

Classification-based framework for binarization on mice eye image in vivo with optical coherence tomography

Fei Ma^{1*} | Cuixia Dai² | Jing Meng^{1*} | Ying Li¹ | Jingxiu Zhao¹ | Yuanke Zhang¹ | Shengbo Wang¹ | Xueting Zhang¹ | Ronghua Cheng¹

¹School of Computer Science, Qufu Normal University, Shandong, China

²College Science, Shanghai Institute of Technology, Shanghai, China

*Correspondence

Fei Ma and Jing Meng, School of Computer Science, Qufu Normal University, Shandong, China.
 Email: mafei0603@163.com (F. M.); jingmeng@qfnu.edu.cn (J. M.)

Funding information

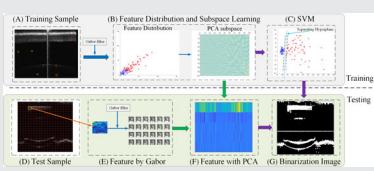
Guangdong Provincial Key Laboratory of Biomedical Optical Imaging Technology, Grant/Award Number: 2020B121201010; Natural Science Foundation of Shandong Province, Grant/Award Number: ZR2020MF105; Shandong Province Science Funder, Grant/Award Number: 2021360096

Abstract

Optical coherence tomography (OCT) angiography has drawn much attention in the medical imaging field. Binarization plays an important role in quantitative analysis of eye with optical coherence tomography. To address the problem of few training samples and contrast-limited scene, we proposed a new binarization framework with specific-patch SVM (SPSVM) for low-intensity OCT image, which is open and classification-based framework. This new framework contains two phases: training model and binarization threshold. In the training phase, firstly, the patches of target and background from few training samples are extracted as the ROI and the background, respectively. Then, PCA is conducted on all patches to reduce the dimension and learn the eigenvector subspace. Finally, the classification model is trained from the features of patches to get the target value of different patches. In the testing phase, the learned eigenvector subspace is conducted on the pixels of each patch. The binarization threshold of patch is obtained with the learned SVM model. We acquire a new OCT mice eye (OCT-ME) database, which is publicly available at <https://mip2019.github.io/spsvm>. Extensive experiments were performed to demonstrate the effectiveness of the proposed SPSVM framework.

KEY WORDS

binarization threshold, contrast-limited image, OCT mice eye database, optical coherence tomography, specific-patch SVM framework



1 | INTRODUCTION

Optical coherence tomography angiography (OCTA) [1] is a light-based and non-invasive imaging technique, and can get image of tissue *in vivo* at micrometer resolutions. OCTA has been widely used in medical diagnostics, such as diabetic retinopathy (DR), multiple sclerosis (MS), multiple sclerosis (MS), glaucoma and age-related macular degeneration (AMD) [2–4]. In practical researches, it is critical to get high-quality OCTA images, which has applications in clinical trials to monitor disease

progression. For example, vessel area density (VAD) is the most widely used OCTA measurement, which uses the binarized image to quantify the percentage of the image occupied by ROI as a fraction of the whole image.

Plenty of studies have focused on the quantitative analysis of retinal vascular and choriocapillaris OCTA images in recent years [5–7]. Machine learning methods have seen substantial growth in the processing of biomedical images, including the biomedical image reconstruction, image denoising and disease classification [8–10]. Binarization is a fundamental task in analysis of

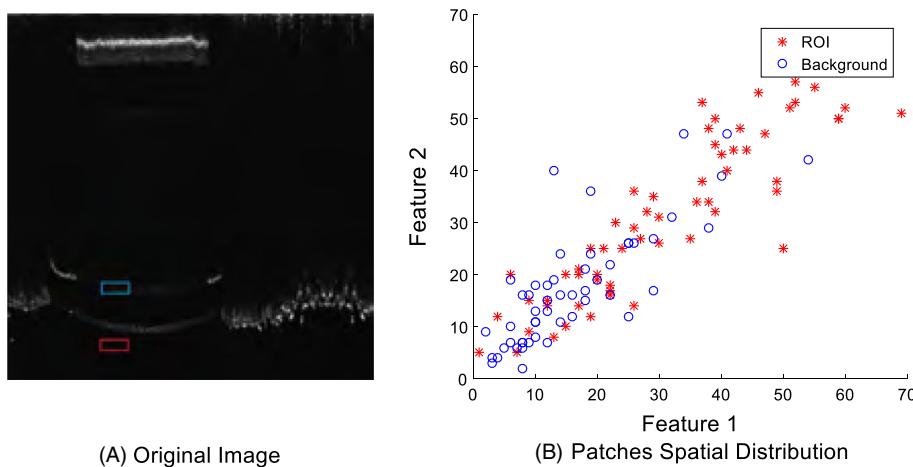


FIGURE 1 The feature spatial distribution of patches from the ROI and background. (A) The representative sample. (B) The illustration of pixels distribution for the ROI and background. Feature1 is from the first dimension of each patch and feature2 is from the second dimension of each patch

retinal image *in vivo* by optical coherence tomography (OCT) [9, 11–16], which has been widely applied for simple quantification of area and more complex vessel analyses [17, 18].

The binarization methods can be divided into two categories: global threshold and local threshold. (1) Global threshold-based methods are effective and simple, which are suitable for the pixels with large difference, including Otsu [19–22], Iterative binarization [23–25]; (2) the local threshold methods can obtain the optimal threshold for the subimage of samples, including Niblack [26, 27], K-means [28, 29]. Although these algorithms are simple and fast, they are sensitive to noise.

Recently, many deep-learning-based methods have been presented for OCTA images and focused on the retinal vessel segmentation. Specifically, the convolutional neural networks (CNNs) have achieved great success in biomedical image processing [30]. Plenty of networks employ a UNet architecture as the backbone and get good performance for segmentation. UNet is based on an encoder-decoder architecture and utilizes the skip connections to extract effective features. It can integrate the high-level semantic feature maps from the decoder and corresponding low-level feature maps from the encoder [31].

1.1 | Motivation

Inspired by literature [32], to observe the distribution of the ROI and background, we select two areas of the tissue and background respectively, and extract the pixel features from them by patch in Figure 1A. As shown in Figure 1B, one can observe that the feature distributions of the ROI (marked in red) and the background (marked in blue) are overlapped to some extent, which will be difficult to distinct them directly. For low-intensity or uneven illumination images, the effective binarization-thresholding methods are desirable for the complex

scene. Motivated by the above analysis, we intend to design an approach, which can overcome the contrast-limited problem and make full use of patch features for few-shot learning.

1.2 | Contribution

The main contributions in this paper are 3-fold:

1. A novel binarization framework based on specific-patch classification is proposed in this paper, which can handle the low-intensity OCT image. With the learned eigenvector subspace by PCA, we can get the features of each patch from testing samples. With the learned SVM model, the threshold of any patches can be adaptively obtained. The specific-patch framework can make full use of small number of samples and extract many ROIs from each sample.
2. We capture an OCT mice eye (OCT-ME) database, which has 50 images in total. Each image is 2048×2048 pixels. Intensity of some tissues in most images is very low, which will be difficult to binarization.
3. Comprehensive experiments are conducted on OCT-ME database. The results demonstrate the effectiveness of the proposed approach.

The remaining of this paper is organized as follows. A proposed framework of binarization techniques is described in Section 2. In Section 4, comprehensive experiments are conducted. Finally, a conclusion of this work is summarized in Section 4.

2 | PROPOSED METHOD

In this section, we first introduce the proposed method specific-patch SVM. Then we describe the overview of

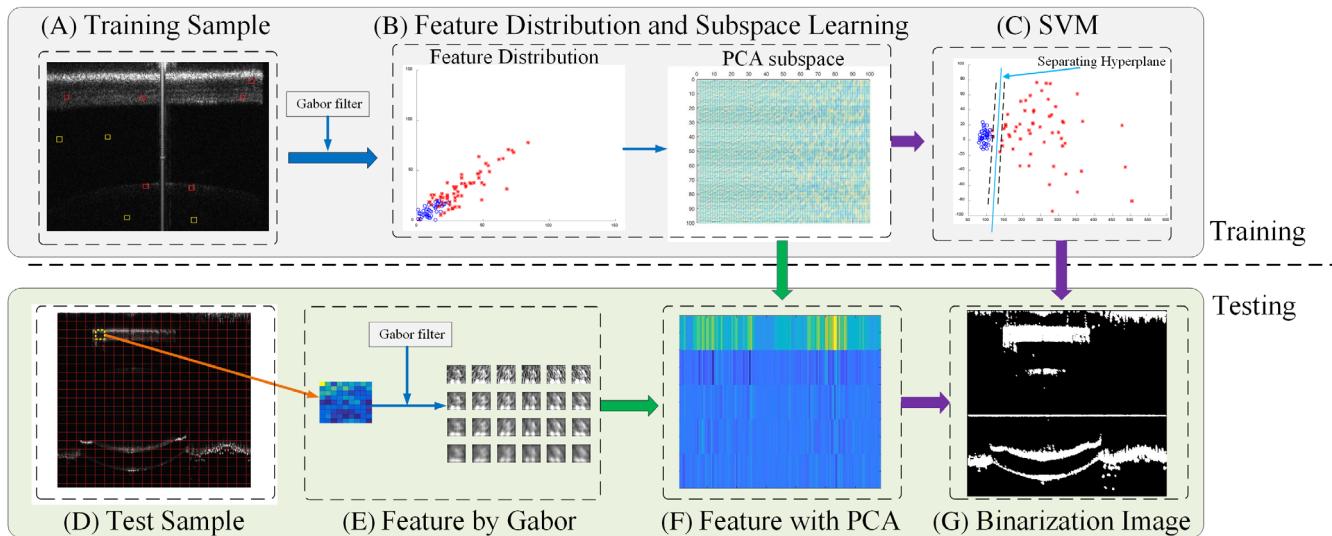


FIGURE 2 The flowchart of our proposed method specific-patch SVM. (A) The selected ROI and background with red and yellow rectangle respectively; (B) The spatial distribution of ROI and background, and the learned eigenvector subspace by PCA; (C) Training the classification model by SVM; (D) The testing images with patches; (E) Features extracted by Gabor filter bank; (F) The feature of patches with the learned eigenvector subspace; (G) The binarization image by the trained SVM model

image preparation and feature extraction. Finally, we detail the SVM based methods for binarization threshold.

2.1 | Binarization pipeline based on specific-patch SVM

To get binarization threshold of the low-intensity eye image *in vivo* by OCT, we propose a new framework specific-patch SVM based on classification, which can obtain threshold for each patch by the way of machine learning. The basic idea is shown in Figure 2. There exist two phases in the framework.

1. In the training phase, we first select a few samples as training set, which can include normal and very low-intensity images. As shown in Figure 2A, we select some typical patches as ROI to extract the features by Gabor filters. The patches in red rectangle are ROI and others in yellow rectangle are background. To get effective features, PCA is utilized to get important vectors of all patches, and the feature eigenvector subspace can be learned in Figure 2B. SVM can learn a separating hyperplane from training data, which can distinct the pixels of ROI and background.
2. In the testing phase, we first split all patches from the testing sample in Figure 2D. The feature of each patch can be got by the Gabor filter bank in Figure 2E. The low-dimension features can be got with the learned eigenvector subspace of PCA in Figure 2F. With the

learned SVM model from the trained samples, we can obtain the binarization threshold of each patch for an OCT image. The final binarization image of mice eye is shown in Figure 2G.

2.2 | Image preparation

1. *Specific-patch selection.* In the novel framework, we manually select the ROI and background from a few training samples by specific patch are shown in Figure 2A. The size of ROI(background) ranges from $[23 \times 35]$ to $[109 \times 95]$ pixels. We select the low-intensity areas of any tissue (such as corneal and iris) as the ROI marked with 1 and the areas not belonged to tissues as the background marked with -1. Then we extract the Gabor features from the selected sub-images by patch, whose size is 10×10 pixels. The procedure of specific-patch selection is shown in Figure 3. We select very low-intensity and normal areas at the ROI from the training samples respectively.
2. *Gabor extraction.* The effective feature can be extracted from samples with rich texture by Gabor filters bank [33], which has been widely used for classification and recognition [34–36]. We analyze the ROI areas from the OCT samples in Figure 4. We select the ROI from the original sample as Figure 4A. The pixel distribution is shown in Figure 4B. One can see that there exists very rich texture in the ROI area. Figure 4C shows that the Gabor features of the

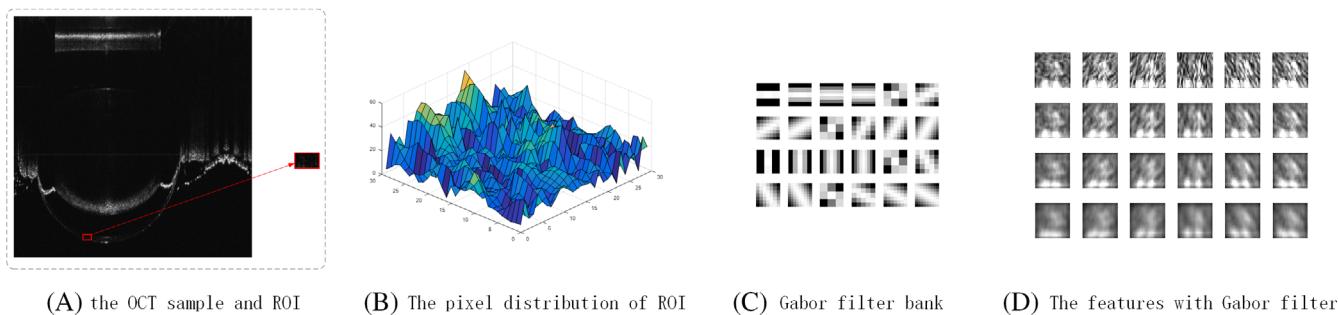


FIGURE 3 The procedure of specific-patch selection. We select the ROI (in red rectangle) and background (in orange rectangle) areas from samples respectively. The patches of the ROI and background are marked as +1 and -1, respectively

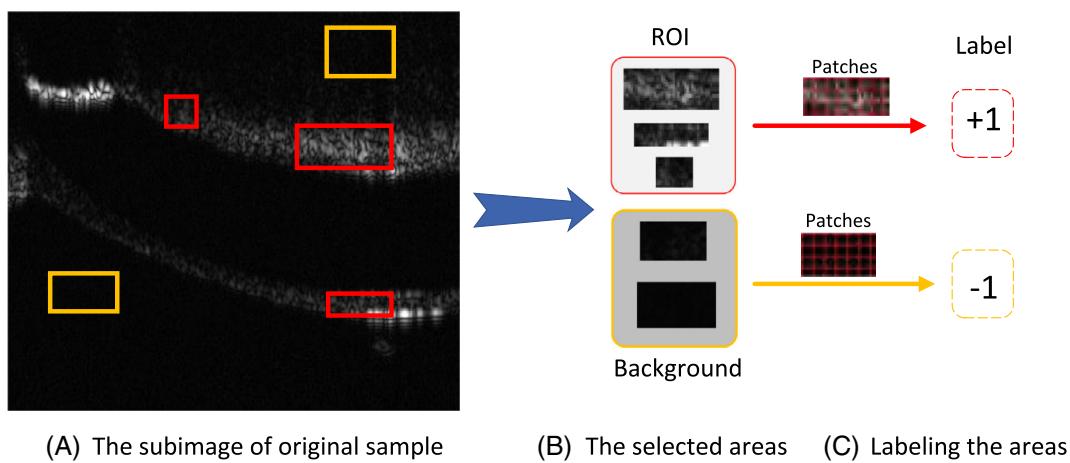


FIGURE 4 The distribution of specific ROI and Gabor features. (A) Selecting the ROI from sample; (B) The pixel distribution of ROI; (C) The Gabor filter bank at 6 orientations and 4 scales; (D) The patch feature of specific ROI with the Gabor filter bank

selected areas are extracted by Gabor filter bank, which can be calculated as follows:

$$G(x,y,\theta,\mu,\sigma) = \frac{1}{2\pi\sigma^2} \exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\} \exp\{iz\}, \quad (1)$$

$$z = 2\pi i(\mu x \cos\theta + \mu y \sin\theta), \quad (2)$$

where σ represents the standard deviation of the Gaussian envelope. $i = \sqrt{-1}$. θ denotes the orientation of Gabor filters, and μ is the frequency of the sinusoidal wave. As shown in Figure 4C, we extract the orientation and scale features from patches of samples by controlling θ and σ in the Gabor filter bank, respectively. The dimension of Gabor features is high. For example, the Gabor feature dimension of 10×10 patch at 6 orientations and 4 scales achieves 2400 in Figure 4C. Therefore, we can get the principal component of features by the algorithm of dimension reduction.

3. *Dimension reduction.* As above analysis, the dimensions of original feature directly extracted from

samples are often high. Principal components analysis (PCA) is a wide and effective method for reducing dimension and learning eigenvector subspace, which can improve the performance of algorithm. The basic idea of PCA is to learn an eigenvector subspace and map the high-dimension features to the low-dimension features. There exist principal components in the high-dimension features, which can improve the effectiveness of the original features. With the learned eigenvector subspace, we can get effective representation of tested samples. We abstract the calculating procedure of PCA as the following function:

$$S_{proj} = F_{PCA}(X), \quad (3)$$

where S_{proj} represents the projection subspace from the training set. $F_{PCA}(\cdot)$ is the function for calculating procedure of PCA. X denotes the covariance matrix of the training samples. The low-dimension feature x_f can be obtained by $S_{proj}^T * x$. And x is the Gabor features of the training (or testing) sample.

ALGORITHM 1 The procedure of our proposed framework

Input: The original OCT image.

Output: The binarized image.

1. Training Phase

(1) **Labeling the training samples.** We construct the training set by marking the ROI as 1 and the background as -1 in Figure 3.

(2) **Extracting the Gabor Features.** The Gabor features of specific patches are extracted by Gabor filter bank Eq. (1);

(3) **Learning subspace.** The projection subspace S_{proj} can be obtained by Eq. (3) from the Gabor features.

(4) **Training SVM.** SVM is used to classify the ROI and background. The optimizing procedure can be obtained by Formula (4).

2. Testing Phase

(1) **Partitioning the patches.** We partition the whole image by patch with 10×10 pixels as shown in Figure 2D.

(2) **Extracting the Gabor Features.** The Gabor feature of each patch can be obtained by Eq. (1).

(3) **Reducing Dimension.** The low-dimension features can be calculated by the eigenvector subspace S_{proj} , which is learned by Eq. (3).

(4) **Thresholding image.** With the learned SVM model from the trained samples, we can classify each patch of the testing OCT image. With the mathematical morphology, the output image is dilated and eroded. The final binarized image is shown in Figure 2G.

2.3 | Training SVM Model

SVM (support vector machine) has been widely used for pattern recognition [37–39] and classification [40–42]. The basic idea is to project the input vectors into a new feature space, which can obtain a separating hyperplane from the training samples.

Given a training set $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, $y_i \in \{-1, +1\}$, for our training model, x_i is the i th feature of patch. y_i is the label of the i th patch, which is written as $y_i = W^T x_i + b$, where $W = (w_1; w_2; \dots; w_d)$ is the normal vector, which decides the direction of hyperplane. b is the displacement value, which presents the distance between the hyperplane and the coordinate origin. The

separating hyperplane can identify the features as the corresponding classes.

The goal related to (W, b) can be written as follows:

$$\begin{aligned} & \min_{W, b} \frac{1}{2} \|W\|^2 \\ & \text{s.t. } y_i(W^T x_i + b) \geq 1, (i = 1, 2, \dots, n). \end{aligned} \quad (4)$$

We will get the optimal separating hyperplane from the training set to classify the background and foreground by Eq. (5).

$$\begin{cases} W^T * x_i + b > = 1, & y_i = 1; \\ W^T * x_i + b < = -1, & y_i = -1. \end{cases} \quad (5)$$

If $y_i = 1$, it means that this patch (x_i) is classified as the ROI; otherwise, the current patch (x_i) is classified as the background. The optimal solutions of W and b can be obtained by using the algorithm in literature [43]. Our framework includes the training and testing phase. We conclude the procedure of our framework as Algorithm 1.

3 | EXPERIMENTS

3.1 | Experimental setup

3.1.1 | Materials

We capture the database of mouse eyes by the customized spectral-domain optical coherence tomography (SD-OCT) system in Figure 5. A total of 50 images taken from mice *in vivo* are recorded by OCT. The size of each image is 2048×2048 pixels. The eye of mouse generally contains four tissues, such as retina, lens, iris and cornea. As shown in Figure 6A, one can see that the background (in yellow rectangle) and foreground (in red rectangle) of some tissues are low intensity. This dataset has been released publicly.* The ground-truth samples were manually made by our experts in Figure 6B, which are aided with our self-developed software. This software is a specialized tool to make the ground truth from original samples under complex scene. Our latest tool (named as MakeGT1.3) for ground truth is developed in C# under .netframework2.0. We can update it according to new requirements, which can be download at URL (see footnote 1). In practice, our tool can provide several types of service for making ground truth in Figure 7, such as cell segmentation, the connected region of tissues, vascular segmentation and so on. The final ground-truth samples should be advised by experts.

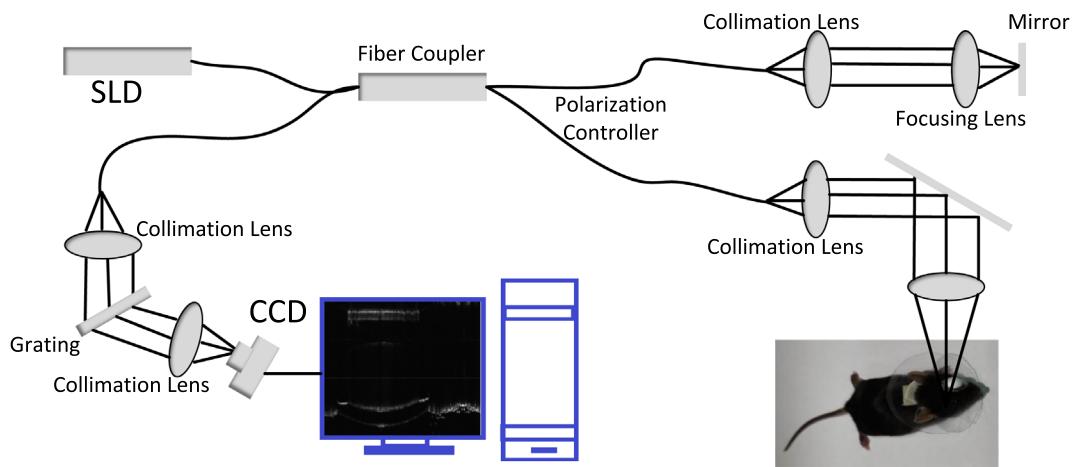


FIGURE 5 Schematic diagram of SD-OCT system. The super luminescent diode (SLD) is centered at 840 nm and the full width at half maximum of 50 nm

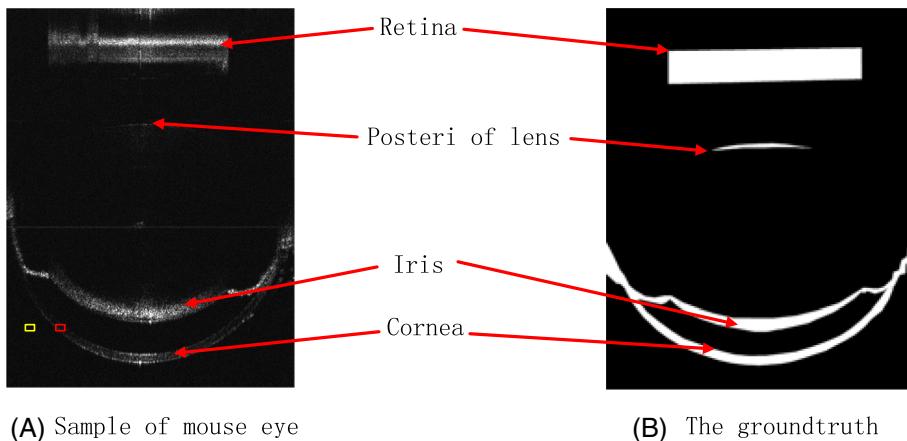


FIGURE 6 The structure of mouse eye. It contains several tissues in the eyes, such as retina, lens, iris and cornea. (A) The original sample of mouse eye. The area in yellow rectangle is background and one in red rectangle is ROI; (B) The corresponding ground-truth image

3.1.2 | Evaluation setting

To assess the binarization thresholding performance of the proposed method, we adopt the evaluation metrics as follows: F1-score [44], dice coefficient, sensitivity and specificity [45].

We select 5 images for the training set and 45 images for the testing set randomly. According to the contrast curve of the testing images, one sample is selected for every nine samples in Figure 8.

3.2 | F1-score

F1-score is also called as F-measure. F1-score is the weighted harmonic average of precision and recall [44]. It is a common evaluation standard and often used to evaluate the quality of classification models. The F1-score can be computed as follows:

$$F_{\beta} = \frac{(\beta^2 + 1)P * R}{\beta^2 * P + R}, \quad (6)$$

where P is the precision and R denotes recall. β is a tuning parameter. $P(\text{precision}) = \text{TP}/(\text{TP} + \text{FP})$, where TP notes the number of correctly classified pixels of ROI. FP is the number of false positives. $R(\text{Recall}) = \text{TP}/(\text{TP} + \text{FN})$, where FN is false negative. When $\beta = 1$, it is the most common F1-score. F1-score can provide an evaluation with precision and recall. The higher F1-score is, the more effective the model is.

The quantitative comparisons of F1-score, precision and recall are shown in Table 1. One can observe that our approach achieves the highest F1-score values on the testing samples. The R values of ours are over 70%. It means that our approach can recognize more areas of ROI than the competitive methods, that is, the highest rate of true positive of ROI. Furthermore, we extract the edge of ROI from the ground truth and cover the results of the representative method and SPSVM with it in Figure 9. One can observe that our method can correctly detect more pixels of ROI (inside the red edge of the ground truth) than the other method (in Figure 9E). However, our method identifies many false positive pixels by

FIGURE 7 The illustration of our tool for making ground truth. In each row the left is the original image with MakeGT and the right is the partial results by our tool. (A) The ground truth of eye tissues by MakeGT; (B) The cell segmentation by MakeGT; (C) The vascular segmentation by MakeGT

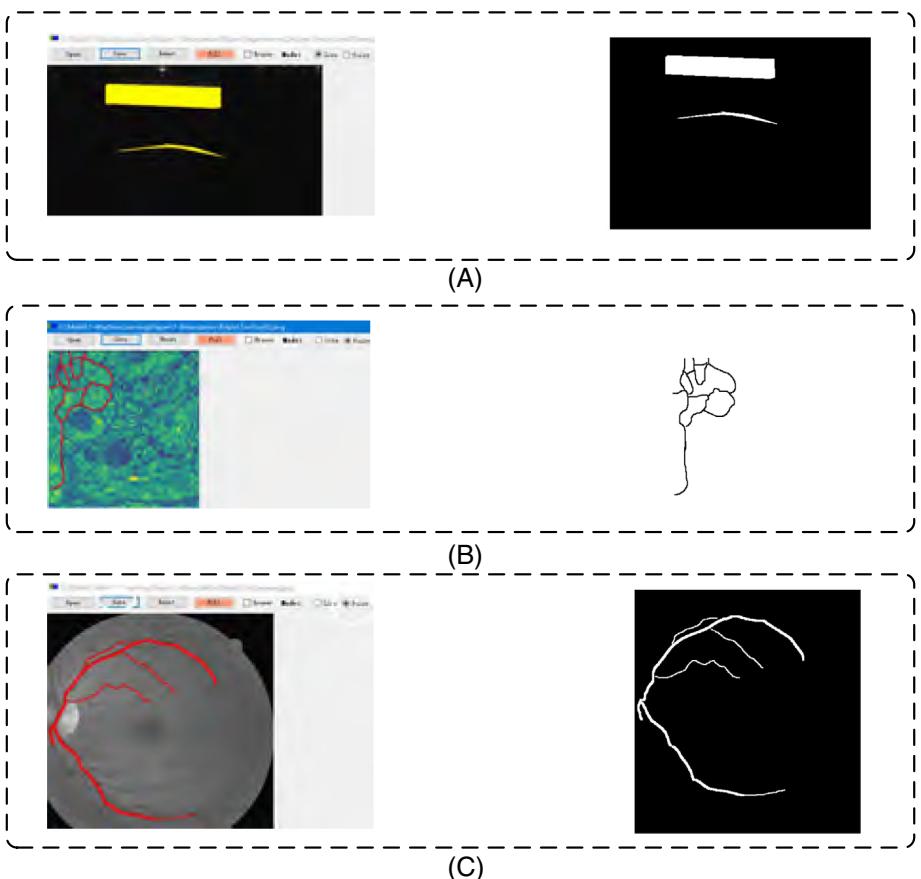
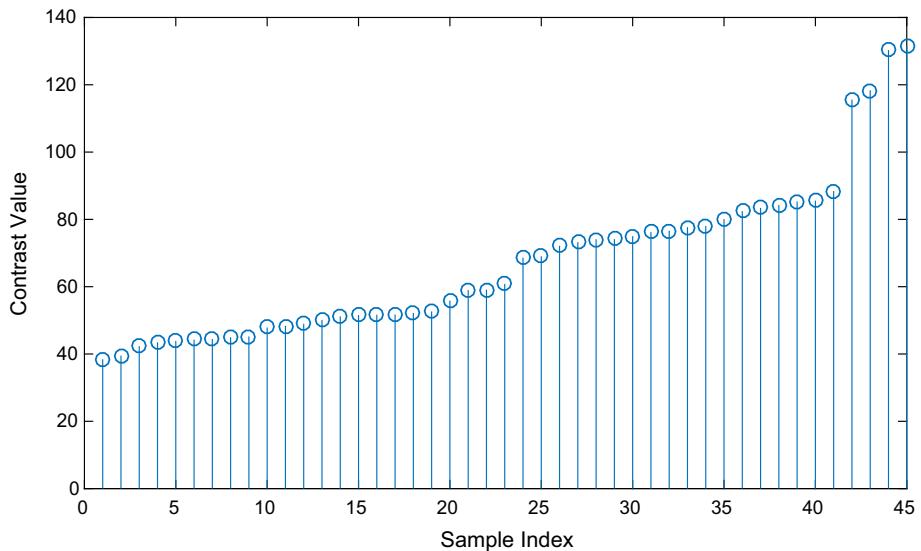


FIGURE 8 The contrast of the testing samples in ascending order

red arrows in Figure 9F, which belong to the background in fact. The reason is that our method classifies the pixels in boundary area as ROI. This suggests that more effective features are learned to identify the ROI correctly.

3.3 | Further performance comparison

To achieve further and objective evaluation of the performance of the proposed method, we utilize the evaluation metrics [45] as follows:

TABLE 1 F1-score (precision/recall)(%) of different algorithms on the selected testing images

Image	Iteration	Otsu	K-means	Ours
#1	39.79(64.18/28.83)	35.18(64.36/24.21)	23.82(60.72/14.82)	68.22 (57.93/82.94)
#2	58.53(66.83/52.06)	50.64(72.74/38.84)	56.71(68.89/48.17)	60.37 (48.72/79.35)
#3	51.16(66.14/41.72)	39.15(64.99/28.01)	46.61(63.31/36.89)	60.27 (47.71/81.81)
#4	52.31(68.09/42.47)	46.31(70.06/34.57)	54.23(67.22/45.45)	65.07 (52.89/85.64)
#5	57.68(60.92/54.76)	46.51(64.44/36.37)	48.69(63.01/39.67)	58.63 (43.95/88.03)
Avg.	51.89(65.23/43.97)	43.55(67.32/32.41)	46.01(64.63/37.01)	62.55 (50.24/83.46)

Note: Best results are in bold.

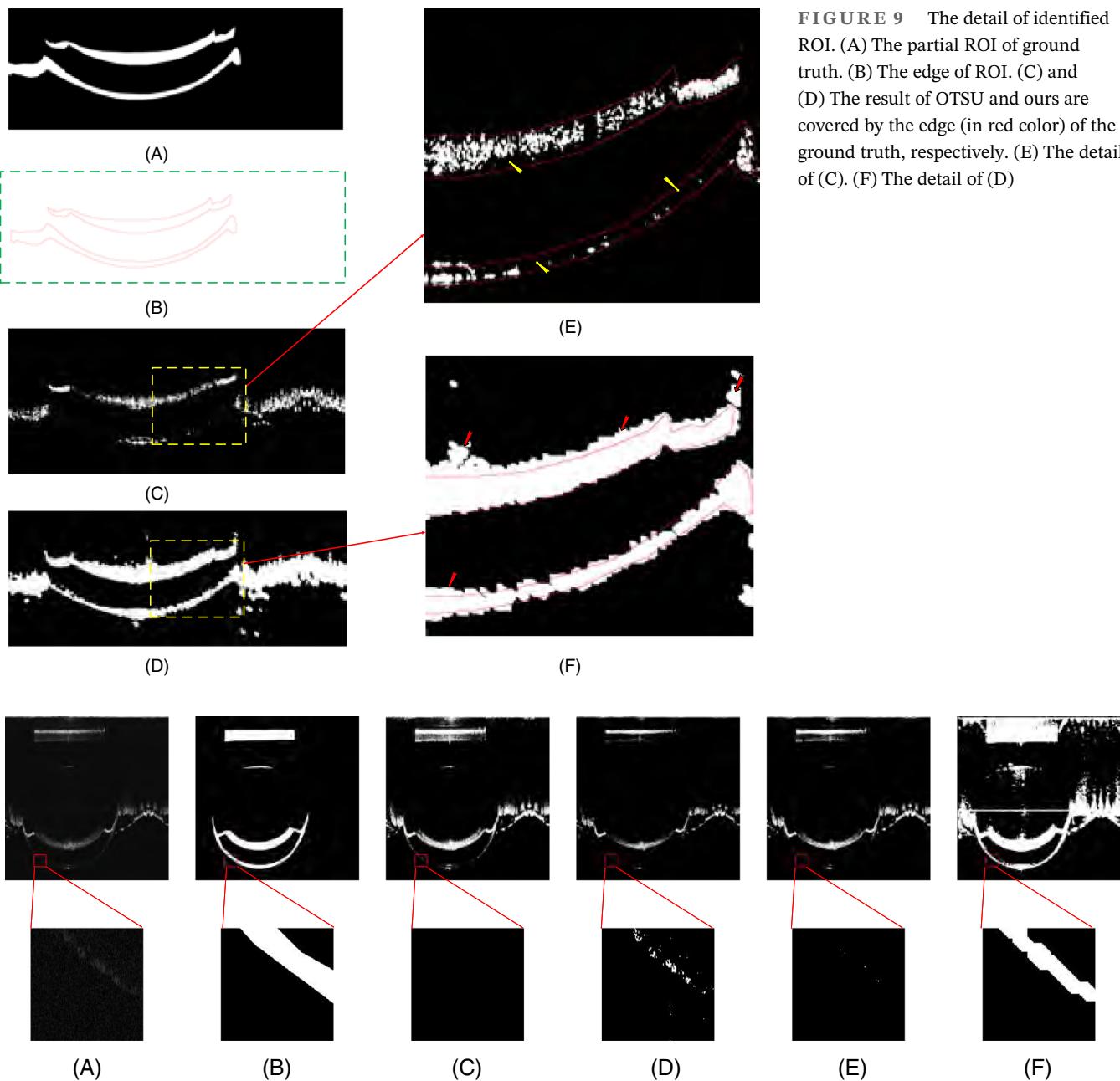


FIGURE 10 The original samples, the ground truth and the results of different methods. (A) the original samples; (B) the ground-truth samples. (C-E): The results by the Iteration, Otsu and K-means methods. (F) The results by specific-patch SVM. One can see that the brightness of the original samples is low. The ROI areas of tissues are remained in the ground-truth samples

- Sensitivity = $TP/(TP + FN)$;
- Specificity = $TN/(TN + FP)$;
- Dice coefficient = $2 \times TP/(FP + FN + 2 \times TP)$, where TN is the number of true negative pixels, and FN is false negative.

As shown in Figure 10, we show five testing examples of segmentation results for comparison. In Table 2, we report quantitative evaluation results of the comparative methods using the sensitivity, specificity and dice coefficient. Our method has achieved the highest rate of sensitivity among the comparative methods. It achieves 83.46% sensitivity, and 62.55% dice coefficient. This means that our method can correctly identify most pixels of ROIs in Figure 10F. As shown in Figure 10C–E, the other methods cannot detect many pixels of ROIs. The specificity of our method is 93.77%, which is the lowest among all methods. It means that our method identifies many pixels of the background or noise areas as the ROI.

3.4 | Detail analysis

Figure 11 shows the local-area results of the proposed method and the competitive methods. Figure 11A is the

TABLE 2 The results of different methods on the OCT-ME database

Methods	Sensitivity	Specificity	Dice coefficient
Iteration	43.97	98.23	51.89
Otsu	32.41	98.84	43.55
K-means	34.65	98.65	42.39
Ours	83.46	93.77	62.55

Note: Best results are in bold.

detail of the original image. The intensity of some tissues is low. Figure 11B shows the ground truth of the original sample. One can see that our method achieves relatively high performance in the low-intensity areas. The competitive methods get low performance. The reasons are as follows: (a) The areas of tissues are far less than the background areas, which will result in the imbalanced ratio between the ROI and background. For the global threshold methods, such as Otsu [19, 22], they cannot exactly count the numbers of ROI and background respectively. (b) The pixels intensity of some tissues is very close to the background. The fixed threshold methods can obtain the threshold of the entire image. However, they cannot adaptively calculate these low-intensity areas.

TABLE 3 Three metric rate (%) of different PCA numbers

PCA Number	Sensitivity	Specificity	Dice Coefficient
4	93.49	91.94	62.41
6	90.61	92.74	63.31
8	89.31	93.31	64.36
10	85.64	94.15	65.07
12	86.13	93.98	64.91
14	88.43	93.47	64.44
16	85.28	94.13	64.95
18	93.61	91.32	60.81
20	92.41	92.49	63.46
22	92.22	92.41	63.14
24	92.41	92.23	62.71
26	94.37	90.97	60.25
28	85.74	93.96	64.63
30	90.49	92.68	63.09

Note: Best results are in bold.

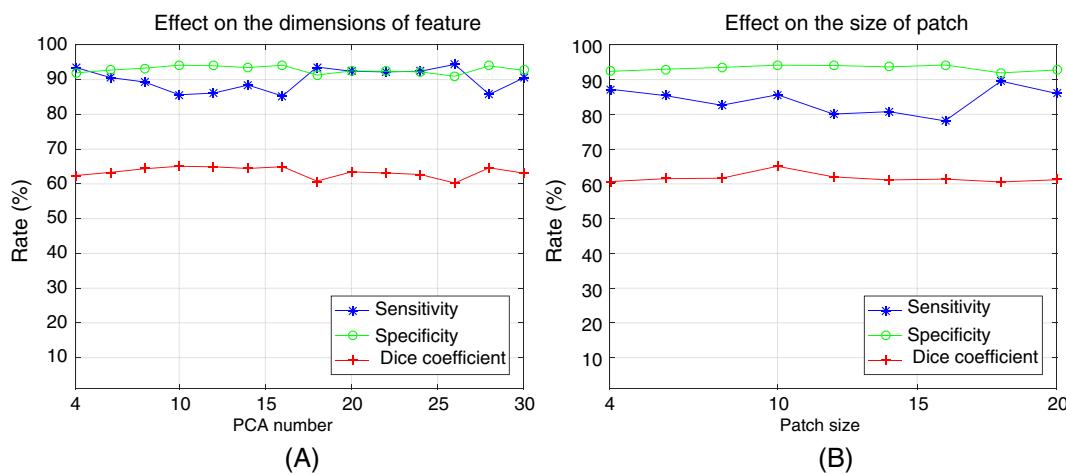


FIGURE 11 The details of the original sample, the ground truth and the results with different methods. (A) The original sample; (B) the ground-truth image. (C–E) The results by the Iteration, Otsu and K-means methods. (F) The result by our specific-patch SVM

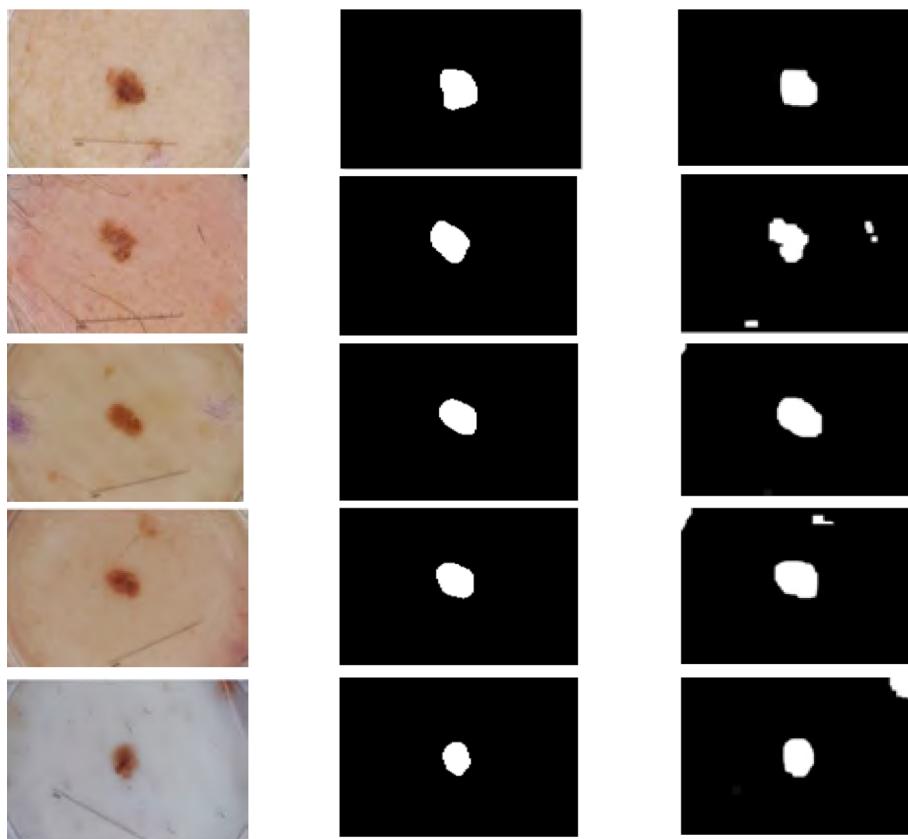


FIGURE 12 The rate curves of three metrics with PCA numbers and patch sizes

TABLE 4 The rate (%) of different patch sizes

Patch Size	Sensitivity	Specificity	Dice coefficient
4 × 4	87.25	92.37	60.66
6 × 6	85.38	92.99	61.51
8 × 8	82.63	93.51	61.64
10 × 10	85.64	94.15	65.07
12 × 12	78.31	94.11	61.48
14 × 14	80.78	93.66	61.13
16 × 16	78.12	94.01	61.38
18 × 18	89.57	91.93	62.55
20 × 20	86.01	92.78	61.21

Note: Best results are in bold.

K-means can learn the different clusters for local areas by setting K as different values. However, the binarized image only includes two types of pixels, that is, white pixels and black pixels. When $K = C$ (C means the number of clusters), we classify the pixels in cluster with $K \geq C$ as the white pixels (ROI) while the pixels in cluster with $K < C$ belong to the background. In this work, $K = 3$. As shown in Figure 11E, only a few pixels of ROI are caught with K-means. As shown in Figures 11F and 10F, most pixels of tissues can be caught by our specific-patch SVM.

3.5 | Effect on feature dimension

In our framework, there exist several hyper parameters, such as, the number of PCA, the patch size and morphology parameters. In this section, we analyze the dimensions of features. In this work, PCA is conducted on the reduction dimension of features. The number of PCA ranges from 4 to 30 while fixing the other parameters. The quantitative results on the PCA number are summarized in Table 3 and Figure 12A. One can see that the sensitivity achieves 94.37% when the number of PCA is 26. The specificity and dice coefficient achieves 94.15% and 65.07% respectively when the number of PCA is 10. In this work, we select the number of PCA as 10.

3.6 | Effect on patch size

For few-shot scene, we extract patches from few training samples and label every patch as the background or ROI. In this section, the patch sizes range from 4×4 (pixels) to 20×20 (pixels) while the other parameters are fixed. The Gabor filters cannot work when the size is less than 4×4 pixels. Some details of organs will be lost when the size is bigger than 20×20 pixels. The quantitative results on the patch sizes are shown in Table 4 and Figure 12B. When the patch size is 18×18 pixels, the sensitivity

TABLE 5 The dice coefficient rate (%) of different sizes of morphology operator

Dilation	Erosion	Dice coefficient
20	10	56.56
20	15	62.13
30	10	49.29
30	15	54.24
30	20	59.37
30	25	63.42
40	10	43.71
40	15	47.77
40	20	52.21
40	25	56.95
40	30	60.95
40	35	62.58
50	10	39.28
50	15	42.61
50	20	46.33
50	25	50.39
50	30	54.79
50	35	58.78
50	40	60.92
50	45	60.97

Note: Best results are in bold.

TABLE 6 The results of different methods on the skin database

Methods	Sensitivity	Specificity	Dice coefficient
Iteration	69.35	88.68	63.53
Otsu	66.73	88.91	63.61
K-means	57.82	86.61	37.83
UNet	94.79	92.63	84.76
R2U-Net	94.14	93.13	86.08
Ours	92.14	96.61	75.71

Note: Best results are in bold.

achieves 89.57%. When the patch size is 10×10 pixels, the specificity and dice coefficient achieve the highest. Therefore, we select the patch size as 10×10 pixels.

3.7 | Morphology analysis

There usually exist some disconnected region and noises in the output image generated by the trained model. Dilatation/erosion is a morphological transformation operator used to increase/decrease the size of the ROI in an image.

To dilate/erode an image, we define a kernel matrix which is made of ones and slide the kernel through the image. The image size of OCT-ME achieves 2048×2048 pixels. Therefore, the sizes of dilation/erosion are also big. In Table 5, the dice coefficient achieves the highest when the kernels of dilation and erosion are around 30 and 25, respectively. When the kernel sizes of dilation and erosion are equal, there will be more disconnected regions. In the experiments, the kernel sizes of erosion are usually smaller than those of dilation.

3.8 | Performance on the skin database

To further validate the performance of our model, we select one representative database skin cancer, which is taken from the Kaggle competition in 2017 [46, 47]. This database contains 2000 samples in total. It consists of 1250 training samples, 150 validation samples, and 600 testing samples. The size of each sample is 700×900 pixels. The results of different methods are shown in Table 6 and Figure 13. One can see that the sensitivity, specificity and dice coefficient of our model achieve 92.14%, 96.61% and 75.71% respectively. The specificity of ours is better than others. With lots of samples, the deep-learning methods, that is, UNet and R2U-Net, obtain better performance.

3.9 | Deep-learning model analysis

Recently, plenty of deep-learning methods have been presented for image processing and machine vision. Our framework can utilize many kinds of features. In this section, we will utilize the deep-learning features to train the SPSVM model. As shown in Figure 14, we embed the fully connected neural (FCN) network into our framework. The parameters of FCN are shown in Table 7. We extract the patches by the way of specific-patch selection in Figure 3. The input layer contains 10×10 pixels, which is fed into the network. The final layer contains two outputs, which represents the ROI and background. To make the network converge faster, the common standard cross-entropy loss function is used to be identity loss.

For the hyperparameters, we empirically set the learning rate to 0.001 and the epochs to 200. In the training phase, we design a fully connected neural network to extract deep-learning feature of specific patches. Deep features are reduced by PCA. Finally, SVM is trained with the deep-learning features. In the testing phase, firstly, deep-learning features of testing samples are extracted by the trained network FCN, and then reduced by PCA. Finally, the binarization image by the trained SVM model.

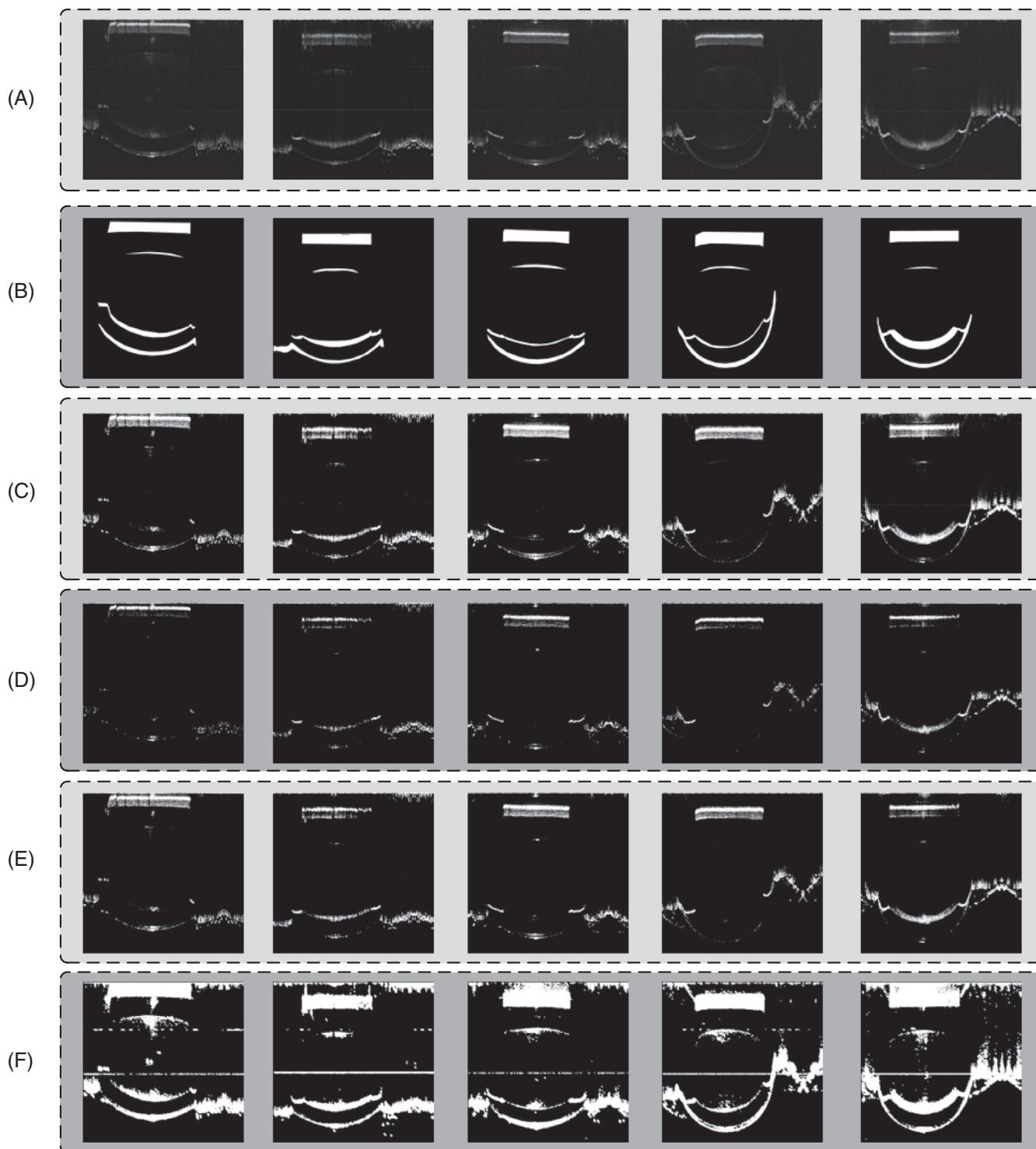


FIGURE 13 The results demonstrate the qualitative assessment of the proposed SPSVM for skin cancer segmentation task. From left to right: the input skin samples, the ground truth, the results of segmentation

3.10 | Performance comparison

To achieve further evaluation of the comparative methods, we utilize the representative evaluation metrics and the popular method UNet.[†] The results of different

methods on the OCT-ME database are shown in Table 8. UNet-*num* notes the model trained by the given number of samples, that is, “UNet-5” means that we train the UNet with five samples. We can see that the sensitivity of our framework with deep-learning features achieves the

FIGURE 14 The flowchart of our specific-patch SVM with deep-learning features. There include the training phase and testing phase

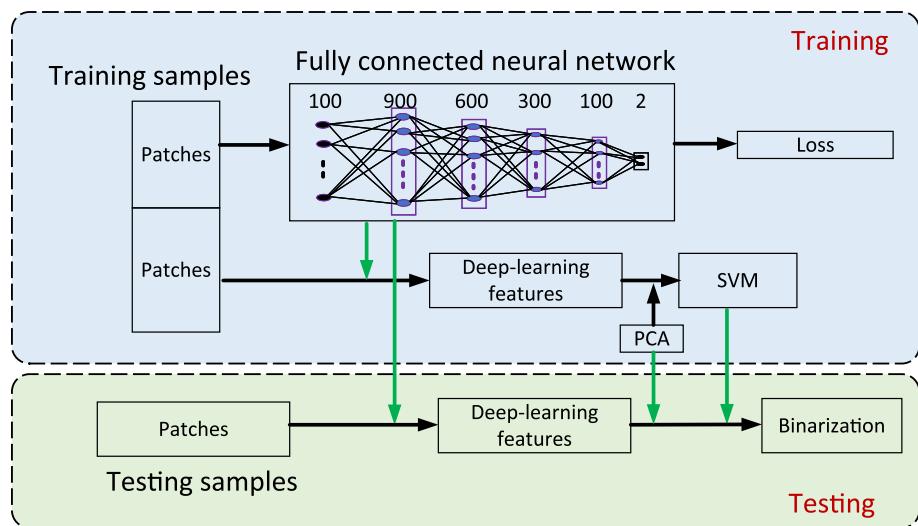


TABLE 7 The architecture of FCN and the parameters for the full connected layers

Layers	Parameters
Input	100
Full connected layer 1	900
Full connected layer 2	600
Full connected layer 3	300
Full connected layer 4	100
Output(softmax)	2

TABLE 8 The results of different methods on the OCT-ME database

Methods	Sensitivity	Specificity	Dice coefficient
Iteration	43.97	98.23	51.89
Otsu	32.41	98.84	43.55
K-means	34.65	98.65	42.39
UNet-5	92.25	97.94	83.69
UNet-10	88.17	98.06	82.17
UNet-20	89.09	98.35	84.61
UNet-30	91.47	98.14	84.47
UNet-45	92.93	97.64	82.58
FCN	91.84	90.96	69.30
Ours+DeepFea	92.96	94.74	71.28

Note: Best results are in bold.

highest. The specificity and dice coefficient of ours are lower than UNet. Three metrics of ours are higher than FCN. It shows that our framework embedding deep-learning features is effective.

4 | CONCLUSIONS

In this paper, we have proposed a novel binarization framework based on specific-patch classification to address the low-intensity OCT image. Specifically, we can extract the features of each patch from testing samples with the learned eigenvector subspace by PCA. We can extract plenty of patches to train the specific-patch SVM model, which can get the adaptive threshold values for the ROI and background. The novel specific-patch framework can provide the new mode to the binarization work in future. It will be suitable for OCT image segmentation and statistics.

We carefully capture an OCT mice eye (OCT-ME) database. Every image is low-intensity. We conduct comprehensive experiments on OCT-ME database. The experimental results demonstrate the effectiveness of the proposed approach.

ACKNOWLEDGMENTS

The authors wish to thank the editors and anonymous referees for their constructive criticism and valuable suggestions. This work was supported by Natural Science Foundation of Shandong Province (No: ZR2020MF105), Guangdong Provincial Key Laboratory of Biomedical Optical Imaging Technology (No: 2020B121201010), the Natural National Science Foundation of China (62175156, 61675134), Science and technology innovation project of Shanghai Science and Technology Commission (19441905800, 22S31903000).

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

ENDNOTES

* <https://mip2019.github.io/spsvm>.

† <https://github.com/Andy-zhujunwen/UNet-ZOO>.

DATA AVAILABILITY STATEMENT

<https://mip2019.github.io/spsvm>

ORCID

Fei Ma  <https://orcid.org/0000-0002-5472-4763>

REFERENCES

- [1] D. Huang, E. A. Swanson, C. P. Lin, J. S. Schuman, W. G. Stinson, W. Chang, M. R. Hee, T. Flotte, K. Gregory, C. A. Puliafito, J. G. Fujimoto, *Science* **1991**, 254, 1178.
- [2] Y. Zhao, Y. Zheng, Y. Liu, J. Yang, Y. Zhao, D. Chen, Y. Wang, *IEEE Trans. Med. Imaging* **2016**, 36, 51.
- [3] N. Eladawi, M. Elmogy, O. Helmy, A. Aboelfetouh, A. M. Riad, H. Sandhu, et al., *Comput. Biol. Med.* **2017**, 89, 150.
- [4] R. Qiu, C.F. Guo, Study of Imaging Performance of Mouse Eyes Based on Optical Coherence Tomography and Ray Tracing Techniques. OPTICS IN HEALTH CARE AND BIOMEDICAL OPTICS X, **2020**.
- [5] X. M. Liu, S. C. Wang, Y. Zhang, D. Liu, W. Hu, *Neurocomputing* **2021**, 452, 576.
- [6] D. H. Lu, M. Heisler, S. Lee, G. W. Ding, E. Navajas, M. V. Sarunic, M. F. Beg, *Med. Image Anal.* **2019**, 54, 100.
- [7] A. Moiseev, L. Snopova, S. Kuznetsov, *J. Biophotonics* **2018**, 11, e201700072.
- [8] P. G. Daneshmand, A. Mehridehnavi, H. Rabbani, *IEEE Trans. Med. Imaging* **2021**, 40(3), 865.
- [9] N. A. Kande, R. Dakhane, A. Dukkipati, P. K. Yalavarthy, *IEEE Trans. Med. Imaging* **2021**, 40(1), 180.
- [10] A. P. Sunija, S. Kar, S. Gayathri, V. P. Gopi, P. Palanisamy, *Comput. Methods Programs Biomed.* **2021**, 200, 105877.
- [11] N. Mehta, P. X. Braun, I. Gendelman, A. Y. Alibhai, N. K. Waheed, *Sci. Rep.* **2020**, 10(1), 15368.
- [12] N. Mehta, *Am. J. Ophthalmol.* **2019**, 205, 54.
- [13] P. X. Zang, L. Q. Gao, T. T. Hormel, J. Wang, Q. S. You, T. S. Hwang, Y. Jia, *I.E.E.E. Trans. Biomed. Eng.* **2021**, 68(6), 1859.
- [14] B. Hassan, S. Y. Qin, T. Hassan, R. S. Ahmed, N. F. Werghi, *IEEE Trans. Instrum. Meas.* **2021**, 70, 1.
- [15] M. Samieinasab, Z. Amini, H. Rabbani, *IEEE Trans. Med. Imaging* **2020**, 39(11), 3475.
- [16] T. Wang, T. Pfeiffer, J. Daemen, F. Mastik, W. Wieser, *IEEE Trans. Med. Imaging* **2020**, 39(5), 1535.
- [17] Q. Zhao, R. Lin, C. Liu, J. Zhao, L. Song, J. Meng, *J. Biophotonics* **2019**, 12, 1.
- [18] B. Y. Tan, J. Chua, V. Amutha, *Opt. Express* **2019**, 10(1), 356.
- [19] N. Otsu, *IEEE Trans. Syst. Man Cybern.* **1979**, 9(1), 62.
- [20] D.J. Liu, J. Yu, Otsu Method and K-means. HIS, **2009**, 344–349.
- [21] H. M. Kim, S. H. Lee, C. Lee, J. W. Ha, Y. Yoon, *EMBC 2014*, 2014, 178.
- [22] J. G. Xing, P. Yang, L. Qing, *Comput. Intell. Neurosci.* **2020**, 5047976(1–5047976), 14.
- [23] S. H. Shaikh, A. K. Maiti, N. Chaki, *ReTIS* **2011**, 24(2), 337.
- [24] A. Dawoud, M. S. Kamel, *IEEE Trans. Image Process.* **2004**, 13(9), 1223.
- [25] A. Dawoud, M. Kamel, *ICIP 2003*, 1, 553.
- [26] S. Mitra, K. C. Santosh, M. K. Naskar, *Int. J. Pattern Recognit. Artif. Intell.* **2021**, 35(4), 2154013:1.
- [27] K. Khurshid, I. Siddiqi, C. Faure, N. Vincent, *DRR* **2009**, 7247, 1.
- [28] J. García, B. Crawford, R. Soto, C. Castro, F. Paredes, *Appl. Intell.* **2018**, 48(2), 357.
- [29] E. T. Zemouri, Y. C. Chibani, *J. Electronic Imaging* **2019**, 28(4), 043021.
- [30] O. Ronneberger, P. Fischer, T. Brox, *MICCAI* **2015**, 9351 (3), 234.
- [31] H. M. Huang, L. F. Lin, R. F. Tong, H. J. Hu, Q. W. Zhang, Y. T. Iwamoto, X. Han, Y. Chen, J. Wu, *ICASSP 2020*, 90, 1055.
- [32] X. Y. Jing, Q. Liu, F. Wu, B. Xu, Y. P. Zhu, S. C. Chen, *IJCAI 2015*, 2015, 2255.
- [33] W. K. Kong, D. Zhang, *International Conference on Pattern Recognition. IEEE* **2004**, 2004, 520.
- [34] S. Verma, S. Chandran, *Int. Arab J. Inf. Technol.* **2019**, 16(1), 23.
- [35] J. Y. Choi, B. Lee, *IEEE Trans. Image Process.* **2020**, 29, 3270.
- [36] C. Y. Low, A. B. Teoh, C. J. Ng, *IEEE Trans. Circuits Syst. Video Technol.* **2019**, 29(1), 115.
- [37] D. D. Chen, Y. J. Tian, X. H. Liu, *Pattern Recognit.* **2016**, 60, 296.
- [38] G. Valentini, M. Muselli, F. Ruffino, *Neurocomputing* **2004**, 56, 461.
- [39] A. S. Reddy, P. C. Reddy, *Soft Comput.* **2021**, 25(5), 4135.
- [40] H. Faris, M. Habib, M. Faris, M. Alomari, A. Alomari, *J. Biomed. Informatics* **2020**, 109, 103525.
- [41] X. C. Tang, Z. W. Ma, Q. Hu, W. Tang, *I.E.E.E. Trans. Biomed. Eng.* **2020**, 67(4), 978.
- [42] S. Gupta, V. Thenkanidiyoor, A. D. Dileep, *ICONIP 2016*, 9950 (4), 321.
- [43] C. J. C. Burges, *Data Min. Knowl. Discov* **1998**, 2(2), 121.
- [44] R. Achanta, S. Hemami, F. Estrada, S. Susstrunk, Frequency-tuned salient region detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, **2009**, 1597–1604.
- [45] S. Jadon, A survey of loss functions for semantic segmentation, In: IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology, **2020**.
- [46] <https://challenge2017.isic-archive.com>.
- [47] M. Z. Alom, M. Hasan, C. Yakopcic, T. M. Taha, V. K. Asari, Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation. IEEE National Aerospace and Electronics Conference, **2018**.

How to cite this article: F. Ma, C. Dai, J. Meng, Y. Li, J. Zhao, Y. Zhang, S. Wang, X. Zhang, R. Cheng, *J. Biophotonics* **2022**, e202100336. <https://doi.org/10.1002/jbio.202100336>