Received: 16 November 2021 | Revised: 14 February 2022 | Accepted: 11 June 2022

IET Image Processing

DOI: 10.1049/ipr2.12580

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

IterNet++: An improved model for retinal image segmentation by curvelet enhancing, guided filtering, offline hard-sample mining, and test-time augmenting

M. Zhu[1] | K. Zeng[1] | G. Lin[1] | Y. Gong[2] | T. Hao[3] | K. Wattanachote[2] | X. Luo[4]

[1]School of Computer Science and Engineering, Sun Yat-Sen University, Guangzhou, China

[2]Guangdong University of Foreign Studies, Guangzhou, China

[3]School of Computer Science, South China Normal University, Guangzhou, China

[4]National & Local Joint Engineering Research Center of Satellite Navigation and Location Service, Guilin University of Electronic Technology, Guilin, China

Correspondence
K. Zeng, School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China.
Email: zengkun2@mail.sysu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: U1711266; Guangzhou Scientific and Technological Plan, Grant/Award Number: 201904010228.; Guangdong Basic and Applied Basic Research Foundation, Grant/Award Number: 2019A1515011078

Abstract

In clinical medicine, the segmentation of blood vessels in retinal images is essential for subsequent analysis in clinical diagnosis. However, retinal images are often noisy and their vascular structure is relatively tiny, which poses significant challenges for vessel segmentation. To improve the performance of vessel segmentation, an improved model IterNet++ based on the architecture of IterNet is proposed. First, curvelet signal analysis is applied to enhance retinal images. Second, residual convolution (ResConv) blocks and guided filters are introduced to utilise the encoder features of previous iterations in the model to reduce overfitting. Third, offline hard-sample mining is used to improve segmentation performance by utilising training samples with low segmentation accuracy as many possible on a few-sample training set. In addition, a test-time augmentation method is applied to testing samples in test dataset during inference. Extensive experiments show that this model achieves Dice scores of 0.8313, 0.8277, and 0.8372 on DRIVE, CHASE-DB1, and STARE datasets, respectively, demonstrating the best performance compared with IterNet and other baseline models.

1 | INTRODUCTION

In clinical medicine, retinal images of human eyes are used as an important basis for clinical diagnosis. Generally speaking, a retinal image includes the retinal microcirculatory system, macula, optic disc, central recess, microaneurysms, exudate etc. Compared with angiography and coherent tomography (OCT) [1–3], retinal images have more advantages for clinical analysis in the diagnosis of diabetic retinopathy, retinal occlusion etc. [4–6] There are many medical and anatomical analysis of retinal images which have been performed to help understand the structure of retina [2, 7]. This suggests that the understanding and analysis of retinal image information are essential for subsequent research, such as the analysis and diagnosis of retinal-related conditions. The segmentation of blood vessels in digital retinal images plays a significant role in further analysis of clinical medicine.

In recent years, methods such as digital image processing and computer vision have emerged to perform vessel segmentation on digital retinal images. Especially, U-Net [8] and deep learning models based on U-Net are applied widely. Among them, there are ET-Net [9], which uses edge information to reinforce supervision in segmentation, and IterNet [10], which enriches features by using iterative U-Net structures to assist segmentation. IterNet is one of the best among these methods, and it introduces multiple segmentations to achieve refinement of segmentation results by the iterations of UNet_Outs. However, it uses arbitrary angle rotation as the data enhancement strategy, which brings ambiguous ground-truth labels near the edges of vessels. Moreover, it fails to pay special attention to the hard

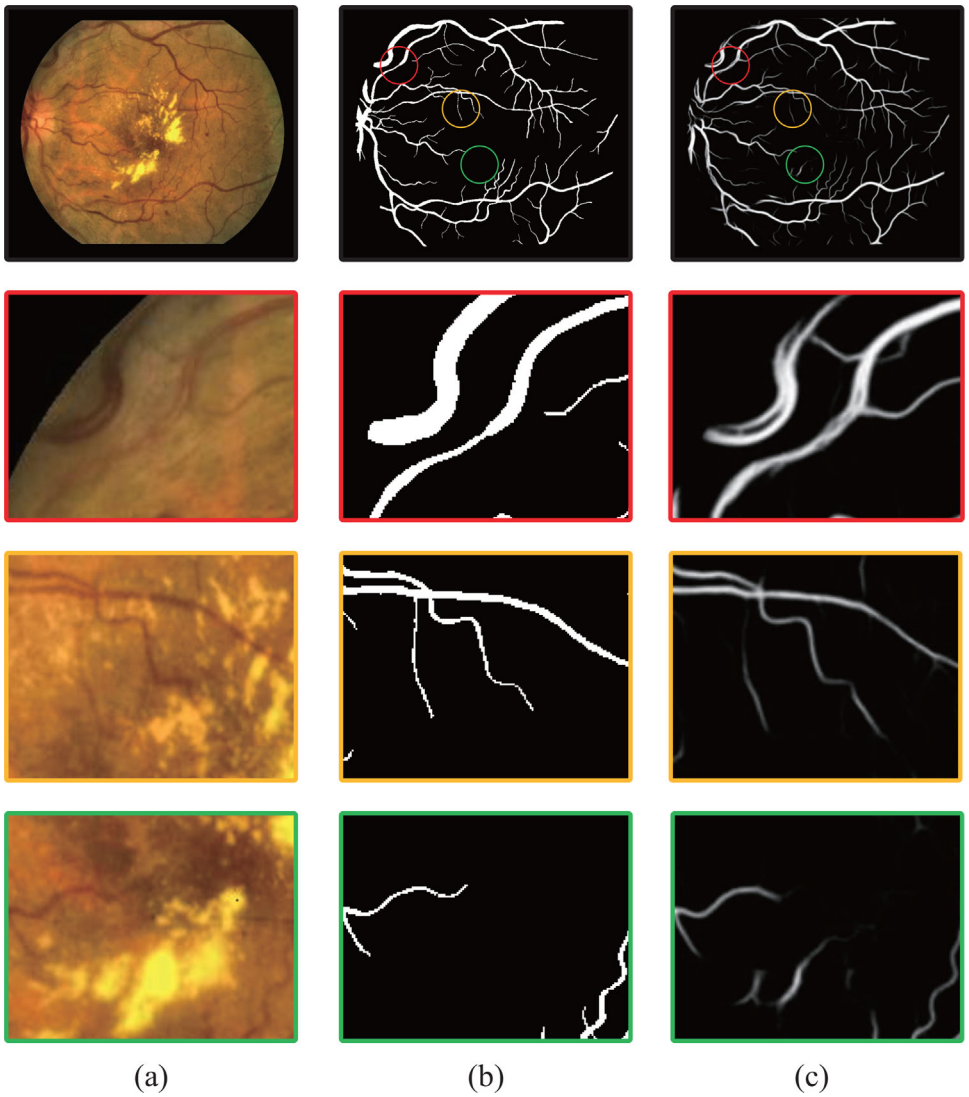

FIGURE 1 A retinal image in STARE dataset and its segmentation result by a state-of-the-art model IterNet [10]. (a) A retinal image in STARE dataset, (b) the corresponding annotation of the image, (c) the pixelwise segmentation probability of the image by IterNet

regions of input images. These drawbacks lower its segmentation performance dramatically on the edges of blood vessels and the hard regions of images.

Therefore, vessel segmentation for retinal images is still a very challenging research problem. Firstly, unlike natural images, medical image including retinal images are generally difficult to obtain a large number of samples. Secondly, retinal images usually have a large amount of noise during acquisition, causing confusions with the subtle structure of tiny vessels. Moreover, blood vessels rarely extend in a single orientation, while the orientation may be constantly changing. These problems bring different degrees of difficulties to segmentation and thus the models frequently produce unsatisfactory performances. Figure 1 shows the results of vessel segmentation of retinal images by IterNet [10] on STARE dataset [11] with problems illustrated below.

- **Segmentation voids:** The red circle in Figure 1 shows a region of brighter vessels in the image. In fact, the whole region should be segmented as vessels. However, these vessels are dark on the sides and bright in the middle, which brings noise to the segmentation. IterNet fails to segment all these regions as vessels, and there are voids in the segmentation.
- **Tiny blood vessels:** The orange circle in Figure 1 shows some tiny blood vessels. Although IterNet is able to segment a part of this vessel, the detailed features are not captured sufficiently. Therefore, IterNet generates incontinuous segmentation results of weak intensity on tiny blood vessels.
- **False positive regions:** The green circle in Figure 1 shows a false positive region of the image, which does not contain any blood vessels. However, some of their textures have

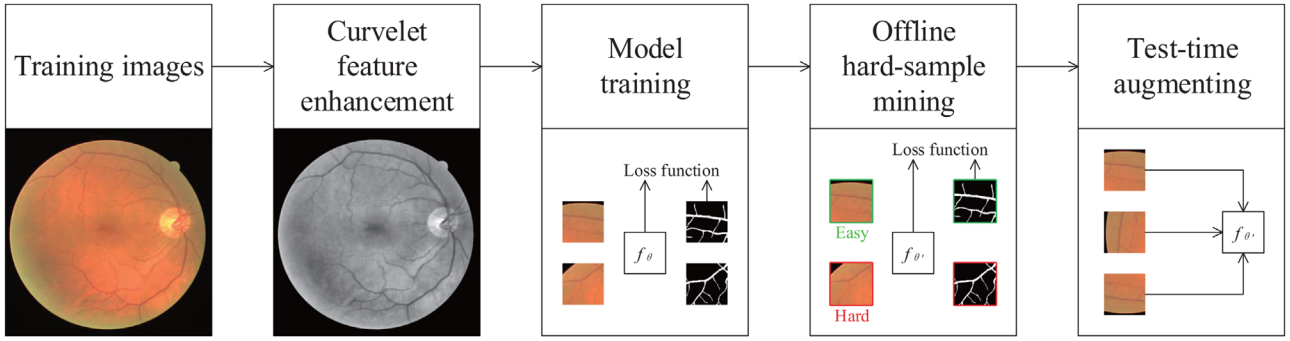

FIGURE 2 The overview procedure of IterNet++ model for vessel segmentation of retinal images

similarities with those of blood vessels. This may mislead models like IterNet for containing false positive predictions.

To alleviate these problems, we proposed a new model, named IterNet++, for segmenting vessels of retinal images. As shown in Figure 2, the model first performs denoising and feature enhancing of images by curvelet signal analysis, and then carries out initial training with these features. After that, offline hard-sample mining on the trained IterNet++ model is leveraged to improve model performance. In testing process, a test-time augmentation (TTA) method is applied to enhance input data, which allows the model to produce more robust predictions with the enhanced data.

The main contributions of this work are as follows:

- An innovate model IterNet++ is proposed by adding guided filters to UNet_Outs to leverage essential features and offline hard sample mining to enhance training data.
- Fast curvelet transform and filtering, as well as the TTA method, are incorporated to fully exploit the latent information in retinal datasets of small sample sizes.
- Extensive experiments and statistical analysis show that our model outperforms existing baseline models on three standard datasets.

The rest of the paper is organised as follows: In Section 2, related work is investigated and presented. In Section 3, the proposed method IterNet++ is introduced. In Section 4, experiments, results and discussion are illustrated, while Section 5 concluded the paper.

2 | RELATED WORK

2.1 | Image enhancement based on wavelet analysis

There has been a considerable amount of work on digital image enhancement through signal analysis. Wavelet analysis [12–14] used finite-length decaying wavelet bases to transform nonstationary signals to obtain a time-frequency spectrum. Moreover, stationary wavelet transform (SWT) [15] replaced downsampling operations in the discrete wavelet transform with upsampling operations and thus the transformed signals were translation invariant. Oliveira et al. [15] applied SWT to enhance retinal images and the segmentation of blood vessels was performed by an FCN of U-Net shape. However, SWT did not perform well with vessels that extended in different orientations in the images, and such vessels were one of the most challenging tasks in segmentation. Therefore, curvelet [16, 17] was proposed for expressing complex image signals such as retinal images containing vessels of various shapes. FDCT was applied to input images to generate its frequent domain representation. Hence, it could filter images in frequent domains [18], and later changed filtered signals back to spatial domain by an inverse FDCT. Chalakkal and Abdulla [19] used curvelet transform for enhancement of retinal images and combined with line operations for vessel segmentation. In previous studies, curvelet analysis was generally used directly in signal analysis of retinal images for unsupervised segmentation [19–22], instead of combining with deep learning models for supervised segmentation. Hence, IterNet++ is designed to benefit from the feature enhancement of curvelet analysis.

2.2 | Deep learning models for medical image segmentation

In recent years, deep learning models [23–26] have become popular among researchers because of higher performance achieved in medical image segmentation. These models are often based on neural networks and supervised segmentation, including FCN-based segmentation models, U-Net-based segmentation models, and deep learning models with auxiliary information, which are briefly described below.

2.2.1 | FCN-based segmentation models

In general, fully convolutional networks (FCNs) are applied to image segmentation tasks to extract semantic features from images by using several convolutional layers as well as pooling layers. Brancati et al. [27] applied some specified types of convolutional filters to the first layer of their FCN for small datasets

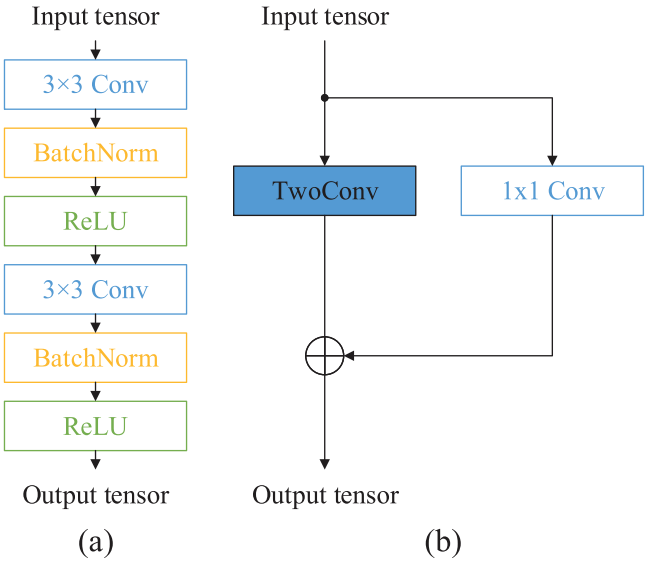

FIGURE 3 The module structure of TwoConv and ResConv, where (a) is the structure of TwoConv, (b) is the structure of ResConv, and ⊕ represents element-wise addition of tensors

of retinal images in the vessel segmentation task. Yan et al. [28] divided their FCN into three functional modules, namely thick segmenter, thin segmenter and fusion segmenter, to perform vessel segmentation from coarse to fine levels. Guo et al. [29] applied a reinforcement sample learning strategy to increase training speed and inference accuracy of their FCN.

2.2.2 | U-net-based segmentation models

However, methods based on FCNs often failed to achieve satisfy results in medical image segmentation on small size dataset and specialised domains. Therefore, U-Net [8] and its variants [24, 25] were more popular among researchers. They exploited multiple layers of decoders instead of a single fully connected layer for segmentation through an encoder-decoder architecture. Moreover, they applied skip connections for making more efficient use of shallow and deep features. As shown in Figure 3(a), a two-layer convolution-ReLU activation function-dropout structure, denoted as TwoConv, was used as the structure of each encoder and decoder layer. In addition, the downsampling and upsampling of extracted features were performed by MaxPool [30] and TransConv [31], respectively. A shallow decoder took the features from a shallow encoder and a deep decoder results as inputs to generate output, thus achieving efficient utilisation of similar features from small size datasets. There are many works based on the U-Net structure in various sub-problems of medical image segmentation. U-Net++ [32] was proposed to enrich feature extraction by adding more decoders and skip connections. V-Net [33] was proposed to enhance feature representation of encoders and decoders in 3D medical image segmentation problem. It stacked more modules on encoders and decoders and added a short-circuit mechanism. M-Net [34] was proposed to enrich multi-scale feature by inputting downsampled versions of input images and deep features into a deep encoder after concatenating them. Moreover, it utilised the output of the deep decoder for segmentation tasks at low resolution. DU-Net [35] replaced all the convolutions in TwoConv in U-Net with deformable convolutions, which enhanced the feature extraction capability of encoders and decoders and extracted more detailed vessels. IterNet [10] used a U-Net (UNet_In) as the first iteration for segmentation and iteratively used mini U-Nets (UNet_Outs) to extract extra features after UNet_In and to reduce overfitting by weight sharing in UNet_Outs. Nevertheless, there are still some ambiguous segmentation results generated by IterNet. To that end, IterNet++ tries to solve this issue by adjusting network architecture and feature extracting strategy.

2.2.3 | Deep learning models with auxiliary information

In general, U-net-based segmentation models tend to perform poorly in segmentation near edges and on tiny targets. Therefore, there are some researches to improve the performance of the existing models for these particular issues. Wang et al. [36] used superpixels to select representative samples and enforce local consistency. ET-Net [9] introduced edge information from ground-truth labels as Supporting Information and treated the edge prediction as an auxiliary task to guide the final segmentation prediction by edge features. Yang et al. [37] captured the topology characteristics by morphological bottom-hat transform to enhance the connectivity of vessels. However, the partial auxiliary information could not handle retinal images well, especially in the segmentation task of intricate vessels. Therefore, general auxiliary information, e.g. the spatial structure of retinal images, is motivated to be exploited in IterNet++.

3 | METHODOLOGY

The architecture of the proposed IterNet++ contains 4 modules: 1) curvelet feature enhancement; 2) a deep learning model framework, which is shown in Figure 5; 3) offline hard-sample mining designed for vessel segmentation of retinal images, which is shown in Figure 7; 4) test-time augmentation (TTA) using test data, which is shown in Figure 8. These modules are described in Sections 3.1, 3.2, 3.3, and 3.4 in detail respectively.

3.1 | Feature enhancement for input images based on curvelets

Unlike wavelet bases that divide frequent domain only by different scales, curvelet bases divide frequent domains by both different scales and orientations. Therefore, it allows to focus more on local detailed vascular structures, so that segmentations are less affected by local noises. For an input image $I(x, y)$, we define a number of curvelet bases $\phi_{\lambda,\theta,k_1,k_2}(x, y)$, where λ

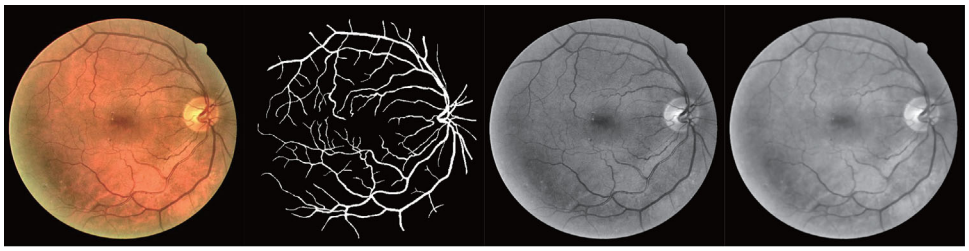

FIGURE 4 A retinal image in DRIVE dataset with corresponding annotation, greyscale image, and curvelet enhanced result. (a) A retinal image in DRIVE dataset, (b) the corresponding annotation of (a), (c) the corresponding greyscale image of (a), (d) the curvelet enhanced result of (a)

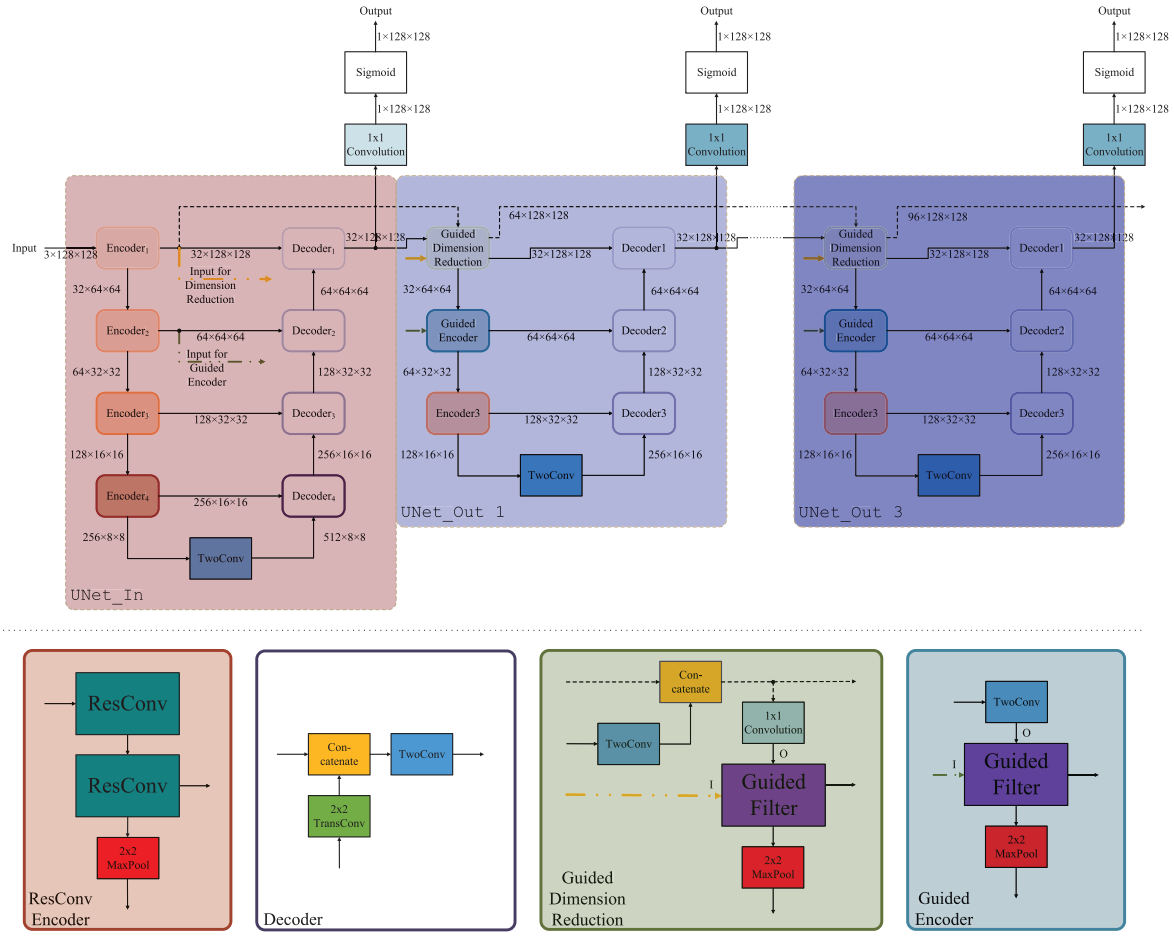

FIGURE 5 The network architecture of IterNet++ for blood vessel segmentation of retinal images. The architecture has 1 UNet_In and 3 UNet_Outs, where the encoders of UNet_In and the deepest encoders of UNet_Outs adopt ResConv encoders, and the decoders of UNet_In and UNet_Outs keep the same as those in IterNet. The guided filters are added in the shallowest-layer encoders of UNet_Outs, i.e. the dimension reduction modules and the second-layer encoders of UNet_Outs

denotes the base scale, θ denotes the orientation of the bases, and k_1 and k_2 denote the spatial location of the bases. The bases of two-dimensional Fourier transform are determined only by the frequency parameters u, v, and the bases of wavelet transform are determined only by the scale λ and spatial position parameters k_1, k_2. However, compared to the wavelet transform, the bases of curvelet transform also take the orientation θ into consideration. These bases after curvelet transform are more suitable for the analysis of retinal images due to the various orientations in space of vessels. The curvelet coefficients are calculated from the input image and these curvelet bases as Equation (1):

$$C(\lambda,\theta,k_1,k_2) = \sum_{0\le x<h,0\le y<w} I(x,y)\phi_{\lambda,\theta,k_1,k_2}(x,y), \quad (1)$$

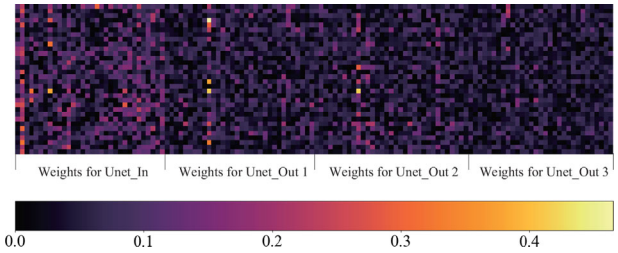

FIGURE 6 The visualisation of the absolute value of the weights of IterNet in the third UNet_Out iteration (UNet_Out 3) of the dimension-reduction module (1 × 1 convolution) after training on DRIVE dataset

where h and w denote the height and width of the input image I, respectively. Generally, the curvelet coefficients are computed by FDCT, which is shown in Algorithm 1.

By following the method described by [18–20], the curvelet coefficients $C(\lambda, \theta, k_1, k_2)$ are adjusted at each scale λ and orientation θ by a nonlinear function $\kappa(\cdot)$. We assume that the image consists of a target signal and a random noise in the frequent domain using Equation (2) and assume that the noise $n(x, y)$ independently and identically obeys a Gaussian distribution $N(0, \sigma^2)$.

$$I(x, y) = s(x, y) + n(x, y) \tag{2}$$

From the empirical results given by ref. [38], we can make a parametric estimation of the standard deviation σ of this Gaussian distribution as Equation (3):

$$\sigma = \sqrt{\frac{\pi}{2}} \frac{1}{6(h-2)(w-2)} \sum |I(x, y) * M|, \tag{3}$$

ALGORITHM 1 FDCT

Input: An image I with the height and width being h and w respectively.

Output: Curvelet coefficients $C(\lambda, \theta, k_1, k_2)$, in which λ represents scale, θ represents orientation, k_1 and k_2 represent spatial positions respectively.

1: Calculate Fourier coefficients F of image I
2: **for** $\lambda \in \{1, 2, \ldots, \lceil \log_2 \min\{h, w\} - 3 \rceil\}$ **do**
3: **for** $\theta \in \{1, 2, \ldots, 2^{\lceil \lambda/2+3 \rceil}\}$ **do**
4: Take curvelet window $\phi(\lambda, \theta)$
5: Compute Fourier coefficients F_ϕ of curvelet window $\phi(\lambda, \theta)$, whose shape is like wedge
6: Compute curvelet coefficient by dot multiplication $F_\phi \cdot F$, whose shape is also like wedge
7: Wrap curvelet coefficient to a square-shaped form $F_C(\lambda, \theta) \leftarrow \text{Wrap}(F_\phi \cdot F)$
8: $C(\lambda, \theta) \leftarrow$ inverse Fourier transform of $F_C(\lambda, \theta)$
9: **return** $C(\lambda, \theta, k_1, k_2)$

where the operator $*$ represents two-dimensional convolution of two matrices and M is the convolution template for noise estimation, which is defined by Equation (4):

$$M = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \tag{4}$$

After that, from the results in refs. [18–20], a nonlinear function κ is introduced to adjust signals of different intensities in C, which is described in

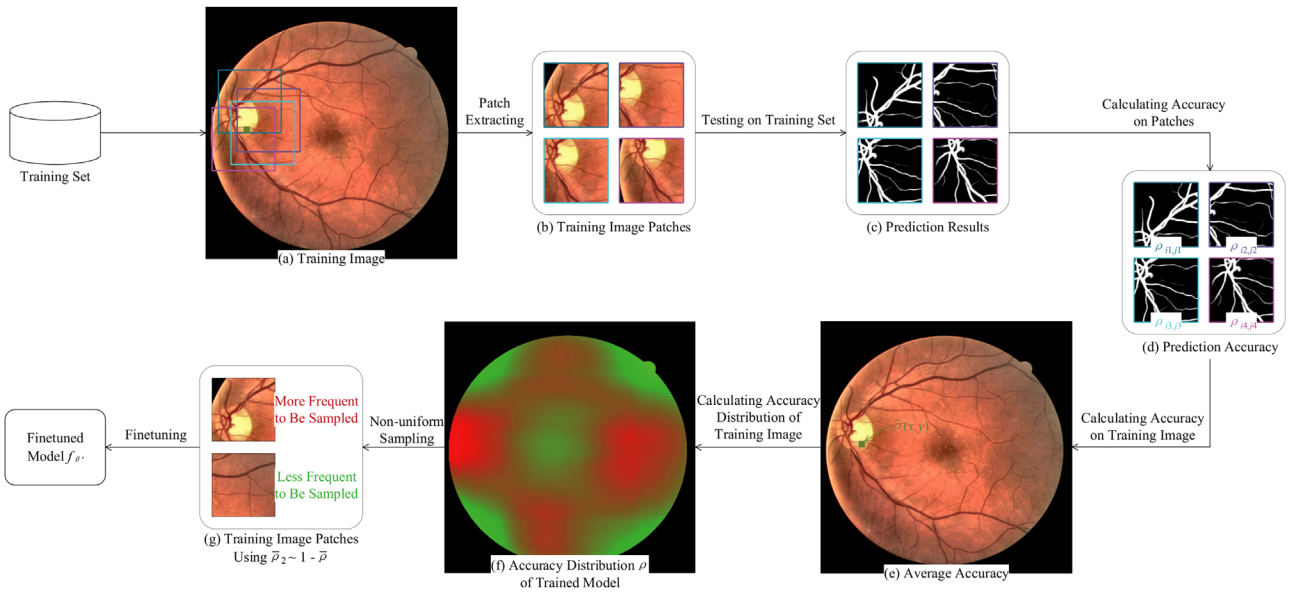

FIGURE 7 The framework of offline hard-sample mining schema. In the accuracy distribution of the predictions by the trained model, the redder colour represents the areas with lower overall accuracy $\bar{\rho}$ and the greener colour represents the areas with higher overall accuracy $\bar{\rho}$

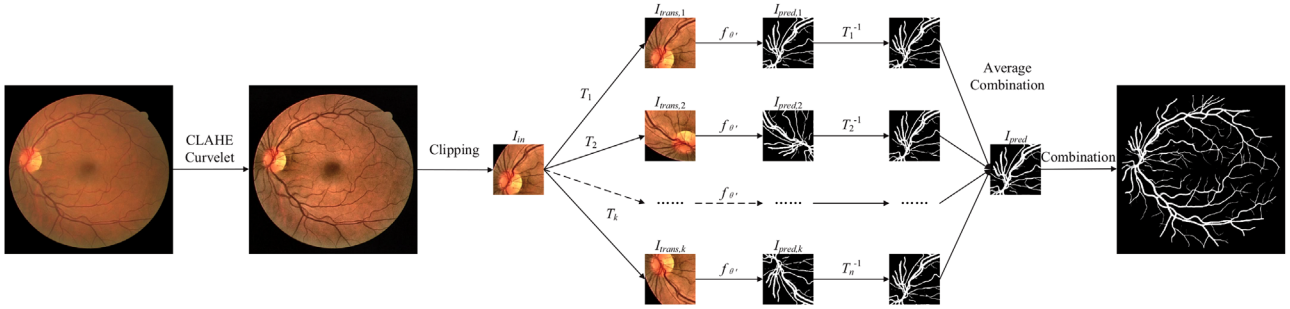

FIGURE 8 The inference process based on TTA. Various transformations are applied to the same patch of input images to obtain different patch inputs. After predicting on these patches l, the inverse transformations are applied to generate different predictions as probability maps on the patch. The final predictions are obtained by averaging these probability maps

TABLE 1 The values of parameters setting in the nonlinear function κ of our model

Parameter	Advised range	Set value
w_1	$w_1 \geq 1$	1
w_2	$w_2 \geq 1$	0.25
w_3	$w_3 > 0$	1
a	$0 < a < m/\sigma$	1.25
p	$0 < p < 1$	0.5

Equation (5):

$$\kappa(c) = \begin{cases} w_1 \cdot \left(\frac{m}{\sigma}\right)^p, & \text{if } |c| < a\sigma \\ w_2 \cdot \left(\frac{m}{|c|}\right)^p, & \text{if } a\sigma \leq |c| < m \\ w_3, & \text{if } |c| \geq m \end{cases} \tag{5}$$

where σ is an parameter estimation of Equation (3), w_1, w_2, w_3 are the weights of each part, and the adjustment factor a allows to adjust the interval of the function flexibly. The values of these parameters are shown in Table 1. m is a threshold value of curvelet coefficients C, and here m is set to be 85th percentile of the magnitude value of all curvelet coefficients $C(\lambda, \theta, k_1, k_2)$ empirically. To be more specific, we sort all the magnitudes of curvelet coefficients $C(\lambda, \theta, k_1, k_2)$ from the smallest to the largest, and then take 85th percentile to be the threshold m. After that, the new curvelet coefficients C' is computed by Equation (6).

$$C'(\lambda, \theta, k_1, k_2) = \kappa(C(\lambda, \theta, k_1, k_2)) \cdot C(\lambda, \theta, k_1, k_2) \tag{6}$$

Unlike approaches in refs. [18–20], the parameters σ and m are empirically computed globally, instead of locally in some scales, or in some orientations. After adjusting the curvelet coefficients, inverse FDCT is then applied to transform all the adjusted curvelet coefficients $C'(\lambda, \theta, k_1, k_2)$ back to the filtered image denoted as $I'(x, y)$. Next, $I'(x, y)$ is filtered with a 5×5 median filter to obtain final filtering of the image $I(x, y)$. Figure 4 shows a retinal image in DRIVE dataset and its corresponding annotation, greyscale image and curvelet enhanced result respectively. It can be inspected that pixel values with higher intensity in the arteries are suppressed and the non-vascular noise is reduced after curvelet enhancing. We concatenate the 1-channel filtering result and the 3-channel original RGB image into a 4-channel input for IterNet++.

3.2 | IterNet++: the neural network model for blood vessel segmentation

As shown in Figure 5, we design the backbone network of the model based on the architecture of IterNet [10], which consists of a four-layer U-Net (denoted as UNet_In) and three-layer mini U-Nets (denoted as UNet_Outs). Both UNet_In and UNet_Outs adopt an encoder–decoder architecture. Nevertheless, the network shares the parameters of UNet_Outs as a mechanism to reduce the number of parameters in the model in order to regularise the model and reduce overfitting. The model has several 1×1 convolutions in UNet_In and each iteration of UNet_Outs is used to generate segmentations based on the output features of shallowest-layer decoder. We note that the network performance is better when the encoders of UNet_In and UNet_Outs are replaced by residual convolutions (ResConvs, whose structure is shown in Figure 3(b)), while the decoders remain TwoConvs.

In the network structure of IterNet, UNet_Outs concatenate current shallowest-layer input features with all previous features obtained by shallowest-layer encoders. After that, it uses 1×1 convolutions to reduce the dimensions of concatenated features and obtain current shallowest-layer encoding features. In order to decrease the number of parameters and reduce model overfitting, IterNet only uses different dimension-reduction modules and 1×1 convolutions of final output in different iterations of UNet_Outs. However, the other encoder and decoder parameters are shared. Figure 6 visualises the absolute value of the weight of the dimension-reduction module (1×1 convolution) in the third iteration of UNet_Outs (i.e. UNet_Out 3). As shown in Figure 6, it can be discovered that the dimension-reduction module in UNet_Out 3 still tends to

focus on encoding features of iterations at beginning, especially the encoding features of UNet_In. We attempt to further explore the features from shallow-layer encoders of UNet_In. Therefore, the edge-preserving filter, i.e. guided filter [39, 40], is introduced to enhance the features from shallow-layer encoders of UNet_Outs. The guided filter takes an input image I and an output guidance O as inputs, and in the window w_k, the input image and output guidance are denoted as I_k and O_k respectively. The guided filter aims to generate the output $\hat{O}_k = a_k I_k + b_k$, so that the residual

$$E_k := \sum_{i \in w_k} \left(\left(a_k I_{k,i} + b_k - O_{k,i} \right)^2 + \lambda a_k^2 \right) \tag{7}$$

is minimised, where $a_k, b_k \in \mathbb{R}$ are coefficients in the window w_k that need optimising, and λ is a normalisation coefficient to restrict the magnitude of a_k. From results in ref. [39], the parameters of Equation (7) are calculated in Equations (8) and (9)

$$a_k = \frac{\mathrm{Cov}(I_k, O_k)}{\mathrm{Var}(I_k) + \lambda}, \tag{8}$$

$$b_k = \sum_{i \in w_k} \left(O_{k,i} - a_k I_{k,i} \right) \tag{9}$$

After obtaining the a_k and b_k in the window w_k, the box filters [39, 40] are used to calculate the results of guided filters for the original input image I and the output guidance O. As shown in Figure 5, guided filters are applied to the first and second-layer encoders of UNet_Outs to strengthen the impact of the result of the encoders from UNet_In on the result of the encoders from UNet_Outs. Therefore, the output features of encoders from UNet_Outs are locally linear with respect to current input features. Besides, the filtering results can be as close as possible to the output features of UNet_In. In the first-layer encoder of UNet_Outs, we take the dimension-reduction result as input I and the output feature from the first-layer encoder of UNet_In as output guidance O. In these guided filters, the size of the sliding window w_k is set to be 8×8 and the regularisation coefficient λ is set to be 10^{-3}. In the second-layer encoder of UNet_Out, we take the result of ResConv as input I and the output feature from the second-layer encoder of UNet_In as output guidance O. In these guided filters, the size of the sliding window w_k is set to be 4×4 and the regularisation coefficient λ is set to be 5×10^{-4}.

We use a UNet_In and 3 UNet_Outs to construct our network. During the training process, the minibatch strategy with size being 8 is employed to update weights. Generally, binary cross entropy (BCE) is applied as the optimisation objective to pixelwise segmentation problems. The calculation of BCE is shown in Equation (10):

$$\mathrm{BCE}(O, T) = -\frac{1}{nchw} \sum_{i=1}^{n} \sum_{j=1}^{c} \sum_{k=1}^{h} \sum_{l=1}^{w} \left(O_{ijkl} \log T_{ijkl} + (1 - O_{ijkl}) \log(1 - T_{ijkl}) \right), \tag{10}$$

where n, c, h, and w are the minibatch size, the number of image channels, the height, and the width of retinal images, respectively. T is the ground-truth vessel label of current minibatch with possible values of 0 or 1. Whilst O is the segmentation results by IterNet++ for the current minibatch with possible values of real numbers in 0∼1. Since IterNet++ generates segmentation results from UNet_In and each iteration of UNet_Outs, the actual loss is the sum of BCE for each output and the corresponding ground-truth label.

In the training process, the batch size is set to be 8. The Adam optimiser [41] is utilised, where the initial learning rate lr_0 is set to be 10^{-3}, the momentum parameter β_1 is set to be 0.9, and β_2 is set to be 0.999. During training, polynomial learning rate scheduling is applied to set the learning rate for current iteration. In the ith iteration, we set the learning rate $lr_i = lr_0(1 - N_{lr})^p$, where N_{lr} is the total number of iterations of the training process, and p is set to be 0.9.

In the training and validating process, each image is cut into blocks of size 128×128 stochastically, and then the blocks are fed into the network after normalisation using our pre-calculated means and variances. The weights in the network are updated using the backpropagation algorithm [10] in the training process. The mean and variance of all images are pre-calculated in the training set on each channel.

3.3 | Offline hard-sample mining schema

Generally, the vessel segmentation for retinal images is trained using uniform sampling, since the model always lacks a priori information about the dataset. Moreover, it is common practice to obtain blocks from each image and label in the training set using a sliding window of predefined size, e.g. 128×128. And then we train our model using these patches with equal probability by sampling them in a determined order or a random order. Nevertheless, it has a problem that the challenging hard samples with tiny vessels and complex surrounding texture are rarely sampled for training. Under this circumstance, the prediction performance of the model on these hard samples may be significantly poorer than those on easy samples. In order to alleviate this problem, an offline hard-sample mining schema is applied to our model. Figure 7 shows the framework of offline hard-sample mining, which is introduced as follows.

- **Patch extracting:** We denote the segmentation model to be f_θ after the training process. Image patches from each training in the training set are obtained with a sliding window of size being 128×128 and step size being 9. For a certain pixel in training, e.g. the green colour pixel in Figure 7(a), it is contained by a set of patches.
- **Testing on training set:** After that, with the center being (i, j), patch $I_{i,j}$ of size being 128×128 is extracted from a training image $I(x, y)$. We fix our weights θ and our model f_θ infers on these patches to obtain the inference results $\mathrm{Prob}(I_{i,j}) = f_\theta(I_{i,j})$. The size of $\mathrm{Prob}(I_{i,j})$ is 128×128, and the value of each element is between 0 and 1.

- **Calculating accuracy on patches:** The patch accuracy $\rho_{i,j}$ is calculated by averaging accuracy of every pixels in patch $I_{i,j}$. The corresponding ground-truth label of $I_{i,j}$ is denoted as $L_{i,j}$, and the patch accuracy is calculated as Equation (11):

$$\rho_{i,j} = \frac{1}{128^2} \sum_{1 \le k_1, k_2 \le 128} \| \text{Prob}(I_{i,j})(k_1, k_2) - L_{i,j}(k_1, k_2) \| \tag{11}$$

- **Calculating accuracy on training image:** From the previous steps, we obtain the accuracy of all patches $\rho_{i,j}$. Nevertheless, for a pixel (x, y) in training image, there are a set of patches containing the pixel in patch extracting. The accuracy of model f_θ on this pixel (x, y) is denoted as $\bar{\rho}(x, y)$, which is calculated by averaging accuracy $\rho_{i,j}$ of all patches $I_{i,j}$ containing this pixel.
- **Calculating accuracy distribution of training image:** The accuracy distribution of the training image I is obtained by traversing all pixels (x, y) in I and calculating the accuracy $\bar{\rho}(x, y)$.
- **Non-uniform sampling:** Figure 7 gives the visualisation results of $\bar{\rho}(x, y)$ on a training retinal image for a model trained with patches sampled in DRIVE dataset with equal probability. By visualising $\bar{\rho}(x, y)$, we observe that the segmentation accuracy of the model has a biased spatial distribution. The model tends to obtain higher accuracy in places with few vessels. However, the segmentation accuracy in areas with tiny and noisy vessels is often lower than expectation, since these areas are often challenging for the vessel segmentation problems. Therefore, in order to alleviate these problems, offline hard-sample mining on the patch dataset is performed based on the segmentation model f_θ obtained from the process of uniform sampling training. In other words, we finetune our model by accuracy-based non-uniform sampling.

  We denote $\rho_2(x, y)$ as the probability that a patch with center point (x, y) in image I is sampled in the non-uniform sampling. Following this definition, $\rho_2(x, y)$ can be computed by Equation (12)

$$\rho_2(x, y) = \frac{1 - \bar{\rho}(x, y)}{Z}, \tag{12}$$

  where Z is a normalisation parameter shown in Equation (13),

$$Z = \sum_{x,y} \left(1 - \bar{\rho}(x, y)\right) \tag{13}$$

- **Finetuning:** We obtain a tuned model $f_{\theta'}$ by performing finetune on the training set with $\rho_2(x, y)$ as the sampling weights. As shown in Figure 7(f,g), the image patches of the redder colour represents hard samples in the patch dataset, which are frequently repeated for sampling in finetuning process as for non-uniform sampling submodule.

  It is worth emphasising that inferring on patches one by one using the model f_θ is time-consuming. As a result, the strategy of computing the sampling weights $\rho_2(x, y)$ is adopted to perform finetune on f_θ for the patch dataset only once.

  In the finetuning process of offline hard-sample mining, minibatch with batch size being 8 is also used to perform finetuning. Nevertheless, the loss function applied to the process of offline hard-sample mining is different from that applied to the previous training process. Here, Dice loss [33] is introduced in the offline hard-sample mining, as Equation (14).

$$\text{Dice loss}(O, T) = \frac{2|O \cap T|}{|O \cap T| + |O \cup T|} \tag{14}$$

In the process of offline hard-sample mining, the loss function is a linear combination of BCE and Dice loss denoted as Equation (15),

$$\text{Loss}(O, T) = \alpha \text{BCE}(O, T) + (1 - \alpha)\text{Dice loss}(O, T), \tag{15}$$

where α is a hyperparameter and is set to be 0.3 empirically in our experiments. The Adam optimiser is utilised, where the initial learning rate lr_0 is set to be 10^{-4}, the momentum parameter β_1 is set to be 0.9 and β_2 is set to be 0.999. Moreover, polynomial learning rate scheduling is also applied to adjust the learning rate in the finetuning process, where p is set to be 0.9.

3.4 | Data augmentation of input images

To enhance input retinal images, RGB images are converted into CIE Lab space. Next, contrast is limited by using adaptive histogram equalisation (CLAHE) [42] for channel L, while channels a and b are kept unchanged. Afterwards, they are transferred back to the RGB space.

Since vessels to be segmented often have tiny and elongated structures, complex image transformations can interpolate the pixels of ground-truth labels. If these transformations are applied, the learning of segmentation of pixels near the edges of the ground-truth label cannot be facilitated. Therefore, only combinations of horizontal flips and clockwise rotations of multiples of 90 degrees are applied to augment input images and the corresponding ground-truth labels simultaneously. In addition, for each input image, colour jittering is utilised to randomly enhance the brightness, contrast, sharpness, and intensity of the image three times. As a result, the sample size of the dataset can reach 24 times of the original dataset after the image transformations mentioned above.

3.4.1 | Test-time augmentation (TTA)

Generally, data augmentation methods are applied to training datasets only. However, for the segmentation of images in the test dataset, TTA [43] can also be utilised to utilise the test dataset and the segmentation model trained on the augmented dataset. As shown in Figure 8, for testing, k transformations $T_1, T_2, \ldots, T_k$ are applied to the input image I_{in} to obtain $I_{\text{trans},1}, I_{\text{trans},2}, \ldots, I_{\text{trans},k}$. Next, the segmentation results

TABLE 2 The details of the models to be compared in the experiments

Model name	Year	Supervised	Enhancement	TTA	Hard-sample mining
U-Net	2016	✓			
Residual U-Net	2018	✓			
Curvelet & line	2018		Curvelet		
Oliveira et al.	2018	✓	Wavelet	✓	
DUNet	2019	✓			
ET-Net	2019	✓	Edge from label		
IterNet	2019	✓			
IterNet++ (ours)	2021	✓	Curvelet	✓	✓

of k images after transformations are predicted respectively by using our trained model $f_{\theta'}$, to obtain the predicted vessel probability maps $I_{\text{pred},1}, I_{\text{pred},2}, \ldots, I_{\text{pred},k}$. Finally, we apply the inverse transformation $T_1^{-1}, T_2^{-1}, \ldots, T_k^{-1}$ to these probability maps respectively, and take the mean value as our final prediction I_{pred}. The calculation of I_{pred} is as Equation (16).

$$\begin{aligned} I_{\text{pred}} &= \frac{\sum_{i=1}^{k} T_i^{-1}(I_{\text{pred},i})}{k} \\ &= \frac{\sum_{i=1}^{k} T_i^{-1}\left(f_{\theta'}(I_{\text{trans},1})\right)}{k} \\ &= \frac{\sum_{i=1}^{k} T_i^{-1}\left(f_{\theta'}(T_i(I_{\text{in}})\right)}{k} \end{aligned} \tag{16}$$

For testing, we use a sliding window with size being 128 × 128 and step size being 9 to cut the image into a number of patches, and feed them into the trained network $f_{\theta'}$ for inference. As shown in Figure 8, when the inference of all the patches of an image is completed, the final probability map is obtained for every pixel: the predicted probabilities of the patches containing a certain pixel are averaged to be the final predicted probability of this pixel. During inference, the model infers on each patch and then combine inference results on each patch spatially to obtain prediction results of the test images.

4 | EXPERIMENTS, RESULTS, AND DISCUSSION

In this section, we present retinal image datasets used in the experiments, namely DRIVE dataset, CHASE-DB1 dataset, and STARE dataset. Moreover, we present experiment results of our model and baselines [8–10, 19], e.g. U-Net and IterNet, on these datasets based on the metrics and methods adopted in these previous works. After that, performance comparisons are analysed and discussed. The details of the models to be compared are shown in Table 2.

4.1 | Datasets

We utilised the retinal image datasets DRIVE [44], CHASE-DB1 [45], and STARE [11] to evaluate the model proposed. DRIVE dataset [44] contains 40 retinal images with a resolution of 768 × 584, in which 20 images are used as the training set (images numbered 21 to 40), and the other 20 images are used as the test set (images numbered 1 to 20). We used images numbered from 21 to 36 as the training set and those numbered from 37 to 40 as the validation set. CHASE-DB1 dataset [45] contains 28 retinal images with a resolution of 999×960, without dividing training and test sets. STARE [11] dataset contains 20 retinal images with a resolution of 700 × 605, without dividing training and test sets. All datasets have corresponding ground-truth labels for each image. For CHASE-DB1 and STARE datasets, we use the same 4-fold and 5-fold cross-validations, respectively by referring to the work of Oliveira et al. [15]. These datasets are used mainly due to their relatively small size of samples, which is more challenging for the evaluation of models in the vessel segmentation task.

4.2 | Metrics

Metrics used in model evaluation are described in this subsection. Without loss of generality, we set the threshold to be 0.5 to obtain final segmentations. We denote the height of segmented images as h, the width as w, the segmentation result as O, and the ground-truth label as T.

4.2.1 | Accuracy

Accuracy is one of the most intuitive metrics in the segmentation task, which represents the percentage of the number of correctly segmented pixels with regard to the number of pixels in the whole image. The accuracy is computed by Equation (17):

$$\text{Acc}(O, T) = \frac{|O \cap T| + |\bar{O} \cap \bar{T}|}{hw} \tag{17}$$

However, it does not sufficiently reveal the performance of the models for segmentation on retinal images with an imbalanced proportion of foreground and background.

4.2.2 | Intersection over union (IoU)

The metric IoU is widely applied in the performance comparison of segmentation models. It reveals the proportion of correctly segmented foreground, and is concerned more with the foreground, which is the blood vessel in the retinal image segmentation problem. The IoU is shown as Equation (18)

$$\text{IoU}(O, T) = \frac{|O \cap T|}{|O \cup T|} \tag{18}$$

4.2.3 | Dice score

Dice score is widely introduced in the performance comparison of segmentation models. Similar to IoU, it also reveals the proportion of correctly segmented foreground. The Dice score is computed using Equation (19)

$$\begin{aligned}\text{Dice}(O,T) &= \frac{2|O \cap T|}{|O \cap T| + |O \cup T|} \\ &= 1 - \text{Dice loss}(O,T) \\ &= \frac{2}{1 + 1/\text{IoU}(O,T)}\end{aligned} \tag{19}$$

4.2.4 | Matthews correlation coefficient (MCC)

MCC is used to calculate the correlation of two binary variables and therefore is often applied to model performance comparisons [35, 46]. It is defined in Equation (20):

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}, \tag{20}$$

where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative respectively.

4.2.5 | Area under curve (AUC)

AUC is utilised to evaluate the performance of models when they cannot determine the best threshold for segmentation. For a certain threshold value, a model can segment input images using the threshold, and compare prediction with a ground-truth label to calculate the sensitivity and specificity of the model. Traversing through all possible thresholds, we can obtain many pairs of sensitivity and specificity values, and plot them as a receiver operating characteristic (ROC) curve. AUC is the area of the region enclosed by this curve with lines $y = 0$ and $x = 1$.

4.3 | Result comparisons of various models

The comparison between our model and baseline models on DRIVE dataset is shown in Table 3. The accuracy of our proposed model, i.e. IterNet++, is not as good as that of IterNet and our specificity is not as good as that of certain baseline models such as Residual U-Net. However, IterNet++ outperforms baseline models in terms of most evaluation metrics, especially Dice score, IoU and MCC which are more concerned with the results of foreground segmentation. Since offline hard-sample mining is performed, and our loss function is designed to be a linear combination of BCE and Dice loss, our model achieves a better Dice score than that of baseline models. The visualisation of each model on DRIVE dataset is presented in Figure 9, where the red circle displays the central region of the image. It can be observed that IterNet++ produced fewer false positives in the red circle due to the process of offline hard-sample mining, while all baseline models produced more false positives in the red circles. The orange and green circles give the noisy regions of the image. It can be inspected that IterNet without using the signal enhancement method was disturbed by the noises and thus generated more false positives. The model proposed by Oliveira et al. [15] uses wavelet transform to enhance input images and manages to alleviate the disturbance caused by image noises in the orange circle. However, it still generates many false positives in the green circle. ET-Net uses the edges of the ground-truth label as a guidance for attention, thus it produces fewer false positives in the red and green circles. However, since there is no signal enhancement to input images, it inevitably produces more false positives in the orange circle. In addition, the model has the low performance in AUC. Due to the selection of the loss function, ET-Net tends to generate more polarised segmentation results. IterNet++ prevented generating false positives in the orange and green circles by leveraging input enhancement with curvelet filtering.

The comparison between our model and baseline models on CHASE-DB1 dataset is shown in Table 4. As can be observed, our proposed model outperforms all baseline models in various metrics, especially Dice score, sensitivity, IoU and MCC. The visualisation of each model on CHASE-DB1 dataset is shown in Figure 10. The red and orange circles display tiny vessels. It is worth noting that our proposed model utilises a linear combination of BCE and Dice loss as the loss function in the process of offline hard-sample mining. Hence, IterNet++ achieves a higher segmentation probability on tiny vessels than baseline models. The arteries disturbed by noise are shown in the green circle. It can be inspected that segmentation result of IterNet++ in the green circle is more continuous than that of baseline models. This is mainly because that our model performs noise reduction with curvelet filtering and integrates edge-preserving guided filters to the network architecture.

The comparison of our model with baseline models on STARE dataset is shown in Table 5. Although there is a little gap in Dice score, accuracy, IoU, and MCC between our model and the model proposed by Oliveira et al., the AUC of our model is the best among the models. Moreover, the Dice score, IoU and MCC of our model is much higher than those of IterNet. The visualisation results of each model on STARE dataset are given in Figure 11. The tiny vessel branches are displayed in the red circle. IterNet++ had a higher segmentation probability on tiny vessels than baseline models due to the offline hard-sample mining performed utilising a linear combination of BCE and Dice loss as the loss function. The arteries with weak signal strength and reflections are displayed in the orange circle. Since IterNet++ reduced noise with curvelet enhancing, its segmentation in the orange circle is more continuous than those of baseline models. The noise area is displayed in the green circle. Due to offline hard-sample mining, IterNet++ was more robust against noise than the models other than ET-Net, and thus less probable to produce false positive segmentation results.

To demonstrate that the performance improvement of our model compared with baseline models, significance tests are

TABLE 3 Performance comparison with various baseline models on DRIVE dataset

Model name	Year	Dice score	Accuracy	Sensitivity	Specificity	IoU	MCC (*p*-value)	AUC (*p*-value)
U-Net	2016	0.8174	0.9555	0.7822	0.9808	0.6912	0.7994 (4.19×10^{-4})	0.9752 (1.68×10^{-7})
Residual U-Net	2018	0.8149	0.9553	0.7726	**0.9820**	0.6878	0.7983 (4.41×10^{-2})	0.9779 (4.47×10^{-2})
Curvelet & line	2018	0.7559	0.9542	0.7653	0.9735	0.6076	0.7467 (N/A)	N/A (N/A)
Oliveira et al.	2018	**0.8274**	**0.9576**	**0.8582**	0.9713	0.7060	0.8053 (9.58×10^{-1})	**0.9818** (8.15×10^{-2})
DUNet	2019	0.8190	0.9558	0.7863	0.9805	0.6934	0.8026 (4.35×10^{-2})	0.9778 (1.19×10^{-2})
ET-Net	2019	0.8279	0.9563	0.8294	0.9752	**0.7067**	**0.8059** (4.45×10^{-2})	0.9443 (6.10×10^{-11})
IterNet	2019	0.8205	**0.9573**	0.7735	**0.9838**	0.6953	0.8008 (7.19×10^{-4})	0.9816 (1.16×10^{-2})
IterNet++ (ours)	2021	**0.8313**	0.9569	**0.8399**	0.9742	**0.7115**	**0.8080**	**0.9828**

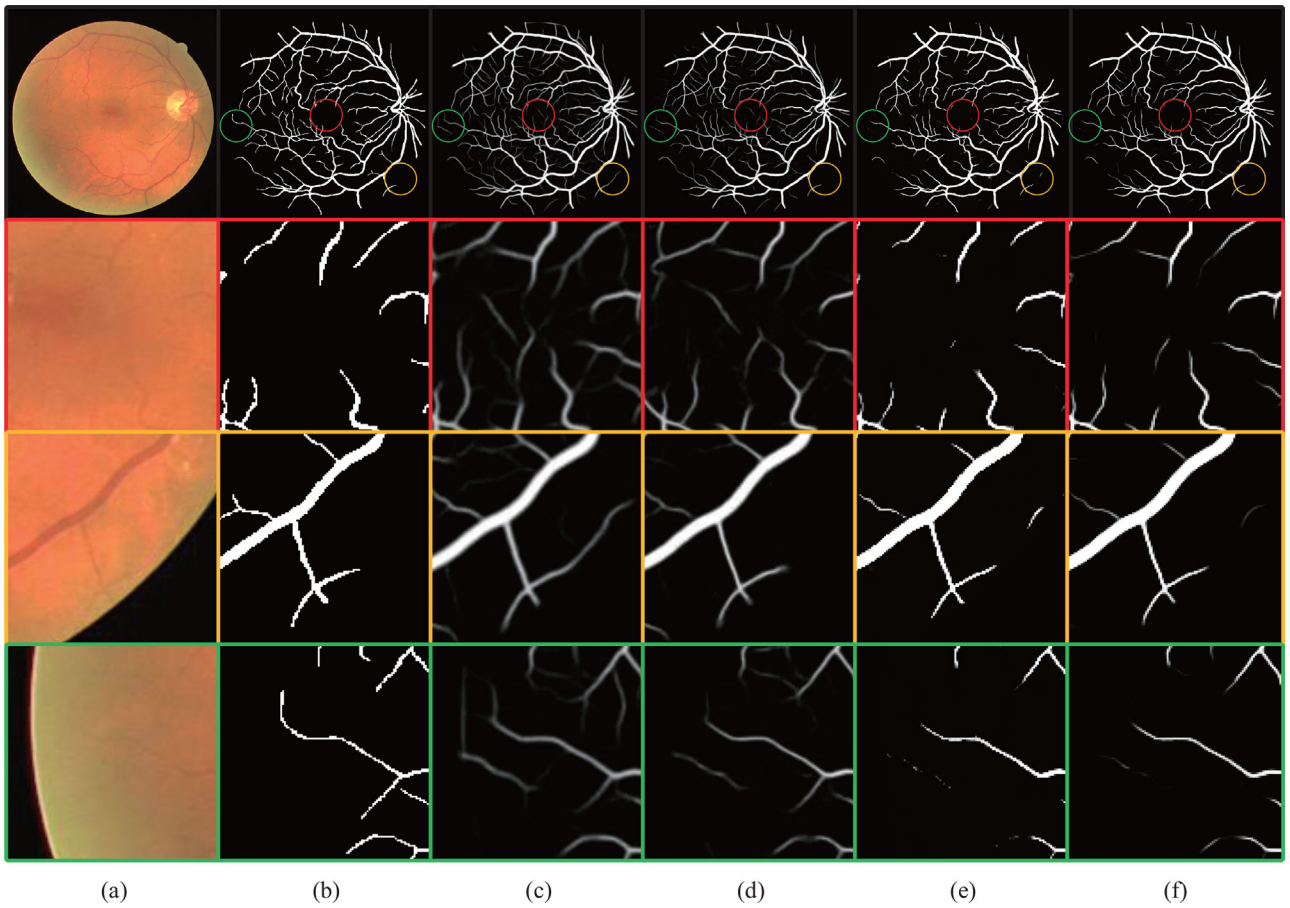

FIGURE 9 Result comparison of IterNet++ with baseline models on DRIVE dataset. (a) An image from the test set of DRIVE dataset, (b) the corresponding ground-truth label of (a), (c) segmentation results by IterNet, (d) segmentation results by the model proposed by Oliveira et al., (e) segmentation results by ET-Net, and (f) segmentation results by IterNet++

carried out. For IoU, Dice score and MCC, paired *t*-tests are conducted with $P < 0.05$ between our results and results from baseline models. The results are shown in Tables 3–5, indicating that IterNet++ has significant overall improvements than the state-of-the-art baseline models.

4.4 | Ablation study

We conduct an ablation study on the DRIVE dataset to verify the effectiveness of the modules in our proposed model.

TABLE 4 Performance comparison with various baseline models on CHASE-DB1 dataset

Model name	Year	Dice score	Accuracy	Sensitivity	Specificity	IoU	MCC (*p*-value)	AUC (*p*-value)
U-Net	2016	0.7993	0.9643	0.7841	**0.9823**	0.6660	0.7905 (1.12×10^{-12})	0.9812 (2.35×10^{-10})
Oliveira et al.	2018	**0.8172**	**0.9653**	0.7768	**0.9865**	**0.6915**	0.8006 (2.99×10^{-4})	**0.9851** (4.33×10^{-5})
DUNet	2019	0.8001	0.9644	0.7859	0.9822	0.6668	0.7967 (5.19×10^{-8})	0.9834 (6.26×10^{-7})
ET-Net	2019	0.7916	0.9602	0.7638	0.9821	0.6565	0.7721 (8.02×10^{-10})	0.9505 (4.65×10^{-16})
IterNet	2019	0.8073	**0.9655**	**0.7970**	**0.9823**	0.6768	**0.8024** (6.37×10^{-8})	**0.9851** (4.05×10^{-6})
IterNet++ (ours)	2021	**0.8277**	**0.9659**	**0.8247**	0.9820	**0.7069**	**0.8108**	**0.9867**

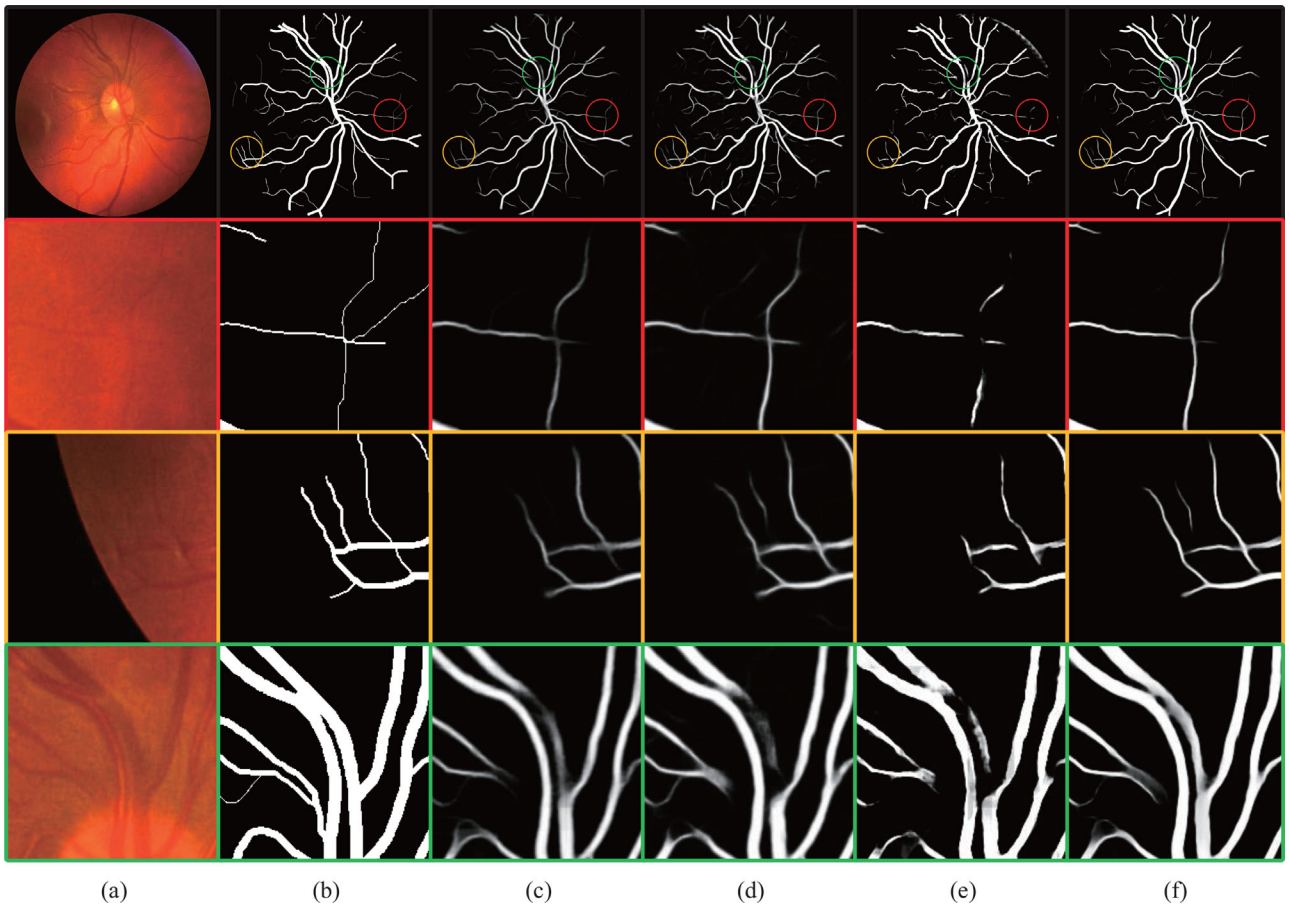

FIGURE 10 Comparison results of IterNet++ with baseline models on CHASE-DB1 dataset. (a) An image from the test set of CHASE-DB1 dataset, (b) the corresponding ground-truth label of (a), (c) segmentation results by IterNet, (d) segmentation results by the model proposed by Oliveira et al., (e) segmentation results by ET-Net, and (f) segmentation results by IterNet++

TABLE 5 Performance comparison with various baseline models on STARE dataset

Model name	Year	Dice score	Accuracy	Sensitivity	Specificity	IoU	MCC (*p*-value)	AUC (*p*-value)
U-Net	2016	0.8046	0.9634	0.7608	**0.9862**	0.6789	0.7912 (4.01×10^{-5})	0.9849 (2.90×10^{-3})
Oliveira et al.	2018	**0.8399**	**0.9683**	**0.8219**	0.9859	**0.7265**	**0.8247** (6.62×10^{-1})	0.9885 (6.26×10^{-1})
DUNet	2019	0.8192	0.9655	0.7855	0.9858	0.6996	0.8055 (3.51×10^{-4})	0.9865 (6.03×10^{-3})
ET-Net	2019	0.8146	0.9652	0.7805	**0.9861**	0.6954	0.8034 (3.16×10^{-2})	0.9497 (8.96×10^{-5})
IterNet	2019	0.8268	0.9664	0.7940	0.9859	0.7103	0.8116 (2.21×10^{-3})	**0.9886** (4.72×10^{-2})
IterNet++ (ours)	2021	**0.8372**	**0.9675**	**0.8211**	0.9841	**0.7230**	**0.8218**	**0.9892**

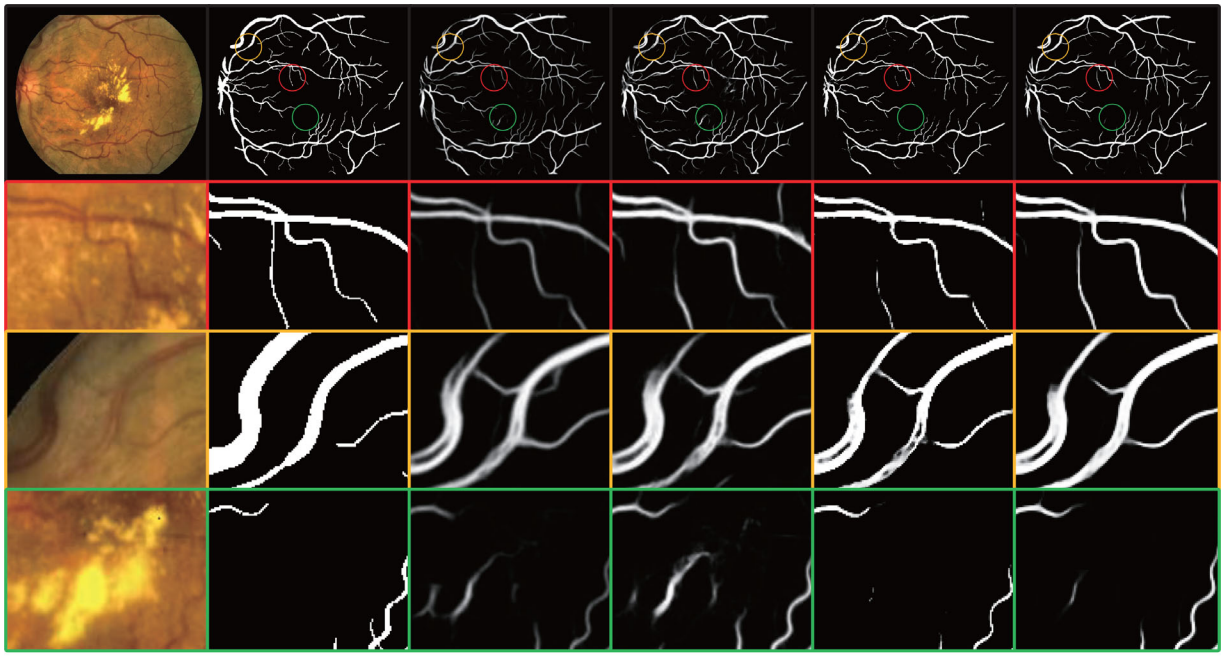

FIGURE 11 Comparison results of IterNet++ with baseline models on STARE dataset. (a) An image from the test set of STARE dataset, (b) the corresponding ground-truth label of (a), (c) segmentation results by IterNet, (d) segmentation results by the model proposed by Oliveira et al., (e) segmentation results by ET-Net, and (f) segmentation results by IterNet++

TABLE 6 The result of module addition performed on DRIVE dataset. In the table header, R denotes ResConv encoder, C denotes image enhancement based on curvelet enhancing, G denotes guided filters in UNet_Outs, H denotes offline hard-sample mining, and T denotes TTA

R	C	G	H	T	Dice score	Accuracy	Sensitivity	Specificity	IoU	MCC	AUC
					0.8205	**0.9573**	0.7735	**0.9838**	0.6953	0.8008	0.9816
✓					0.8228	0.9566	0.7971	0.9801	0.6995	0.7998	0.9805
✓	✓				0.8245	0.9567	0.8043	0.9792	0.7018	0.8018	0.9817
✓	✓	✓			0.8260	0.9571	0.8048	0.9796	0.7039	0.8035	0.9821
✓	✓	✓	✓		0.8289	0.9563	0.8341	0.9745	0.7079	0.8052	0.9814
✓	✓	✓	✓	✓	**0.8313**	0.9569	**0.8399**	0.9742	**0.7115**	**0.8080**	**0.9828**

4.4.1 | Module addition

As shown in Figure 5, we add the following modules one by one based on the primitive IterNet: first, we replace TwoConv module from all encoders with ResConv module (R); second, we integrate image enhancement based on curvelet enhancing to the input (C); third, we attach guided filters in UNet_Outs (G); fourth, we introduce offline hard-sample mining (H), and fifth, we perform TTA method (T). The DRIVE dataset is used for training and testing. The experiment results of the module addition (marked with ✓) and without any module addition are shown in Table 6. It can be seen that with the additions of the modules, the Dice score, IoU, and sensitivity of the models show an increasing trend, while the MCC is also increasing on the whole. Although the performances on the other metrics are fluctuating, their fluctuations are relatively small. This indicates that the modules proposed are all effective in performance improvement to the vessel segmentation of retinal images.

4.4.2 | Module removal

From the complete IterNet++, modules contained ResConv, image enhancement based on curvelet enhancing, guided filters, offline hard-sample mining, and TTA method, are removed one by one. The results of module removal on DRIVE dataset are shown in Table 7. It can be seen that ResConv, hard-sample mining and TTA are quite helpful for the improvements in Dice score, IoU and MCC. Since image enhancement based

TABLE 7 The result of module removal experiments performed on DRIVE dataset

Removed module	Dice score	Accuracy	Sensitivity	Specificity	IoU	MCC	AUC
ResConv	0.8261	0.9552	0.8409	0.9721	0.7040	0.8021	0.9813
Curvelet	0.8304	0.9569	0.8337	0.9752	0.7103	0.8073	0.9825
Guided filter	0.8301	0.9575	0.8199	**0.9779**	0.7098	0.8075	0.9829
Hard-sample mining	0.8278	**0.9576**	0.8054	0.9801	0.7066	0.8057	**0.9831**
TTA	0.8289	0.9563	0.8341	0.9745	0.7079	0.8052	0.9814
None	**0.8313**	0.9569	**0.8399**	0.9742	**0.7115**	**0.8080**	0.9828

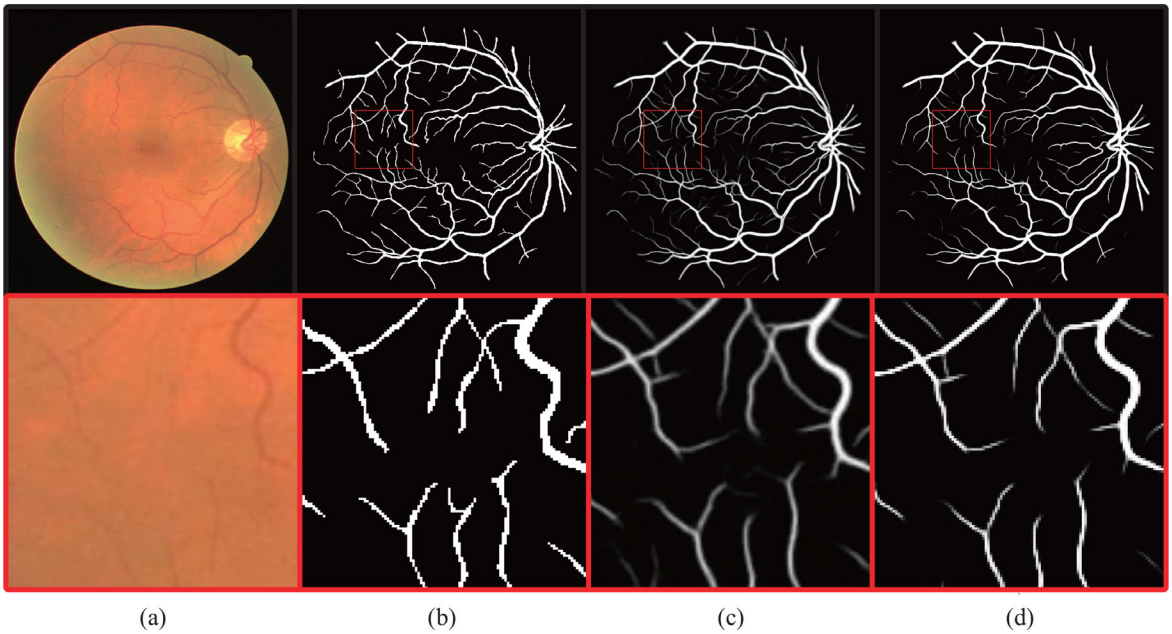

FIGURE 12 The comparison of segmentation results before and after offline hard-sample mining. (a) An image in the test set of DRIVE dataset, (b) the ground-truth label corresponding to (a), (c) the model prediction results before performing offline hard-sample mining, and (d) the model prediction results after performing offline hard-sample mining

on curvelet enhancing and guided filters enhances the features near the edge of vessels, they contribute to the segmentation of tiny vessels, which is a crucial task in vessel segmentation for retinal images. Moreover, they improve the segmentation near the edge of vessels. Therefore, they bring clear performance improvement in the segmentation.

4.4.3 | Offline hard-sample mining

Figure 12 shows the segmentation results of an image in the test set of DRIVE dataset before and after offline hard-sample mining. It can be seen that after offline hard-sample mining, the segmentations of hard samples in the red circle are more determined for tiny vessels and at the edges of the vessels, i.e. prediction values closer to 0 or 1 by adopting the loss function as Equation (15) for finetune.

4.5 | Discussion

4.5.1 | Model analysis

Based on the iterative structure of IterNet, IterNet++ enables multiple predictions and optimisations of segmentations, which play an important role in retinal vessel segmentation that requires the segmentation of tiny vessels. In particular, for datasets with small sample size, e.g. DRIVE [44], CHASE-DB1 [45], and STARE [11], the guided filters are enhanced to UNet_Outs, thus exploiting shallow features extracted by UNet_In, e.g. illumination, edges etc. In addition, during the training process, we utilise an offline hard sample mining strategy, which allows the model to benefit more from the training data. Moreover, it prevents the phenomenon that the model accuracy tends to have an unbalanced distribution spatially.

4.5.2 | Data processing

Data enhancement and augmentation is one of the most important tools in deep learning methods to suppress model overfitting. Fast discrete curvelet transform and filtering is applied to retinal images for better extraction and enhancement of vessel features. These are inspired by results from Quinn and Krishnan [18] and are validated in subsequent experiments. Besides, TTA is introduced as a postprocessing method for inference results in the inference process. To some extent, this alleviates the problem led by limited size of samples in DRIVE, CHASE-DB1, and STARE datasets, and also validates the conclusion of Amiri et al. [43]

4.5.3 | Comparisons with existing methods

IterNet++ is compared with a list of baseline models. Among them, U-Net [8], Residual U-Net [47], Oliveira et al. [15], DUNet [35], ET-Net [9], and IterNet [10] are supervised methods based on deep learning, and Curvelet & Line [19] is an unsupervised method based on curvelet and line operators. Overall, our method outperforms the baseline methods on all the DRIVE, CHASE-DB1 and STARE datasets, especially with the metrics IoU, Dice score and MCC. This indicates that IterNet++ is suitable for the retinal vessel segmentation. Visually, compared to those of other methods, the inference results of our model are better. This is demonstrated by the fact that the inference results are less ambiguous and have fewer artefacts.

4.5.4 | Limitations

Our model may still has the following limitations. The box filters are used in the actual computation of the guided filters, which is not conducive to computing with GPUs. Moreover, in offline hard-sample mining process, the time consumption is actually somewhat high, because computing the accuracy distribution $\bar{p}$ of the model with respect to the images in the training set requires inference over a sliding window of all images. How to further optimise the model, such as using other GPU computationally friendly filters, will be a direction of our future work. Moreover, segmentation voids, tiny blood vessels, and noisy regions are still three major challenges in retinal image segmentation, and the performance of our model still has the room to be further optimised. In addition, how to further utilise the existing prior knowledge from the datasets to input images for feature enhancement may be also a direction for future research.

5 | CONCLUSION

In this paper, an improved network named IterNet++ was proposed based on IterNet. For input images, it used curvelet enhancing to reduce noise and enhances the images. For the network architecture, it applied guided filters to normalise features extracted from UNet_Outs to those from UNet_In for reducing model overfitting. For the training process, it utilised offline hard-sample mining to make more use of samples that were difficult to train in small size datasets. For the inference process, it introduced TTA to make full use of images from test set. By comparing the model with a list of state-of-the-art baseline models, IterNet++ outperformed all baseline models on vessel segmentation of retinal images. Furthermore, we verified the effectiveness of proposed modules through ablation experiments. The results demonstrated that our model was effective to deal with the problems of segmentation voids, small probability of tiny vessel segmentation, and false positives in noisy regions.

ACKNOWLEDGMENT

The publication of this paper is funded by National Science Foundation of China (No. U1711266), Key-Area Research and Development Program of Guangdong Province (No. 2019B010153001), and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2019A1515011078).

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available in DRIVE, CHASE_DB1, and STARE, reference number [11, 44, 45]. These data were derived from the following resources available in the public domain: https://drive.grand-challenge.org/, https://www.idiap.ch/software/bob/docs/bob/bob.db.chasedb1/master/index.html, and http://cecas.clemson.edu/~ahoover/stare/probing/index.html.

ORCID

K. Zeng https://orcid.org/0000-0002-2211-5533

REFERENCES

1. Ma, Y., Hao, H., Xie, J., Fu, H., Zhang, J., Yang, J., et al.: ROSE: a retinal OCT-angiography vessel segmentation dataset and new model. IEEE Trans. Medical Imaging 40, 928–939 (2021)
2. Patton, N., Aslam, T., MacGillivray, T., Deary, I., Dhillon, B., Eikelboom, R., et al.: Retinal image analysis: concepts, applications and potential. Progress Retinal Eye Res. 25, 99–127 (2006)
3. Abramoff, M., Garvin, M., Sonka, M.: Retinal imaging and image analysis. IEEE Rev. Biomed. Eng. 3, 169–208 (2010)
4. Group, T.S.: Long-term complications in youth-onset type 2 diabetes. N. Engl. J. Med. 385, 416–426 (2021)
5. Teo, Z., Tham, Y., Yu, M., Chee, M., Rim, T., Cheung, N., et al.: Global prevalence of diabetic retinopathy and projection of burden through 2045: Systematic review and meta-analysis. Ophthalmology 128(11), 1580–1591 (2021)
6. Wykoff, C., Khurana, R., Nguyen, Q., Kelly, S., Lum, F., Hall, R., et al.: Risk of blindness among patients with diabetes and newly diagnosed diabetic retinopathy. Diabetes Care 44, 748–756 (2021)
7. Kipli, K., Hoque, M., Lim, L., Mahmood, M., Sahari, S., Sapawi, R., et al.: A review on the extraction of quantitative retinal microvascular image feature. Comput. Math Methods Med. 2018, 1–21 (2018)
8. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention, pp. 234–241. Springer, Cham (2015)
9. Zhang, Z., Fu, H., Dai, H., Shen, J., Pang, Y., Shao, L.: ET-Net: a generic edge-attention guidance network for medical image segmentation. In:

Proceedings of the Medical Image Computing and Computer Assisted Intervention, pp. 442–450. Springer, Cham (2019)
10. Li, L., Verma, M., Nakashima, Y., Nagahara, H., Kawasaki, R.: IterNet: retinal image segmentation utilizing structural redundancy in vessel networks. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp. 3645–3654. IEEE, Piscataway, NJ (2020)
11. Hoover, A., Kouznetsova, V., Goldbaum, M.: Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response. IEEE Trans. Med. Imaging 19, 203–210 (2000)
12. Shensa, M.: The discrete wavelet transform: wedding the atrous and mallat algorithms. IEEE Trans. Signal Process. 40, 2464–2482 (1992)
13. Biswal, B., Vyshnavi, E., Sairam, M., Rout, P.: Robust retinal optic disc and optic cup segmentation via stationary wavelet transform and maximum vessel pixel sum. IET Image Process 14, 592–602 (2020)
14. Upadhyay, K., Agrawal, M., Vashist, P.: Unsupervised multiscale retinal blood vessel segmentation using fundus images. IET Image Process 14, 2616–2625 (2020)
15. Oliveira, A., Pereira, S., Silva, C.: Retinal vessel segmentation based on fully convolutional neural networks. Expert Syst. Appl. 112, 229–242 (2018)
16. Starck, J., Candes, E., Donoho, D.: The curvelet transform for image denoising. IEEE Trans. Image Process 11, 670–684 (2002)
17. Candes, E., Demanet, L., Donoho, D., Ying, L.: Fast discrete curvelet transforms. Multiscale Model Simul. 5, 861–899 (2006)
18. Quinn, E., Krishnan, K.: Retinal blood vessel segmentation using curvelet transform and morphological reconstruction. In: Proceedings of the 2013 IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology, pp. 570–575. IEEE, Piscataway, NJ (2013)
19. Chalakkal, R., Abdulla, W.: Improved vessel segmentation using curvelet transform and line operators. In: Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, pp. 2041–2046. IEEE, Piscataway, NJ (2018)
20. Miri, M., Far, A.: Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction. IEEE Trans. Biomed. Eng. 58, 1183–1192 (2011)
21. Esmaeili, M., Rabbani, H., Dehnavi, A., Dehghani, A.: A new curvelet transform based method for extraction of red lesions in digital color retinal images. In: Proceedings of the International Conference on Image Processing, pp. 4093–4096. IEEE, Piscataway, NJ (2010)
22. Esmaeili, M., Rabbani, H., Dehnavi, A., Dehghani, A.: Automatic detection of exudates and optic disk in retinal images using curvelet transform. IET Image Process. 6, 1005–1013 (2012)
23. Thangaraj, S., Periyasamy, V., Balaji, R.: Retinal vessel segmentation using neural network. IET Image Process. 12, 669–678 (2018)
24. Chen, Z., Jin, W., Zeng, X., Xu, L.: Retinal vessel segmentation based on task-driven generative adversarial network. IET Image Process. 14, 4599–4605 (2020)
25. Rammy, S., Abbas, W., Hassan, N., Raza, A., Zhang, W.: CPGAN: conditional patch-based generative adversarial network for retinal vessel segmentation. IET Image Process. 14, 1081–1090 (2020)
26. Chen, C., Chuah, J., Raza, A., Wang, Y.: Retinal vessel segmentation using deep learning: a review. IEEE Access 9, 111985–112004 (2021)
27. Brancati, N., Frucci, M., Gragnaniello, D., Riccio, D.: Retinal vessels segmentation based on a convolutional neural network. In: Proceedings of the Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, pp. 119–126. Springer, Cham (2018)
28. Yan, Z., Yang, X., Cheng, K.: A three-stage deep learning model for accurate retinal vessel segmentation. IEEE J. Biomed. Health. Inf. 23, 1427–1436 (2018)
29. Guo, Y., Budak, U., Vespa, L., Khorasani, E., Sengur, A.: A retinal vessel detection approach using convolution neural network with reinforcement sample learning strategy. Measurement 125, 586–591 (2018)
30. Nagi, J., Ducatelle, F., Caro, G.D., Ciresan, D., Meier, U., Giusti, A., et al.: Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: Proceedings of the 2011 IEEE International Conference on Signal and Image Processing Applications, pp. 342–347. IEEE, Piscataway, NJ (2011)
31. Im, D., Han, D., Choi, S., Kang, S., Yoo, H.: DT-CNN: dilated and transposed convolution neural network accelerator for real-time image segmentation on mobile devices. In: Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 1–5. IEEE, Piscataway, NJ (2019)
32. Zhou, Z., Siddiquee, M., Tajbakhsh, N., Liang, J.: UNet++: a nested U-Net architecture for medical image segmentation. In: Proceedings of the Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pp. 3–11. Springer, Cham (2018)
33. Milletari, F., Navab, N., Ahmadi, S.: V-Net: fully convolutional neural networks for volumetric medical image segmentation. In: Proceedings of the Fourth International Conference on 3D Vision, pp. 565–571. IEEE, Piscataway, NJ (2016)
34. Fu, H., Cheng, J., Xu, Y., Wong, D., Liu, J., Cao, X.: Joint optic disc and cup segmentation based on multi-label deep network and polar transformation. IEEE Trans. Medical Imaging 37, 1597–1605 (2018)
35. Jin, Q., Meng, Z., Pham, T., Chen, Q., Wei, L., Su, R.: DUNet: a deformable network for retinal vessel segmentation. Knowl. Based Syst. 178, 149–162 (2019)
36. Wang, S., Yin, Y., Cao, G., Wei, B., Zheng, Y., Yang, G.: Hierarchical retinal blood vessel segmentation based on feature and ensemble learning. Neurocomputing 149, 708–717 (2015)
37. Yang, J., Dong, X., Hu, Y., Peng, Q., Tao, G., Ou, Y., et al.: Fully automatic arteriovenous segmentation in retinal images via topology-aware generative adversarial networks. Interdiscip. Sci.: Comput. Life Sci. 12, 323–334 (2020)
38. Zhao, Z.-B., Yuan, J.-S., Gao, Q., Kong, Y.-H.: Wavelet image de-noising method based on noise standard deviation estimation. In: Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, pp. 1910–1914. IEEE, Piscataway, NJ (2007)
39. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Trans. Pattern. Anal. Mach. Intell. 35, 1397–1409 (2013)
40. Kleesiek, J., Urban, G., Hubert, A., Schwarz, D., Maier-Hein, K., Bendszus, M., et al.: Deep MRI brain extraction: a 3D convolutional neural network for skull stripping. NeuroImage 129, 460–469 (2016)
41. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations, pp. 1–13. ICLR, La Jolla, CA (2015)
42. Pizer, S., Amburn, E., Austin, J., Cromartie, R., Geselowitz, A., Greer, T., et al.: Adaptive histogram equalization and its variations. Comput. Vision, Graphics, Image Process. 39, 355–368 (1987)
43. Amiri, M., Brooks, R., Behboodi, B., Rivaz, H.: Two-stage ultrasound image segmentation using U-Net and test time augmentation. Int. J. Comput. Assisted Radiol. Surg. 15, 981–988 (2020)
44. Staal, J., Abramoff, M., Niemeijer, M., Viergever, M., Ginneken, B.: Ridge-based vessel segmentation in color images of the retina. IEEE Trans. Med. Imag. 23, 501–509 (2004)
45. Owen, C., Rudnicka, A., Mullen, R., Barman, S., Monekosso, D., Whincup, P., et al.: Measuring retinal vessel tortuosity in 10-year-old children: validation of the computer-assisted image analysis of the retina (CAIAR) program. Invest. Ophthalmol. Visual Sci. 50, 2004–2010 (2009)
46. Chicco, D., Jurman, G.: The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21, 1–13 (2020)
47. Alom, M., Hasan, M., Yakopcic, C., Taha, T., Asari, V.: Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation. arXiv:1802:06955 (2018)

How to cite this article: Zhu, M., Zeng, K., Lin, G., Gong, Y., Hao, T., Wattanachote, K., Luo, X.: IterNet++: An improved model for retinal image segmentation by curvelet enhancing, guided filtering, offline hard-sample mining, and test-time augmenting. IET Image Process. 16, 3617–3633 (2022). https://doi.org/10.1049/ipr2.12580