. Jadon, "A survey of loss functions for semantic segmentation," in *Proc. IEEE Conf. Comput. Intell. Bioinf. Comput. Biol., (CIBCB)*, 2020, pp. 1–7, doi: 10.1109/CIBCB48159.2020.9277638.

[53] A. Atapour-Abarghouei and T. P. Breckon, "Monocular segment-wise depth: Monocular depth estimation based on a semantic segmentation prior," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 4295–4299, doi: 10.1109/ICIP.2019.8803551.

[54] J. Pérez, J. Díaz, J. Garcia-Martin, and B. Tabuenca, "Systematic literature reviews in software engineering—Enhancement of the study selection process using Cohen's Kappa statistic," *J. Syst. Softw.*, vol. 168, Oct. 2020, Art. no. 110657, doi: 10.1016/j.jss.2020.110657.

[55] Y. Xu, Y. Wang, J. Yuan, Q. Cheng, X. Wang, and P. L. Carson, "Medical breast ultrasound image segmentation by machine learning," *Ultrasonics*, vol. 91, pp. 1–9, Jan. 2019, doi: 10.1016/j.ultras.2018.07.006.

[56] Z. Lin, J. Huang, Y. Chen, X. Zhang, W. Zhao, Y. Li, L. Lu, M. Zhan, X. Jiang, and X. Liang, "A high resolution representation network with multi-path scale for retinal vessel segmentation," *Comput. Methods Programs Biomed.*, vol. 208, Sep. 2021, Art. no. 106206, doi: 10.1016/j.cmpb.2021.106206.

[57] S. Moccia, E. De Momi, S. El Hadji, and L. S. Mattos, "Blood vessel segmentation algorithms—Review of methods, datasets and evaluation metrics," *Comput. Methods Programs Biomed.*, vol. 158, pp. 71–91, May 2018, doi: 10.1016/j.cmpb.2018.02.001.

[58] C. Guo, M. Szemenyei, Y. Pei, Y. Yi, and W. Zhou, "SD-Unet: A structured dropout U-Net for retinal vessel segmentation," in *Proc. IEEE 19th Int. Conf. Bioinf. Bioeng. (BIBE)*, Oct. 2019, pp. 439–444, doi: 10.1109/BIBE.2019.00085.

[59] K.-B. Park, S. H. Choi, and J. Y. Lee, "M-GAN: Retinal blood vessel segmentation by balancing losses through stacked deep fully convolutional networks," *IEEE Access*, vol. 8, pp. 146308–146322, 2020, doi: 10.1109/ACCESS.2020.3015108.

[60] J. Zhang, Y. Zhang, and X. Xu, "Pyramid U-Net for retinal vessel segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 1125–1129.

[61] A. Atapour-Abarghouei and T. P. Breckon, "Monocular segment-wise depth: Monocular depth estimation based on a semantic segmentation prior," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 4295–4299, doi: 10.1109/ICIP.2019.8803551.

[62] M. Z. Alom, C. Yakopcic, M. Hasan, T. M. Taha, and V. K. Asari, "Recurrent residual U-Net for medical image segmentation," *J. Med. Imag.*, vol. 6, no. 1, Mar. 2019, Art. no. 014006, doi: 10.1117/1.JMI.6.1.014006.

[63] P. Yin, R. Yuan, Y. Cheng, and Q. Wu, "Deep guidance network for biomedical image segmentation," *IEEE Access*, vol. 8, pp. 116106–116116, 2020, doi: 10.1109/ACCESS.2020.3002835.

[64] Y. Zhang, M. He, Z. Chen, K. Hu, X. Li, and X. Gao, "Bridge-Net: Context-involved U-Net with patch-based loss weight mapping for retinal blood vessel segmentation," *Exp. Syst. Appl.*, vol. 195, Jun. 2022, Art. no. 116526, doi: 10.1016/j.eswa.2022.116526.

[65] T. Mostafiz, I. Jarin, S. A. Fattah, and C. Shahnaz, "Retinal blood vessel segmentation using residual block incorporated U-Net architecture and fuzzy inference system," in *Proc. IEEE Int. WIE Conf. Electr. Comput. Eng. (WIECON-ECE)*, Dec. 2018, pp. 106–109, doi: 10.1109/WIECON-ECE.2018.8783182.

[66] D. Li, D. A. Dharmawan, B. P. Ng, and S. Rahardja, "Residual U-Net for retinal vessel segmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1425–1429, doi: 10.1109/ICIP.2019.8803101.

[67] L. Li, M. Verma, Y. Nakashima, H. Nagahara, and R. Kawasaki, "IterNet: Retinal image segmentation utilizing structural redundancy in vessel networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis., (WACV)*, Mar. 2020, pp. 3645–3654, doi: 10.1109/WACV45572.2020.9093621.

[68] D. Li and S. Rahardja, "BSEResU-Net: An attention-based before-activation residual U-Net for retinal vessel segmentation," *Comput. Methods Programs Biomed.*, vol. 205, Jun. 2021, Art. no. 106070, doi: 10.1016/j.cmpb.2021.106070.

[69] D. Chen, W. Yang, L. Wang, S. Tan, J. Lin, and W. Bu, "PCAT-UNet: UNet-like network fused convolution and transformer for retinal vessel segmentation," *PLoS ONE*, vol. 17, no. 1, pp. 1–22, 2022, doi: 10.1371/journal.pone.0262689.

[70] R. Dhanagopal, P. T. V. Raj, R. S. Kumar, R. M. Das, K. Pradeep, and O.-A. Kwadwo, "An efficient retinal segmentation-based deep learning framework for disease prediction," *Wirel. Commun. Mob. Comput.*, vol. 2022, Jun. 2022, Art. no. 2013558, doi: 10.1155/2022/2013558.

[71] J. I. Orlando, E. Prokofyeva, and M. B. Blaschko, "A discriminatively trained fully connected conditional random field model for blood vessel segmentation in fundus images," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 1, pp. 16–27, Jan. 2017, doi: 10.1109/TBME.2016.2535311.

[72] L. Zhou, Q. Yu, and X. Xu, "Improving dense conditional random field for retinal vessel segmentation by discriminative feature learning and thin-vessel enhancement," *Comput. Methods Programs Biomed.*, vol. 148, pp. 13–25, Apr. 2017, doi: 10.1016/j.cmpb.2017.06.016.

[73] S. Guo, K. Wang, H. Kang, Y. Zhang, Y. Gao, and T. Li, "BTS-DSN: Deeply supervised neural network with short connections for retinal vessel segmentation," *Int. J. Med. Informat.*, vol. 126, pp. 105–113, Jun. 2019.

[74] Z. Zhuo, J. Huang, K. Lu, D. Pan, and S. Feng, "A size-invariant convolutional network with dense connectivity applied to retinal vessel segmentation measured by a unique index," *Comput. Methods Programs Biomed.*, vol. 196, Nov. 2020, Art. no. 105508, doi: 10.1016/j.cmpb.2020.105508.

[75] S. Nagdeote and S. Prabhu, "Hybrid UNet architecture based on residual learning of fundus images for retinal vessel segmentation," *J. Phys., Conf.*, vol. 2070, no. 1, Nov. 2021, Art. no. 012104, doi: 10.1088/1742-6596/2070/1/012104.

[76] K.-B. Park, S. H. Choi, and J. Y. Lee, "M-GAN: Retinal blood vessel segmentation by balancing losses through stacked deep fully convolutional networks," *IEEE Access*, vol. 8, pp. 146308–146322, 2020, doi: 10.1109/ACCESS.2020.3015108.

[77] J. Huang, Z. Lin, Y. Chen, X. Zhang, W. Zhao, J. Zhang, Y. Li, X. He, M. Zhan, L. Lu, X. Jiang, and Y. Peng, "DBFU-Net: Double branch fusion U-Net with hard example weighting train strategy to segment retinal vessel," *PeerJ Comput. Sci.*, vol. 8, p. e871, Feb. 2022, doi: 10.7717/peerj-cs.871.

ANITA DESIANI was born in Palembang, Indonesia, in 1977. She received the bachelor's degree in mathematics from Universitas Sriwijaya, in 2000, and the Magister degree in computer science from Universitas Gadjah Mada, in 2003. She is currently working on a project for her Doctoral Program with the Mathematics and Natural Science Faculty, Universitas Sriwijaya. Since 2003, she has been a Lecturer with the Mathematics Department, Universitas Sriwijaya. She is also the author of *Artificial Intelligence* and *Matlab Programming* books. Her current research interests include data mining, image processing, pattern recognition and computer vision, and artificial intelligent.

ERWIN (Member, IEEE) was born in Palembang, Indonesia, in 1971. He received the bachelor's degree in mathematics from Universitas Sriwijaya, Indonesia, in 1994, the M.Sc. degree in actuarial from the Bandung Institute of Technology (ITB), Bandung, Indonesia, in 2002, and the Ph.D. degree in engineering with the Faculty of Engineering, Universitas Sriwijaya, in 2019. In 1994, he joined as a Lecturer with Universitas Sriwijaya, where he was an Associate Professor, in 2011. Since December 2006, he has been with the Department of Informatics, Universitas Sriwijaya. Since 2012, he has been with the Department of Computer Engineering, Universitas Sriwijaya. His current research interests include image processing and computer vision. He is a member of IAENG.

BAMBANG SUPRIHATIN was born in Salatiga, Indonesia, in 1971. He received the bachelor's degree in mathematics from Universitas Sriwijaya, Indonesia, in 1994, the M.Sc. degree in mathematics from the Bandung Institute of Technology (ITB), Bandung, Indonesia, in 2002, and the Ph.D. degree in mathematics with Universitas Gadjah Mada (UGM), in 2016. In 1994, he joined as a Lecturer with Universitas Sriwijaya. He was an Associate Professor, in 2011. He has experience as the author of *Statistics* and *Mathematics* books. His current research interests include statistics and modeling.

FILDA EFRILIYANTI was born in Palembang, in April 1999. She received the bachelor's degree in mathematics from Universitas Sriwijaya, in 2021. In 2018, she joined the Computer Laboratory, Faculty of Mathematics and Natural Sciences, Sriwijaya University, as an Assistant Lecturer. Her current research interests include image processing, pattern recognition and computer vision, data mining, and artificial intelligence.

MUHAMMAD ARHAMI was born in Pulo Leung Teuga, in October 1974. He received the bachelor's degree in mathematics from Universitas Syiah Kuala, in 2000, and the Magister degree in computer science from Universitas Gadjah Mada, in 2004. He was an Associate Professor, in 2013. He has been with Politkenik Negeri Lhokseumawe, since 2001. He already has experience as the author of *Artificial Intelligence*, *Expert System*, *Matlab Programming*, and *Mathematics* books. His research interests include artificial intelligent, mathematics, data mining, software engineering, and data structure.

EMY SETYANINGSIH was born in Semarang, Indonesia, in 1972. She received the bachelor's degree in computer science from IST AKPRIND, Indonesia, in 1996, and the M.Com. and Ph.D. degrees in computer science from Universitas Gadjah Mada (UGM), Yogyakarta, Indonesia, in 2004 and 2019, respectively. In 1997, she joined as a Lecturer with IST AKPRIND. She was an Associate Professor, in 2009. Her current research interests include cryptography, digital image processing, and pattern recognition.