Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

Decoding Retinal Patterns through Permutation Importance Analysis

Sumit Mukherjee[1], Ranjit Ghoshal[2], Bibhas Chandra Dhara[3]

[1]Research Scholar, Information Technology, Jadavpur University,
Kolkata : 700098, India
[2]Associate Professor, Information Technology, St. Thomas' College of Engineering and Technology,
Kolkata : 700023, India
[3]Professor, Information Technology, Jadavpur University,
Kolkata : 700098, India

E-mail: [1]sum.mukherjee@gmail.com, [2]ranjit.ghoshal.stcet@gmail.com, [3]bcdhara@gmail.com

Abstract

Vascular map within the inner surface of retina provides insights about ophthalmic abnormalities or early signs of different eye diseases. Devising a straight forward strategy to analyze the vascular map is quite challenging, given the complex and delicate nature of these vessels, as well as the high level of noise in the data. This article outlines a new method for analyzing vascular map by extracting various features and allocating weightage to these features. The most influential features in identifying the vascular map have been determined using a method called Permutation Importance. Final model is built using these highly qualified features. The performance of the model has been validated using internationally recognized standards and found to be among the top performers in its category.

Keywords: Retinal disorders; Permutation Importance; Vascular Map; Feature extraction

1. Introduction

Ophthalmoscopy examination of the eyes equips the eye specialists to check the fundus and other components within the eye. A crucial aspect of this examination involves assessing the structural orientation of retinal vessels or vascular map, which can provide valuable diagnostic information for conditions such as diabetic retinopathy, hypertension, and glaucoma [1, 2]. To obtain relevant information about eye-related diseases, it is highly recommended to extract vascular maps from images of the retinal fundus. While there are tools to aid ophthalmologists in manually marking vessel maps, there is no system to automatically identify these vascular

DOI: https://doi.org/10.36548/jiip.2024.4.006
Received: 5.10.2024, received in revised form: 29.11.2024, accepted: 22.12.2024, published: 23.01.2025

maps, which makes it tough for medical professionals to take informed decisions in a short period of time.

Identifying vessel maps is tough because vascular structures differs person to person, and image noise often confuses algorithms. Researchers are engaged in developing complex methods [4], like morphological operations [12, 13] and deep learning [6, 7], to extract vessel maps from eye images. Each method has pros and cons. Figure. 1 presents fundus images and vascular maps annotated by human experts, with the images sourced from the DRIVE dataset [9].

(a) (b)

(c) (d)

Figure 1: (a) and (b) Sample fundus images, (b) and (d) corresponding vascular maps.

Over the years researchers have suggested various techniques to enhance the recognition and segmentation of vascular maps in fundus images, aiming to improve diagnosis and treatment of eye-related diseases. Proposed techniques encompass a wide range of approaches, ranging from traditional image processing methods to machine learning algorithms, and deep learning architectures. Marin et al. [3] introduced a technique that utilizes grayscale and moment invariants for classifying functions related to vessels in fundus image. Ricci et al. [8] suggested a way using support vector machines (SVMs) trained on features extracted by orthogonal line operators to classify pixels into vascular and non-vascular categories. Several studies have explored directional map-based methods for vessel segmentation. Frucci et al. [11] proposed novel approaches based on directional maps, which offer robustness in capturing vessel structures with varying orientations. Roychowdhury et al. [14] investigated the application of morphological processing and ML learning approaches for vessel extraction, respectively. Mendonca et al. [15] introduced an idea founded on centerline identification using morphological (mathematical) operations, offering a systematic approach to vessel segmentation. Oliveira et al. utilized convolutional neural networks (CNNs) for vessel segmentation, achieving impressive results. Guo et al. proposed a deeply trained multi-scale multilevel CNN with short links for vessel segmentation, showcasing the potential of deep learning models with hierarchical architectures. Despite the advancements in deep learning-based methods, challenges such as computational complexity and data requirements

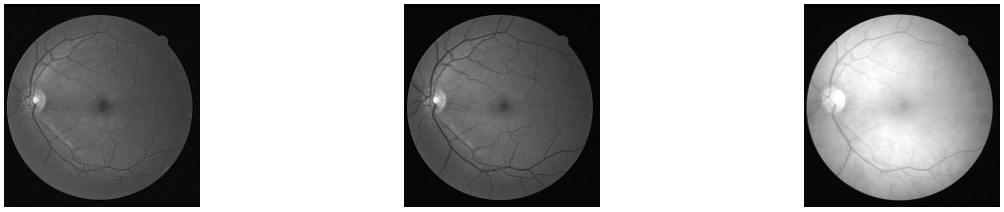
still exist. Powerful system design and extensive data collection are often necessary for training these models effectively. In the proposed methodology, a comprehensive array of 41 distinct filters was applied to the images, encompassing 32 variations of Gabor filters, along with Canny Edge, Roberts Edge, Sobel Edge, Scharr Edge, Prewitt Edge, Sigma 3 and 7 Gaussian filters, Sigma 3 and 7 Median filters, and Variance filters. Subsequently, the original and ground truth images were augmented into one-dimensional matrices, forming a comprehensive data frame alongside the 41 filtered images. This facilitated the training, testing, and refinement of N models, each corresponding to an N-image dataset, within their respective data frames, utilizing a leave-one-out cross-validation approach to generate a base model for each input image. Following this , each of the 41 features for an input image underwent random swapping with other input features, yielding a set comprising 41 features with one feature swapped, along with the ground truth image. In the next step, each set was tested on all models, and the weightage of features was calculated based on the deviation in the outcome. These feature weightage for each model were combined to find the average weightage of features. Best 30% of features is considered to construct a final model. This methodology is executed on the DRIVE dataset [9]. The subsequent sections delineate the structural flow of this study: Section 2 elaborates the different methodologies of applying filters, while Section 3 expounds on feature swapping. The operational flow of the proposed technique is elucidated in Section 4, which also delves into the selection and amalgamation process of the top 30% features and outlines strategies to streamline functionality for enhanced test time efficiency without compromising performance. Section 5 evaluates the performance of the proposed method, followed by the concluding remarks in Section 6.

2. Feature Provisioning

The proposed method for detecting vascular maps is divided into several basic steps. The method first extracts the green channel of the input fundus image, as it provides the best contrast between vascular and non-vascular components (Fig. 2, image used from the [9]), a technique widely adopted in nearly all state-of-the-art approaches. The process involves passing fundus images through various filters, kernels, and processing steps to extract different features. In the life cycle of the proposed scheme, these features undergo further refinement and some of them qualify to be a part of the final model by the *Permutation Importance* method. Features taken into account are listed below. The Figure 2 shows the fundus RGB images

2.1 32x Gabor Filters

Gabor filters are useful for analysis of texture, detection of edges, and extracting features, which can be utilized to train machine learning algorithms. As bandpass filters, Gabor kernels allow certain frequencies while rejecting others. The Gabor function is mathematically represented



(a) (b) (c)

Figure 2: (a), (b), and (c) correspond to the red, green, and blue channels of the RGB fundus image, respectively.

in equation (1), where a and b define the kernel size, σ is the standard deviation of the Gaussian envelope, θ is the orientation of the stripes, λ is the wavelength of the sinusoidal factor, γ is the spatial aspect ratio, and ψ is the phase offset. For the proposed scheme, 32 different Gabor kernels are used [5]. Examples of some kernels applied to images from the DRIVE dataset [9] are shown in Figure. 3.

$$g(a,b,\sigma,\theta,\lambda,\gamma,\psi) = \frac{-(a^2+\gamma^2 b^2)}{(2\sigma^2)} \exp[i(2\pi(\frac{a}{b})+\psi)] \tag{1}$$

2.2 Various Edge Detectors

Edge detectors separates out areas with abrupt intensity or colour changes. High values refer steep transitions, while low values signify gradual changes. The following Figure 4 shows Points of Interest (PoI), representing pixels and their surrounding neighborhood as a matrix.

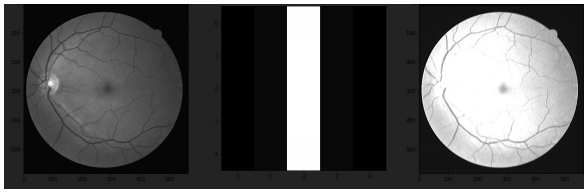
Roberts Masks detect diagonal edges, improving upon ordinary masks that only detect horizontal and vertical edges (Figure. 4). Despite their smaller 2x2 size, they have limited neighborhood coverage.

To overcome these limitations, Sobel and Prewitt masks are used (see Figure. 5). The Sobel mask uses two 3x3 kernels to detect edges in both horizontal and vertical directions by convolving with the image to compute the intensity gradient at each pixel.

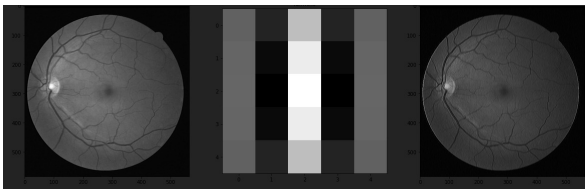
The Prewitt mask is similar to the Sobel mask, with slight differences. Unlike Sobel, which gives more weight to points closer to (x,y), Prewitt assigns symmetric values around the center. The figure illustrates the Prewitt x and y operator values.

The Scharr serves as a filtering technique utilized for edge detection and feature enhancement through the 1st derivative. Typically employed to ascertain gradients along the x-axis ($dx = 1$, $dy = 0$) and y-axis ($dx = 0$, $dy = 1$) independently, its functionality closely resembles that of the Sobel filter.

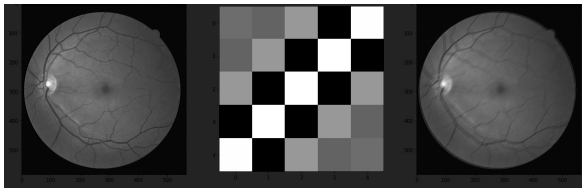
The Canny Edge detection technique offers distinct advantages over other methods due to its unique features. Notably, it effectively suppresses noise while accurately identifying edges in images. However, it is important to note that Canny Edge operates exclusively on grayscale images; therefore, if the input is in RGB format, it must first be converted to grayscale. The



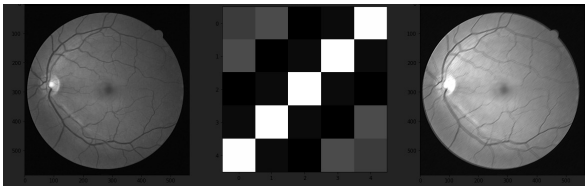
(a)



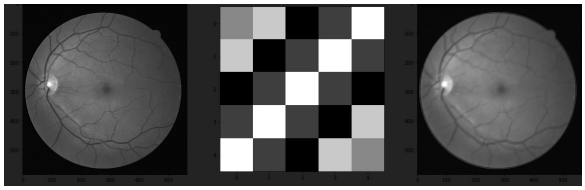
(b)



(c)



(d)



(e)

Figure 3: (a), (b), (c), (d), and (e) represent channel green of fundus images, associated Gabor kernels and images processed by Gabor kernels.

1	0
0	-1

(a)

0	1
-1	0

(b)

Figure 4: (a) Robert Mask for first diagonal identification and (b) Robert Mask for second diagonal identification

-1	0	1
-2	0	2
-1	0	1

(a)

1	2	1
0	0	0
-1	-2	-1

(b)

-1	0	1
-1	0	1
-1	0	1

(c)

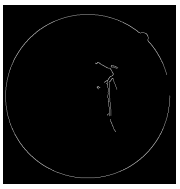
1	1	1
0	0	0
-1	-1	-1

(d)

Figure 5: (a) and (b) display the Sobel kernels (Vertical and Horizontal), respectively, while (c) and (d) show the Prewitt kernels (Vertical and Horizontal), respectively.

application of Canny Edge involves multiple steps to achieve optimal results. Initially, the image undergoes Gaussian Blur processing, which serves to reduce noise and smooth the input image, facilitating subsequent operations. Following this, the intensity gradient of the image is calculated. These preparatory steps pave the way for the core Canny Edge transformation process, as illustrated in Figure. 6 (input image sourced from DRIVE dataset [9]).

(a)



(b)

Figure 6: (a) Fundus image and (b) outcome after applying the Canny edge kernel.

2.3 Blurring Methods

Blur filters are commonly used in image processing to soften the appearance of pixels or entire images, making them particularly useful for retouching purposes. These filters work by averaging the colour values of neighboring pixels, resulting in a smoother transition between adjacent areas. Gaussian Blur, a specific type of blur filter, is achieved by applying a Gaussian function to the image. This technique produces a smooth, translucent-like blur effect, akin to viewing an image through a frosted glass screen. The representation of Gaussian Blur is typically denoted as follows.

$$Y[m,n] = MEDIAN\ X[i,j],\ (i,j) \in W \tag{2}$$

Where $Y[m,n]$ represents the output pixel value at position (m,n), $X[i,j]$ represents the input pixel values within the filter window W, and W defines the neighborhood of pixels considered

for the median calculation.

The median filters are non-linear digital filter in nature and extensively employed for noise removal, particularly in image processing. This noise reduction step is crucial as it enhances the effectiveness of subsequent analysis techniques. Primarily used to eliminate salt and pepper noise from image set and mitigate their irregularities, the median filter offers distinct advantages over other methods. While neighborhood averaging techniques may blur sudden changes like line features and sharp edges along with noise, the median filter excels in preserving such details while effectively removing isolated out-of-range noise. By analyzing the median values within a window, this filter can discern between genuine image features and aberrant noise, thereby contributing to improved image quality and analysis outcomes.

3. Feature Permutation Importance

Permutation Importance assesses the significance of features in a trained model by measuring the increase in prediction error when the values of a single feature are randomly shuffled. A characteristic is deemed significant if changing it increases model error since the model depends on it for prediction. On the other hand, it suggests the feature was disregarded during prediction and is thought to be irrelevant if rearranging the feature does not increase model error. This technique is used on a validation set after training and shuffles the data points of one characteristic while maintaining the integrity of the other data. Reduced accuracy following shuffling suggests the feature's importance, particularly if the model was heavily reliant on it.

For each feature i in $(1,\ldots,F)$, and for each repetition r in $(1,\ldots,R)$, the process is as follows:

1. Permute or shuffle column i of the feature matrix X to create a permuted dataset X'.
2. Use the trained model M to predict labels based on the permuted data, obtaining $\hat{y}'$.
3. Calculate the error E' based on the predictions $\hat{y}'$ and the actual labels y.
4. Compute the difference in errors: $|E - E'|$.

Finally, the Feature Importance for feature i is calculated as the average of these differences over all repetitions:

$$\text{Feature Importance}_i = \frac{1}{R}\sum_{r=1}^{R} |E - E'_r|$$

Where M is Trained model, X is Feature Matrix, y represents Labels, and $E(y,M)$ is Error function. E'_r represents the error obtained from the permuted data in repetition r. The higher the value of Feature Importance, the more important the feature is considered.

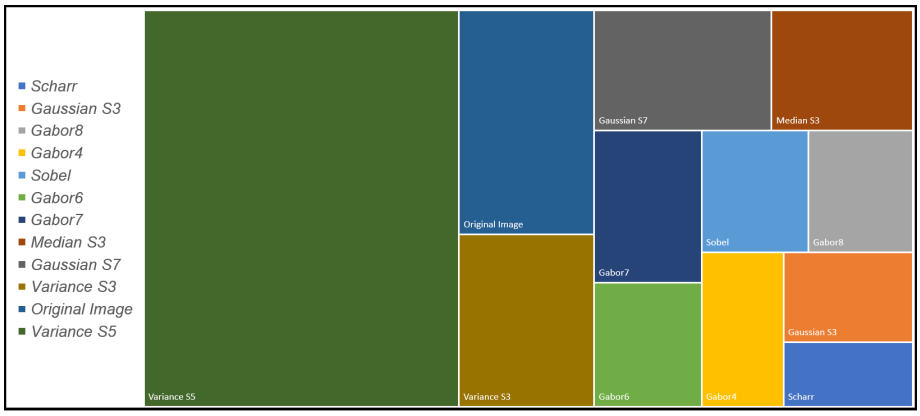


Figure 7: Top 30% Features based on Permutation Importance

4. Workflow of the Algorithm

The algorithm begins by loading a collection of 20 images from the DRIVE dataset and their corresponding ground truths. A variety of kernels/filters, as described in Section 2 above, have been applied to extract 41 distinct features from each of these images. They are then arranged into vectors for each image to ensure smooth processing. In the next step, the algorithm builds a base model for each image using the extracted features, the image itself, and its ground truths. The generated base model is further tested and calibrated with the rest of the cross-sectional data of the other 19 images. In the subsequent step, a feature swapping procedure is deployed, where one feature is exchanged for the input image, and the impact on model accuracy is assessed as mentioned in the section 3. The algorithm calculates the average impact of each feature across all images, determining the most crucial features for model performance. Features with negligible impact are discarded, and the remaining important features are used to build a final model. This final model undergoes testing to validate its efficacy. The steps are elaborated in the algorithm below more precisely.

The algorithm limits the reduction of features to 30%. From the DRIVE Dataset, the derived permutation importance method identified 12 features that exert the greatest influence on the model's outcome. Figure. 7 depicts these 12 features along with their corresponding levels of influence. For running the proposed method, the framework used is Python 3.8 on Ubuntu 22.04.

5. Analysis and Evaluation of Results

The DRIVE dataset[1] of retinal fundus images is made available to the public for the endeavor of advancing eye anomaly screening. Twenty training and test images with actual result are included in this dataset. As a part of the Dutch Diabetic Retinopathy Screening Program, the picture was taken. Images of 400 diabetes individuals, ranging in age from 25 to 90, were

[1]DRIVE dataset is available at `http://www.isi.uu.nl/Research/Databases/DRIVE/download.php`

Algorithm Feature Permutation Importance

Load N image array **IMG** and N ground truth array **GT**
for $i = 1$ to N **do**
  Extract x features for **IMG**$[i]$ using different kernels and filters
  for $j = 1$ to 41 **do**
    Push features in another array **FEATURE**$[i][j]$
  end for
end for
for $i = 1$ to N **do**
  Build a base model, say **Model**(i), with **FEATURE**$[i][j]$, **IMG**$[i]$, and **GT**$[i]$
  for $k = 1$ to N excluding $k = i$ **do**
    Train and test the model using **FEATURE**$[k][j]$, **IMG**$[k]$, and **GT**$[k]$
    Tuned model is stored as **Model**(i)
  end for
end for
for $i = 1$ to N **do**
  for $j = 1$ to $(x - i)$ where $i < x$ **do**
    for $l = 1$ to N **do**
      Model(i) run by swapping a feature **FEATURE**$[i][j]$ for all the **IMG**(l)
      Store the result in **PermutationResult**$[i]$
      Calculate accuracy of **PermutationResult**$[i]$ with **GT**$[i]$ and store in array **Accuracy**(i)
    end for
    Store the average of (**Accuracy**(i) + ... + **Accuracy**(N))$/N$ to **FEATURE_IMPACT**(j)
  end for
  Discard the **FEATURE_IMPACT**(x) having the highest value (i.e., Feature swapping has no impact in the outcome)
  Exit the process if more than 30% features are discarded
end for
Mark the 30% un-discarded features from **FEATURE_IMPACT**(x)
Build a Final Model using these features
Evaluation of **Final Model**

included. Of these, 40 images were chosen at random, and 7 of them displayed mild to early diabetic retinopathy symptoms, while the other 33 showed no symptoms at all. This set of 40 images is divided into two groups, testing and training, each group consists of twenty images for carrying out the validation of the proposed mechanism.

Three samples from the dataset, upon which the proposed method is applied, are depicted in the Figure. 8.

Figure 8: Sample fundus from DRIVE image set [9]

The respective human expert segmentation are depicted in Figure. 9. One important 1 to note that this segmentation slightly varies from expert to expert.

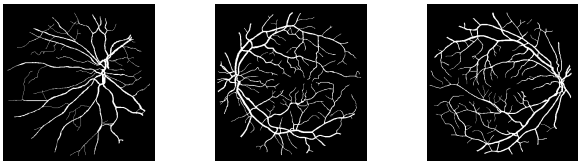


Figure 9: Ground truth images for fundus images shown in Fig. 8.

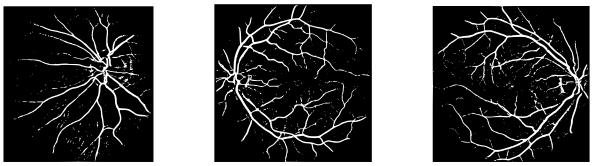


Figure 10: Output images obtained applying the suggested technique on image set shown in Fig. 8.

In evaluating the effectiveness of the proposed approach, key metrics: sensitivity, specificity, and accuracy, have been used (refer the equations 3,4, and 5). These metrics are fundamental in assessing the method's performance across various scenarios. They are determined by pixel classifications: True Positives (TP), which are correctly predicted positive pixels, and True Negatives (TN), which are correctly predicted negative pixels. False Positives (FP) occur when negative pixels are incorrectly classified as positive, while False Negatives (FN) represent positive pixels that were wrongly classified as negative. The formulas for these key metrics are provided in Figure. 11.

$$\text{Sensitivity} = \frac{TP}{TP+FN} \tag{3}$$

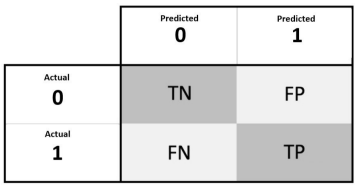


Figure 11: Pixel Class Definitions

Table 1: Comparison with Other Approaches

References	Sensitivity	Specificity	Accuracy
Staal et al. (2004) [9]	0.7194	0.9773	0.9442
Soares eta al. (2006) [10]	0.7332	0.9782	0.9466
Martinez-Perez et al. (2007) [18]	0.7246	0.9655	0.9344
Zhang et al. (2010) [19]	0.7125	0.9725	0.9385
Martin et al. (2011) [3]	0.7067	0.9801	0.9452
Miri et al. (2011) [20]	0.7155	0.9765	0.9435
Singh et al. (2014) [21]	0.7138	0.9801	0.9460
Fu et. al (2016) [22]	0.7603	N.A	0.9523
Oliveira et. al (2017) [16]	0.7810	0.9800	0.9543
Guo et. al (FS-DSN) (2018) [17]	0.7756	0.9802	0.9542
Guo et. al (S-DSN) (2018) [17]	0.7893	0.9789	0.9547
Fakhouri et. al (2024) [23]	0.9961	0.8048	0.9629
Proposed Method	**0.6441**	**0.9880**	**0.9579**

$$\text{Specificity} = \frac{TN}{TN+FP} \tag{4}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

These class attributes are formulated to measure the similarity and non-similarity between ground truth and proposed method's outcome. The proposed method yielded: TP=253, TN=4053, FP=49, FN=140, resulting in Sensitivity=0.6441, Specificity=0.9880, Accuracy=0.9579. On comparing (shown in Table 1), it shows that suggested method to extract vascular map from retinal fundus image has superior accuracy and specificity to most of the other methods considered as milestone in this domain.

6. Conclusion and Future Scope

The proposed approach in this article offers a comprehensive approach to feature analysis and model optimization to emphasize the vascular map in Fundus image. By systematically evaluating feature importance and automating feature selection, it streamlines the model

development process. The method works best when there is a very minimal amount of data is available, and using deep learning techniques cannot guarantee accuracy. However the result come of the proposed techniques lacks accuracy for some of the Fundus images. Furthermore improvement in this scheme can be achieve by improvising the techniques, including additional steps in the pre-processesed and post-processing. The suggested method is a potential candidate for inclusion in pre-screening programs for the clinical prognosis of ocular disorders due to its effectiveness, robustness, and user-friendliness.

References

[1] Franklin, S. W., & Rajan, S. E. (2014). Computerized screening of diabetic retinopathy employing blood vessel segmentation in retinal images. *Biocybernetics and Biomedical Engineering, 34*(2), 117–124.

[2] Bourne, R., Flaxman, S., Braithwaite, T., Cicinelli, M., Das, A., Jonas, J., & Vision Loss Expert Group. (2017). Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: A systematic review and meta-analysis. *The Lancet Global Health, 5*(9), e888–e897.

[3] Marin, D., Aquino, A., Gegndez-Arias, M. E., & Bravo, J. M. (2011). A new unsupervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features. *IEEE Transactions on Medical Imaging, 30*(1), 146–158.

[4] Li, X., Li, C., Rahaman, M. M., et al. (2022). A comprehensive review of computer-aided whole-slide image analysis: From datasets to feature extraction, segmentation, classification and detection approaches. *Artificial Intelligence Review, 55*, 4809–4878. https://doi.org/10.1007/s10462-021-10121-0

[5] Mukherjee, S., Ghoshal, R., & Dhara, B. C. (2022). Features selection for vessel extraction inspired by survival of the fittest method. *Computational Intelligence in Pattern Recognition. Lecture Notes in Networks and Systems, 480*, 559–568. Springer, Singapore. https://doi.org/10.1007/978-981-19-3089-8_36

[6] Feng, Z., Yang, J., & Yao, L. (2017). Patch-based fully convolutional neural network with skip connections for retinal blood vessel segmentation. *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, 1742–1746. https://doi.org/10.1109/ICIP.2017.8296580

[7] Zhang, J., Li, C., Yin, Y., Zhang, J., & Grzegorzek, M. (2023). Applications of artificial neural networks in microorganism image analysis: A comprehensive review from conventional multilayer perceptron to popular convolutional neural network and potential visual transformer. *Artificial Intelligence Review, 56*, 1013–1070.

[8] Ricci, E., & Perfetti, R. (2007). Retinal blood vessel segmentation using line operators and support vector classification. *IEEE Transactions on Medical Imaging, 26*(10), 1357–1365.

[9] Staal, J. J., Abramoff, M. D., Niemeijer, M., Viergever, M. A., & van Ginneken, B. (2004). Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging, 23*(4), 501–509.

[10] Soares, J. V. B., Leandro, J. J. G., et al. (2006). Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification. *IEEE Transactions on Medical Imaging, 25*(9), 1214–1222.

[11] Frucci, M., Riccio, D., Sanniti di Baja, G., & Serino, L. (2016). SEVERE: Segmenting vessels in retina images. *Pattern Recognition Letters, 82*, 162–169.

[12] Zana, F., & Klein, J. C. (2001). Segmentation of vessel-like patterns using mathematical morphology and curvature evaluation. *IEEE Transactions on Image Processing, 10*(7), 1010–1019.

[13] Imani, E., Javidi, M., & Pourreza, H. R. (2015). Improvement of retinal blood vessel detection using morphological component analysis. *Computer Methods and Programs in Biomedicine, 118*(3), 263–279.

[14] Roychowdhury, S., Koozekanani, D., et al. (2015). Blood vessel segmentation of fundus images by major vessel extraction and subimage classification. *IEEE Journal of Biomedical and Health Informatics, 19*(3), 1118–1128.

[15] Mendonca, A. M., & Campilho, A. (2006). Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. *IEEE Transactions on Medical Imaging, 25*(9), 1200–1213.

[16] Oliveira, A., Pereira, S., & Silva, C. A. (2017). Augmenting data when training a CNN for retinal vessel segmentation: How to warp? *2017 IEEE 5th Portuguese Meeting on Bioengineering (ENBENG)*, Coimbra, 1–4. https://doi.org/10.1109/ENBENG.2017.7889443

[17] Guo, S., Gao, Y., Wang, K., & Li, T. (2018). Deeply supervised neural network with short connections for retinal vessel segmentation. *arXiv:1803.03963v1 [cs.CV]*.

[18] Martinez-Perez, M. E., Hughes, A. D., Thom, S. A., Bharath, A. A., & Parker, K. H. (2007). Segmentation of blood vessels from red-free and fluorescein retinal images. *Medical Image Analysis, 11*(1), 47–61.

[19] Zhang, B., Zhang, L., Zhang, L., & Karray, F. (2010). Retinal vessel extraction by matched filter with first-order derivative of Gaussian. *Computerized Biology and Medicine, 40*(4), 438–445.

[20] Miri, M. S., & Mahloojifar, A. (2011). Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction. *IEEE Transactions on Biomedical Engineering, 58*(5), 1183–1192.

[21] Singh, D., Dharmveer, & Singh, B. (2014). A new morphology-based approach for blood vessel segmentation in retinal images. *2014 Annual IEEE India Conference (INDICON)*, 1–6.

[22] Fu, H., Xu, Y., Lin, S., Wong, D. W. K., & Liu, J. (2016). Deepvessel: Retinal vessel segmentation via deep learning and conditional random field. *Medical Image Computing and Computer-Assisted Intervention, 1*(1), 132–139.

[23] Fakhouri, H. N., Alawadi, S., Awaysheh, F. M., Alkhabbas, F., & Zraqou, J. (2024). A cognitive deep learning approach for medical image processing. *Scientific Reports, 14*(1), 4539. https://doi.org/10.1038/s41598-024-55061-1

Author's Biography

Sumit Mukherjee is currently a Research Scholar in the Department of Information Technology at Jadavpur University, India. He received his B.E. degree in Computer Science and Engineering from the University of Burdwan in 2003 and later pursued an M.Tech degree in Computer Science and Engineering from West Bengal University of Technology in 2006. He has over a decade of professional experience, having worked as a Software Professional at Tata Consultancy Services (TCS). His research interests encompass Medical Image Processing, Computer Vision, Machine Learning, Unlearning, and Explainable AI.

Ranjit Ghoshal is an Associate Professor in the Department of Information Technology at St. Thomas's College of Engineering & Technology, Kolkata, India. He

obtained his M.Tech. degree in Computer Science and Engineering from Kalyani Government Engineering College in 2006 and his Ph.D. from Jadavpur University in 2017. His research focuses on Machine Learning and its applications in image, text, and pattern analysis.

Bibhas Chandra Dhara is a Professor in the Department of Information Technology at Jadavpur University, India. He earned his B.Sc. degree (Hons.) in Mathematics and B.Tech. in Computer Science and Engineering from the University of Calcutta in 1997 and 2000, respectively. He further pursued M.Tech. and Ph.D. degrees in Computer Science at the Indian Statistical Institute, completing them in 2002 and 2008. With over 100 peer-reviewed publications, his research areas include Image Processing, Computer Vision, and Information Security.