

A novel fern-like lines detection using a hybrid of pre-trained convolutional neural network model and Frangi filter

Heri Pratikno^{1,2}, Mohd Zamri Ibrahim¹, Jusak^{2,3}

¹Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang, Pahang, Malaysia

²Faculty of Informatics Technology, Department of Computer Engineering, Universitas Dinamika, Surabaya, Indonesia

³School of Science and Technology, James Cook University, Singapore

Article Info

Article history:

Received Sep 02, 2021

Revised Apr 05, 2022

Accepted Apr 13, 2022

Keywords:

Deep learning

Fern-like lines

Frangi filter

ResNet34

Salivary ferning

ABSTRACT

Full ferning is the peak of the formation of a salt crystallization line pattern shaped like a fern tree in a woman's saliva at the time of ovulation. The main problem in this study is how to detect the shape of the salivary ferning line patterns that are transparent, irregular and the surface lighting is uneven. This study aims to detect transparent and irregular lines on the salivary ferning surface using a comparison of 15 pre-trained convolutional neural network models. To detect fern-like lines on transparent and irregular layers, a pre-processing stage using the Frangi filter is required. The pre-trained convolutional neural network model is a promising framework with high precision and accuracy for detecting fern-like lines in salivary ferning. The results of this study using the fixed learning rate model ResNet50 showed the best performance with an error rate of 4.37% and an accuracy of 95.63%. Meanwhile, in implementing the automatic learning rate, ResNet18 achieved the best results with an error rate of 1.99% and an accuracy of 98.01%. The results of visual detection of fern-like lines in salivary ferning using a patch size of 34×34 pixels indicate that the ResNet34 model gave the best appearance.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mohd Zamri Ibrahim

Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang

Pahang, Pekan Campus, Malaysia

Email: zamri@ump.edu.my

1. INTRODUCTION

In several previous studies, there have been many discussions about detecting crack lines on several objects, including road asphalt, walls, concrete, iron, and steel. The methods used range from computer vision to deep learning; almost all of them use datasets based on concrete images so that the crack lines are visible, for example, to detect cracks in buildings [1], [2], detection of concrete pavement cracks [3], [4]. The main contribution of this study is the development of deep learning-based image processing methods to detect fern-like lines from salivary ferning on overlapping layers, transparent layers, and irregular fern-like lines from raw images that have uneven lighting on the surface, as shown in Figure 1. During the study, one important challenge encountered in this research with salivary ferning image as the main object is the level of different hormonal fertility of each woman. It affects the shape of the pattern and the number of fern-like lines in salivary ferning within their ovulation period. Besides that, even in the same woman, the number and form of fern-like patterns in each menstrual calendar cycle are different due to progesterone and estrogen caused by fatigue, stress levels, long trips, illness, smoking, and drinking alcohol. The form of fern-like lines and their number in salivary ferning will peak when a woman is ovulating; this condition is known as full ferning (FF). Meanwhile, the condition where there are no fern-like lines in salivary ferning or a small amount of

fern-like lines is found; this is called no ferning (NF). Commonly a woman's fertile period, especially to detect her ovulation period, can be done by using several ways, including a menstrual cycle calendar system, ovulation prediction kits (OPK) tools, test-packs, measuring basal body temperature, cervical fluid analysis and the ovutest scope.

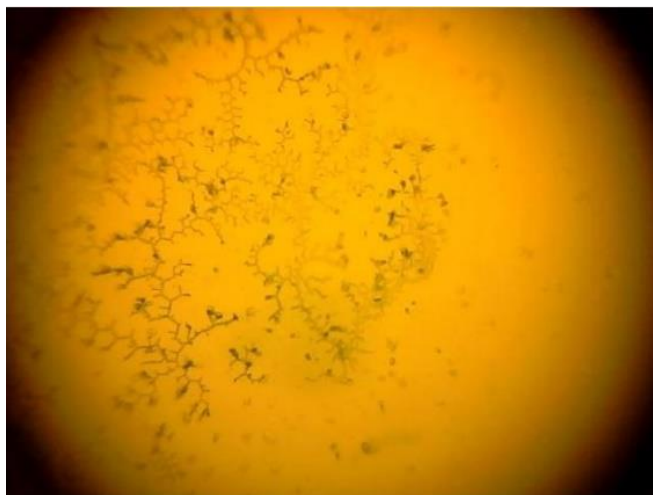


Figure 1. Display of fern-like lines from salivary ferning

In the previous study, Eissa *et al.* [5] made a microcontroller-based tool using an infrared thermometer sensor to detect a woman's ovulation time through body temperature by collecting three parameters: temperature, time, and time factor decision making. The first drawback is that the tool is still in a prototype form. Second, it is quite difficult for ordinary people to operate it because they must first understand how the hardware and software work. Wu *et al.* [6] conducted a study to predict women's ovulation period through salivary images using a conventional method approach: image processing and data mining using the J48 decision tree algorithm in the Weka program, with the classification results of 84% of 100 saliva samples. Ovulation detection through salivary ferning can also be used in animals; Ravinder *et al.* [7] conducted a study related to salivary crystallization patterns to detect estrus in eight females *Bubalus bubalis* buffaloes for three months. Meanwhile, Kubátová and Fedorova [8] studied the relationship between salivary crystallization and the fertile period of three female Bornean orangutans in their menstrual cycle.

Potluri *et al.* [9] made smartphone-based hardware and microfluidic devices to predict women's ovulation period through artificial saliva and human saliva with claims of accuracy > 99%. Luo *et al.* [10] made a tool used to detect and predict ovulation by measuring the temperature of the ear canal using an in-ear thermometer. The temperature measurement data is carried out every 5 minutes during sleep hours and then sent to a smartphone for analysis. The results of 34 volunteers have a detection accuracy with a sensitivity of 92.31% and a prediction rate of only 23.07% to 31.55%.

We argue that image processing using deep learning methods can have promising results in detecting fully autonomous fern-like line structures. Applying deep learning methods will reduce computation time and perform a faster process and provide more precise feature measurements to avoid human error factors. For example, the convolutional neural network (CNN) has high accuracy prediction capabilities in image recognition and classification tasks. The CNN model performs the training process by building a filter in a 3-dimensional space, with two dimensions (length and width) and one channel. The work in this study aimed to detect hidden fern-like lines using several types of pre-trained convolutional neural network models combined with pre-processing of independent test image data using the Frangi filter.

2. RESEARCH METHOD

The main problem in detecting the shape of the fern-like line pattern described in the introduction section is the display form, as shown in Figure 1. Visually, the fern-like line cannot be processed directly using deep learning; a pre-processing approach requires morphological operations, transformations, or adaptive filters to perform feature extraction. Therefore, feature extraction will be used to distinguish between fern-like and non-fern-like lines in the salivary ferning image, divided into several small patches. The block diagram of the method used in this study is shown in Figure 2.

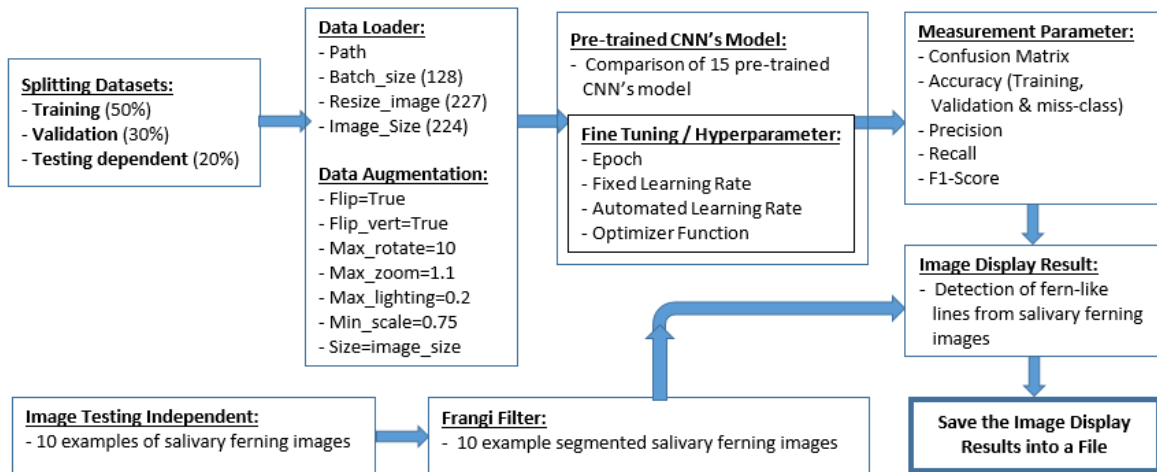


Figure 2. Block diagram of the method used in the study

Detecting fern-like lines in salivary ferning images used in this work are based on the block diagram method in Figure 2, and this study combines two techniques: deep learning and Frangi filter. In pre-trained CNN's, a comparison process will be carried out from 15 models, and this will be done to find out which transfer learning from the fifteen models visually produce the best fern-like lines detection. In comparison, the Frangi filter method will be used for the detection process and visually displays the shape of the tubular line on a transparent, hidden, and uneven layer of illumination from fern-like lines.

The process of detecting fern-like lines in this study will be carried out directly on independent testing images of salivary ferning, which the Frangi filter had segmented. So that in this study, there will be no need for stages of creating ground-truth, labelling, captioning, and annotation images of all dataset images used in the training, validation, and testing processes to save time, costs with less effort. In other studies, the same method was applied to detect cracks in the image of an object where the fracture line of the object is visually more apparent and more accurate so that it can be seen clearly which parts of the image reveal crack lines and which parts have no crack lines. The application of this research, for example, can be used to detect crack lines in walls, floors, cracks in steel, iron, and to detect cracks in road asphalt. Research conducted by Yang *et al.* [11] emphasized that deep learning as new technology has great potential to replace traditional crack detection methods in recognizing and detecting surface cracks for structural safety.

2.1. Data acquisition

This study's salivary ferning dataset images were obtained from ten female volunteers for three consecutive months with an average of one menstrual cycle from the ten volunteers for 24 days. The ten volunteers were of productive age (20 to 40 years), in good health, did not smoke, use contraceptive pills, or consume alcoholic beverages. The process of taking dataset images was carried out every day through saliva dripping on the glass surface of the ovutest scope lens after waking up from sleep, before eating and drinking anything, and before brushing teeth. Based on empirical data, the form of full ferning image visualization from each volunteer in each menstrual cycle has a different fern-like line pattern structure, influenced by stress levels, fatigue factors, and changes in the hormones progesterone and estrogen.

To increase the number of image datasets in the training, validation, and independent testing directory in this study, data augmentation (flip, rotate, zoom, lighting, and scale) processes were carried out; this was necessary so that the results of the computational approach in deep learning did not experience overfitting or underfitting conditions. Some artificial salivary ferning images were also taken from the internet. Therefore, the total number of salivary ferning images in this study is 3,779,000 images with a size of 227×227 pixels. Splitting datasets in the training directory for ferning images (positive) and no-ferning (negative) images by 50%, in the validation directory containing ferning and non-ferning images, obtained randomly with a percentage of 30% while in the independent testing directory getting images randomly by 20%.

2.2. Pre-trained CNN's model

Pretrained models are an architecture that has been trained on other datasets for different purposes. For example, we use ResNet34 as a pre-trained network. This residual network has 34 layers that have trained more than one million images from the ImageNet dataset so that the pre-trained network can classify pictures of more than 1000 classes. Therefore, the model can recognize various objects before training on the pretrained network dataset.

Transfer learning is a technique that allows us to use a pre-trained model to perform different tasks on a new set of datasets. Fine-tuning, in principle, is a transfer learning technique used to update the weights of the pre-trained model through a training process of several epochs, determining fixed or automated learning rates and selecting the required optimizer function in our new dataset. Once we have selected a pre-trained model, we update the previously trained data to suit our new task.

This study compares fifteen pre-trained CNN architecture models to get the best visualization results on the fern-like lines detection process; fifteen pre-trained CNN architecture models were compared. The fifteen CNN architecture designs include: residual neural network (ResNet)18/34/50, AlexNet, SqueezeNet1.0/1.1, visual geometry group (VGG)11, XresNet18/Deep/Deeper, XresNet34/Deep/Deeper, and XresNext18/34. Each number that follows the name of the pre-trained CNN's model and shows the version also indicates how many layers are in the architecture. The pre-trained model used in this study is Keras-based, trained on a large image dataset, namely ImageNet. ImageNet consists of image datasets arranged in a hierarchy where the hierarchy of each node contains hundreds to thousands of images [12]. For example, a dog subtree will have many branches consisting of dog images that have been grouped by breed. Millions of annotated images are divided into different image categories; all images on ImageNet are obtained from the web and then labelled or graded using Amazon's Mechanical Turk.

2.3. Measurement parameters

To measure the performance of the classification process in deep learning with the output in the form of 2 classes which are commonly referred to as binary classification, and the output of multiple classes classification can be seen through the confusion matrix table. This study's output has two classes: full ferning (positive) and no-full ferning (negative), so the confusion matrix table has different prediction and actual values. There is a true positive (TP), true negative (TN) area box, a false positive (FP) area box is considered a type-I error, and a false negative (FN) is a type-II error. Figure 3 is an example of the display form of the confusion matrix table.

Actual	Negative	669	9
	Positive	9	68
		Negative	Positive
		Predicted	

Figure 3. The display of the confusion matrix table

Based on the confusion matrix table data, we can get a report on the classification of values from accuracy, precision, recall, *F1-score*, and support scores for the model. $Precision = TP / (TP + FP)$ is the accuracy of optimistic prediction, $Recall = TP / (TP + FN)$ is the positive fraction correctly identified, $F1\text{-score} = 2 \times (precision \times recall) / (precision + recall)$ is harmonic average of precision and recall so that the best value of the *F1-score* is one while the worst value is 0. At the same time, the measured support parameter indicates the number of occurrences of a given class in the dataset.

2.4. Frangi filter

Segmentation is one of the crucial stages in image processing, where there is a process of separating the image into several homogeneous areas. If extracted in that area, it will become an object observed in the region of interest [13]. Region of interest segmentation ferning pattern of saliva can be contour and fern-like based on the intensity values of similarity and discontinuity. The image similarity approach in similar regions

is based on a criterion: region growing, region splitting, and thresholding. In contrast, the image discontinuity approach is divided based on a sudden change in intensity, for example, edge detection [14].

In the medical field, segmentation using the Hessian matrix is widely used, including detecting blood vessels [15] and the respiratory tract [16]. In the plant biology field, the Hessian matrix was applied to detect the branching of the main stem of plants [17], [18] combined Hessian matrix and Hough transform to segment plant stems. Channel structures, planes, and bubbles from 2D and 3D images can be detected by analyzing the eigenvalues of the Hessian matrix.

The matrix contains an array of numbers or elements arranged in a row and column format; the Hessian matrix is a matrix in which each element is formed from the second partial derivative of a function. For example, a function $f(x)$ with n variables has a second partial derivative, and the derivative is continuous, then the Hessian matrix of $f(x)$ is the matrix H [19]. The Hessian matrix is used to test the derivatives of two functions of more than one variable by identifying the local optimum. The two-variable function is used because the pixel intensity of image $I(x, y)$ has two variables, namely x and y . The optimum value can be found using the eigenvalues of the Hessian matrix as follows, if A is a matrix of order $(n \times n)$ and λ is a scalar that satisfies the equation $Ax = \lambda x$ for a non-zero column vector in n -dimensional space, then:

- 1) λ is called the eigenvalue or characteristic roots of matrix A
- 2) x is called the eigenvector or characteristic vector of matrix A
- 3) The eigenvectors x form the linearly independent eigenvector space of A called the basis for the eigenspace corresponding to the eigenvalue λ

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \quad (1)$$

The Hessian matrix provides the second derivative of the local intensity variation of the image concerning the surrounding pixels. The eigenvalues and eigenvectors of the Hessian matrix are used to analyze the image structure. Frangi (1998) defines the relationship between the eigenvalues $\lambda_1, \lambda_2, \lambda_3$ with $|\lambda_1| \leq |\lambda_2| \leq |\lambda_3|$ in Table 1.

Table 1. The relationship of eigenvalues on the Hessian matrix and image structure [20]

2D		3D			Structure orientation	Information
λ_1	λ_2	λ_1	λ_2	λ_3		
L	L	L	L	L	No structure	L = low
L	L	L	L	H-	Sheet-like (bright)	H+ = high positive
L	L	L	L	H+	Sheet-like (dark)	H- = high negative
L	H-	L	H-	H-	Tubular (bright)	+/- = eigen value sign
L	H+	L	H+	H+	Tubular (dark)	
H-	H-	H-	H-	H-	Blob-like (bright)	
H+	H+	H+	H+	H+	Blob-like (dark)	

To get the Hessian matrix on a 2D image, the second partial derivative of an image is calculated: D_{xx}, D_{yy} , and D_{xy} .

$$H(x, y) = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (2)$$

The Frangi filter combines an image enhancement smoothing process using a Gaussian convolution with a second derivative to detect “vesselness” in the image. For example, in the 1D case, the response image of the filter is given in (3) [21]. $I(x)$ is the input image, and $*$ is the convolution operator.

$$D(x, \sigma) = \left\{ -\frac{d^2}{dx^2} G(x, \sigma) \right\} * I(x) \quad (3)$$

The calculation of the response of the $D(x, \sigma)$ image is a Gaussian scale-space; Linderberg [22] explains the theory of scale-space as a series of 1D images blurred with a blur index σ or the standard deviation of the Gaussian function. The blur index σ is defined as an “s scale” i.e. the Gaussian kernel’s size

affects the resulting image blur. Increased smoothing can cause shape distortion in the extracted edges. The application of smoothing at a low scale will extract too many edges, so selecting the s scale must be done correctly. For the simple Gaussian smoothing function shown in (4), to get the second partial derivative Hessian matrix element from the image, the image is convoluted with the second derivative Gaussian function, namely: $D_{xx} = I(x) * G_{xx}$, $D_{xy} = I(x) * G_{xy}$, $D_{yy} = I(x) * G_{yy}$. The Gaussian scale-space functions are G_{xx} , G_{xy} , and G_{yy} and are shown in (5), (6), and (7).

$$D(x, y, s) = \frac{1}{2\pi s^2} e^{-\frac{x^2 + y^2}{2s^2}} \quad (4)$$

$$G_{xx} = \frac{\partial^2 G(x, y, s)}{\partial^2 x} = \frac{1}{2\pi s^4} \left(\frac{x^2}{s^2} - 1 \right) e^{-\frac{x^2 + y^2}{2s^2}} \quad (5)$$

$$G_{yy} = \frac{\partial^2 G(x, y, s)}{\partial^2 y} = \frac{1}{2\pi s^4} \left(\frac{y^2}{s^2} - 1 \right) e^{-\frac{x^2 + y^2}{2s^2}} \quad (6)$$

$$G_{xy} = \frac{\partial^2 G(x, y, s)}{\partial^2 xy} = \frac{xy}{2\pi s^6} e^{-\frac{x^2 + y^2}{2s^2}} \quad (7)$$

$$\lambda_2 = \frac{D_{xx} + D_{yy} + \alpha}{2} \quad (8)$$

$$\lambda_1 = \frac{D_{xx} + D_{yy} - \alpha}{2} \quad (9)$$

With $x, y = [-3s: 3s]$, the second derivative gaussian filter kernel is generated. The eigenvalues are searched using (8) and (9), where $\alpha = ((D_{xx} + D_{yy})^2 + 4D_{xy}^2)$ then the eigenvalues are sorted so that $\lambda_2 > \lambda_1$. The eigenvalues are used to detect the structure of all pixels; based on Table 1, pixels that are part of the vessel area are marked with $\lambda_1 \approx 0$ and $\lambda_2 \gg \lambda_1$. This requirement is formulated by the blobness measure feature, namely $Rb = (\lambda_1/\lambda_2)^2$. At the same time, $S = ||\mathcal{H}||F = (\lambda_2 + \lambda_1)^2$ is a “second-order structure.” The value of S will be low if the background has no structure. The norm value will be greater in areas with high contrast because one eigenvalue will increase. The features of Rb and S are mapped by the vesselness measure (10). In the (10), β and c are threshold values that control the sensitivity of the Hessian matrix line filter. The line filter response will be maximum at a scale that matches the original ferning. Thresholding results were analyzed on different s scales as in (11).

$$vo(s) = \begin{cases} 0, & \lambda_2 > 0 \\ \exp\left(-\frac{Rb^2}{2\beta^2}\right) \left(1 - \exp\left(-\frac{S^2}{2c^2}\right)\right) & \end{cases} \quad (10)$$

$$Vo = \max_{s_{min} \leq s \leq s_{max}} Vo(s) \quad (11)$$

In this study, we propose a method with the name Harmony Frangi filter, which is a development of the Frangi filter method introduced by Niessen *et al.* [23] in 2002, improved by Kroon [24] in 2009, then further developed by Jerman *et al.* [25] in 2015. This study carried out harmonization starting from the pre-processing stage, Frangi filtering, to the post-processing stage. The Frangi filtering method has been compared with twelve other filtering methods, including Canny, Robinson, Kirsch, gradient, extracted largest blob, convolve, hill shade, differentials Laplacian, Frangi, Frangi German team, and Coye algorithm [26]. Furthermore, the Frangi filter method is harmonized with three thresholding methods: imadjust, histeg, and adapthisteg to get the best foreground display results.

The next step is to configure several parameters from the Frangi filter method to get the most realistic and natural foreground, approaching the ferning appearance of the original image. The parameters that can be configured in the Frangi filter include scale range (sigma), scale ratio, Frangi beta 1 and Frangi beta 2. The comparison results of the Frangi filter parameter configuration can be seen in Figure 4(a) for a scale range of 1, Figure 4(b) with a scale range of 2, Figure 4(c) configured for a scale range of 4, and Figure 4(d) using a scale range of 8. The four images show that there are differences in the results of the detection process in the thickness of fern-like lines.

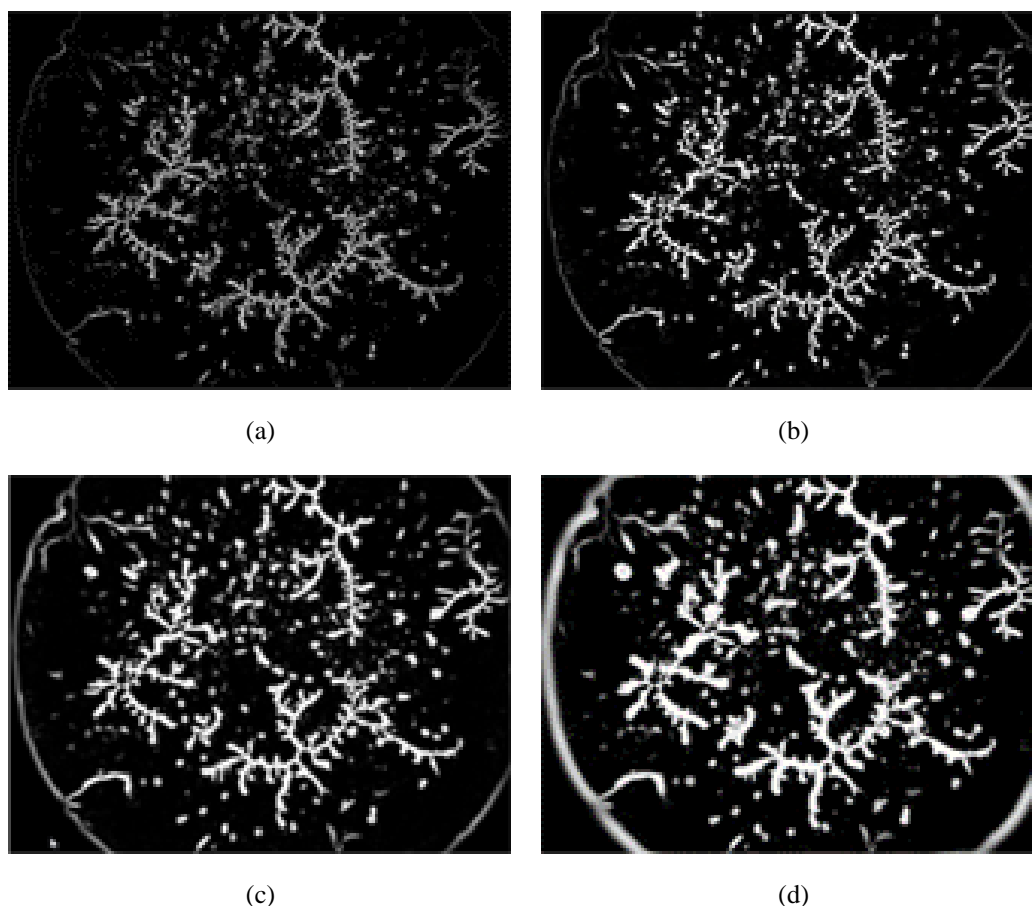


Figure 4. Display of Frangi filter parameter comparison results: (a) scale range = 1, (b) scale range = 2, (c) scale range = 4, and (d) scale range = 8

3. RESULTS AND ANALYSIS

In this study, a comparison of different pre-trained CNN model architectural families will be carried out, including the architectures of residual neural network (ResNet), AlexNet, SqueezeNet, visual geometry group (VGG), XresNet, and XresNext, to get the best visualization results from the detection of fern-like lines in salivary ferning. A brief explanation of each of the above network architecture models is as follows: ResNet, ResNet is a deep neural network developed by He *et al.* [27] that allows for good training performance with hundreds to thousands of layers. ResNet has good capabilities for recognition tasks and is one of the popular architectures for various computing tasks in computer vision.

Second, AlexNet is a deep convolutional neural network used for good performance and fast computation time for photo classification processes. Krizhevsky *et al.* [28] are discussed in a famous paper entitled “Imagenet classification with deep convolutional neural network”. In the competition in ImageNet large-scale visual recognition, the AlexNet network model has an error of 15.3% and has a percentage point of 10.8% lower than the runner up.

SqueezeNet is a deep neural network released in 2016 with the main task in computer vision developed by researchers at the University of California, DeepScale, Stanford University, and Berkeley [29]. The researcher’s goal in designing SqueezeNet was to make the neural network and parameters smaller to be quickly loaded into computer memory and transmitted remotely over a computer network. The paper states that SqueezeNet has an accuracy level equivalent to AlexNet on ImageNet with 50x fewer parameters. The compression technique in SqueezeNet is less than 0.5 MB or 510× smaller than AlexNet.

The VGG network is a convolutional neural network model introduced by Simonyan and Zisserman [30]. The paper title “Very deep convolutional networks for large-scale image recognition” has a top 5 test accuracy of 92.7% on ImageNet with more than 14 features million images in a thousand classes. VGG is one of the well-known architectures in deep learning environments with its first large layers 11 and 5 is an upgrade from the AlexNet architecture and has several filters with a kernel size of 3×3 one by one. The previous AlexNet model focused on smaller window sizes and strides in the first convolution layer, whereas VGG was more concerned with the depth aspect of CNN’s.

The basic idea for designing and building XresNets was to introduce the concept of a “Bag of tricks for image classification with convolutional neural networks.” He *et al.* [31] of Amazon web services introduced XresNets; XresNets featured only three tweaks with different names (ResNet-A, ResNet-B, and ResNet-C), focusing on improving three separate convolution steps in the ResNet architecture. In XresNets, applying some model modification and training tricks, heuristics is one way to improve parallel processing in training, lower precision computing, and modify bias or learning rate.

While ResNext is a variant of ResNet, which is very similar to the Inception module, where both have a split-transform-merge paradigm, the difference is that ResNext different output lines are combined by adding them together, and all lines have the same topology [32]. ResNext has a hyper-parameter named cardinality, the number of independent paths to provide a new method of adjusting the model capacity. Based on experiments that have been carried out, accuracy is more efficient by increasing cardinality rather than deepening and expanding. ResNext is easier to adapt to new datasets or tasks because it has a simple paradigm, and only one hyper-parameter must be adjusted. Table 2 shows the number of parameters from the fifteen pre-trained CNN’s described.

Table 2. The number of parameters of the fifteen pre-trained CNN’s model

Pre-trained model	Parameters	Total		Optimizer	Loss function
		Trainable parameters	Non-trainable parameters		
ResNet18	11,704,896	537,984	11,166,912		
ResNet34	21,813,056	545,408	21,267,648		
ResNet50	25,615,424	2,160,512	23,454,912		
AlexNet	2,734,912	265,216	2,469,696		
SqueezeNet1.0	1,263,888	528,384	735,424		
SqueezeNet1.1	1,250,880	528,384	722,496		
VGG11_BN	9,754,368	533,888	9,220,480	Adam	Flattened loss of cross-entropy
XresNet18	11,724,128	538,112	11,186,016		
XresNet18_Deep	14,543,712	277,504	14,266,208		
XresNet18_Deeper	11,003,744	276,480	10,727,264		
XresNet34	21,832,288	545,536	21,286,752		
XresNet34_Deep	24,651,872	284,928	24,366,944		
XresNet34_Deeper	27,013,216	286,976	26,726,240		
XresNext18	13,571,168	541,952	13,029,216		
XresNext34	23,998,688	553,088	23,445,600		

3.1. Performance using fixed learning rate

The system can improve performance and accuracy by tuning and setting the hyper-parameters, one of the most important things in determining the value of the learning rate. To get the best and highest accuracy, the value of the learning rate parameter must be correct, experimentation is needed, and of course, this step requires a lot of time and experimentation. In this study, the learning rate value will be searched and determined automatically from a curve point called the valley point, and this valley point is the point with the maximum slope. In addition to determining the valley point on the curve graph in this study, other point values will also be sought, including minimum point, steep point, and slide point, as shown in Figure 5. Based on the display in Figure 5, the valley point has an optimal slope angle with the learning rate value close to 10^{-3} , and then the value used is the fixed learning rate value.

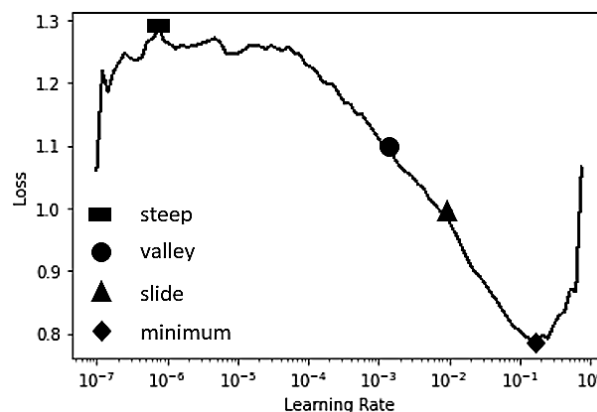


Figure 5. Finding and determining the point value of the valley

The results of determining the value of the fixed learning rate will be obtained. A table containing the actual and predicted values of the confusion matrix, error rate, and the percentage of accuracy values from the validation set is shown in Table 3. Based on Table 3, the pre-trained model ResNet50 and VGG11_BN have the best performance from all sides, both from the confusion matrix table, error rate, and accuracy value, namely: 95.63%. Figure 6 shows the results of the classification process where an error occurred in the validation set between the actual and predicted conditions. Prediction results may be wrong, and this can happen because some of the original labels seem to be inaccurate, in this case causing difficulty in judging. This misclassification or prediction will provide insight into how well our model performs.

Table 3. Percentage of error rate, accuracy, and confusion matrix on fixed learning rate

Pre-trained model	Actual / Prediction				Error rate (%)	Accuracy (%)	Time
	TP	TN	FP	FN			
ResNet18	61	651	27	16	5.70	94.30	00:13
ResNet34	58	656	22	19	5.43	94.57	00:17
ResNet50	62	660	18	15	4.37	95.63	00:29
AlexNet	51	652	26	26	6.89	93.11	01:12
SqueezeNet1.0	62	659	19	15	4.50	95.50	00:12
SqueezeNet1.1	64	651	27	13	5.30	94.70	00:17
VGG11_BN	58	664	14	19	4.37	95.63	00:24
XresNet18	52	623	55	25	10.60	8.40	00:13
XresNet18_Deep	44	646	32	33	8.61	91.39	00:14
XresNet18_Deeper	48	631	47	29	10.07	89.93	00:13
XresNet34	54	605	73	23	12.72	87.28	00:15
XresNet34_Deep	55	588	90	22	14.83	85.17	00:16
XresNet34_Deeper	64	578	100	13	14.97	85.03	00:16
XresNext18	39	667	11	38	6.49	93.51	03:14
XresNext34	39	668	10	38	6.36	93.64	00:18

3.2. Performance using auto-learning rate

The steps and discussion to see the performance of the fifteen pre-trained network models using the auto-learning rate are precisely the same as measuring performance using the fixed learning rate on the validation set, as discussed in point 3.1. The main difference is that if you use a fixed learning rate, the learning rate value is only one value, for example, $1e - 3$ (10^{-3}), while in the auto-learning rate, the learning rate value is between two values, for instance, between $1e - 3$ and $1e - 4$ (10^{-3} up to 10^{-4}). Based on experiments conducted empirically, the results are obtained in Table 4. Based on Table 4, the model of pre-trained networks with the highest performance are ResNet18 in terms of the confusion matrix, error rate, and accuracy value of 98.01%. The results of the comparison between the fix learning rate in Table 3 and the auto-learning rate in Table 4 for the percentage value of the error rate and accuracy obtained the following data: first, there was a decrease in the average error rate value of the fifteen network models by 3.08%, and second, there was an increase in the average accuracy value of 3.08%, as shown in Figure 7.

Table 4. Percentage of error rate, accuracy, and confusion matrix on auto-learning rate

Pre-trained model	Actual / Prediction				Error rate (%)	Accuracy (%)	Time
	TP	TN	FP	FN			
ResNet18	68	672	6	9	1.99	98.01	00:14
ResNet34	68	669	9	9	2.38	97.62	00:21
ResNet50	65	674	4	12	2.12	97.88	00:37
AlexNet	59	657	21	18	5.17	94.83	00:12
SqueezeNet1.0	60	669	9	17	3.44	96.56	00:14
SqueezeNet1.1	55	669	9	22	4.11	95.89	00:19
VGG11_BN	70	667	11	7	2.38	97.62	00:29
XresNet18	48	657	21	29	6.63	93.38	00:16
XresNet18_Deep	39	667	11	38	6.49	93.51	00:17
XresNet18_Deeper	48	651	27	29	7.42	92.58	00:16
XresNet34	54	641	37	23	7.95	92.05	00:20
XresNet34_Deep	47	669	9	30	5.17	94.83	00:21
XresNet34_Deeper	51	646	32	26	7.68	92.32	00:21
XresNext18	39	672	6	38	5.83	94.17	00:17
XresNext34	40	668	10	37	6.23	93.77	00:24

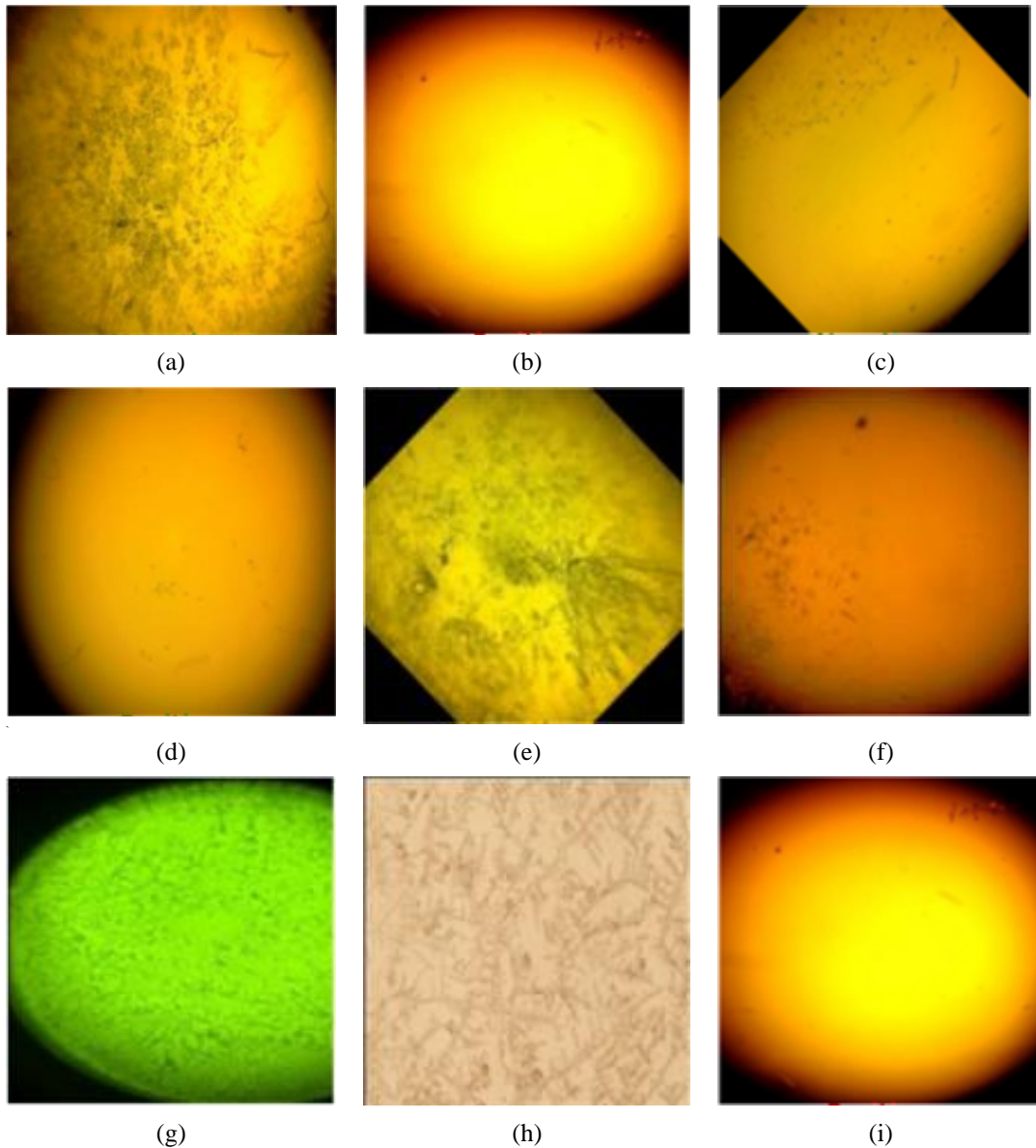


Figure 6. Checking the wrong classification: (a) negative-negative, (b) negative-negative, (c) negative-negative, (d) negative-negative, (e) positive-negative, (f) negative-negative, (g) positive-positive, (h) positive-positive, and (i) negative-negative

Before the prediction process is carried out, a classification process is needed to check the class misclassification of the salivary ferning image between the trained dataset and the validation dataset. The results of the class classification process can be seen in all the images in Figure 6. Figure 6(a), Figure 6(b), Figure 6(c), Figure 6(d), Figure 6(e) Figure 6(f) are images that are included in the no ferning dataset (negative) and are correctly predicted as images in the no ferning (negative) class. Figure 6(g) and Figure 6(h) are images on the full ferning dataset (positive) and are correctly predicted as images in the full ferning class (positive). In contrast, Figure 6(i) is an image that shows an error in the classification process, which is predicted to be a full ferning class (positive). Still, the image is included in the image in the no ferning dataset (negative).

3.3. Performance on independent dataset test

At this stage, the model's performance is checked on an independent dataset test to get predictive results from the test data to get the actual label of the encoded label. This step produces a value in the confusion matrix where the parameter values in the confusion matrix can be calculated using a formula to get the parameter values measured from precision, recall, *F1*-score, support, and accuracy, as shown in Table 5. In Table 4, for example, the ResNet34 model resulting from the validation dataset has an accuracy of 97.62%.

In comparison, in Table 5, the ResNet34 model results from the independent dataset test have an accuracy of 99.88%, with a difference of 2.27%. Overall, all the independent dataset test accuracy values in Table 5 increase 4.33% on average. The training process for all models in this study used images of 224×224 pixels.

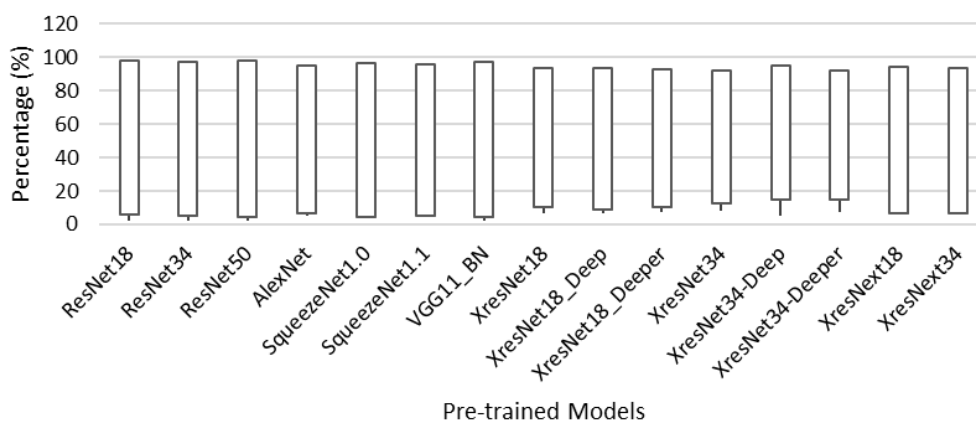


Figure 7. Comparison graph of fix learning rate and auto-learning rate

Table 5. The results of the measured value parameters on the independent dataset test
Independent dataset tes (*epochs* = 5)

Pre-trained model	Actual / prediction				Precession	Recall	F1-score	Accuracy (%)
	TP	TN	FP	FN				
ResNet18	132	718	2	0	0.99	1.00	0.99	99.77
ResNet34	132	719	1	0	0.99	1.00	0.99	99.88
ResNet50	132	717	3	0	0.98	1.00	0.99	99.65
AlexNet	132	716	4	0	0.99	1.00	0.99	99.53
SqueezeNet1.0	130	717	3	2	0.98	0.98	0.98	99.41
SqueezeNet1.1	132	713	7	0	0.99	1.00	0.99	99.72
VGG11_BN	132	718	2	0	0.99	1.00	0.99	99.77
XresNet18	132	707	13	0	0.99	1.00	0.99	98.47
XresNet18_Deep	130	713	7	2	0.95	0.98	0.97	98.94
XresNet18_Deeper	130	713	7	2	0.95	0.98	0.97	98.94
XresNet34	132	706	14	0	0.90	1.00	0.95	98.36
XresNet34_Deep	128	715	5	4	0.96	0.97	0.97	98.94
XresNet34_Deeper	132	705	15	0	0.90	1.00	0.95	98.24
XresNext18	124	711	9	8	0.93	0.94	0.94	99.77
XresNext34	130	714	6	2	0.96	0.98	0.97	99.06

3.4. Detection of fern-like lines in saliva images

The model training process in this study uses an image size of 224×224 pixels, whether it is processed by training datasets, dataset validation, and independent dataset tests. The final step in this study is to detect fern-like lines in saliva images with a size of 578×814 pixels. The saliva image with a large enough size will be divided into several patches or windows size with a smaller size, where each patch will detect the presence of a line of saliva. If any fern-like lines are detected, the patch and the image will be put back together so that the detection of fern-like lines in other parts of the original image can be continued. If the patch size is smaller than the image size used for training, the patch will be discarded. The patch size or windows size can be changed according to the size of the line to be detected in the image, for example, with a size of 16×16 pixels, 32×32 pixels, 64×64 pixels, 128×128 pixels, and so on. In this study, fifteen models used a patch size of 32×32 pixels to detect fern-like lines in saliva. The visualization display of fern-like lines detection results from the fifteen pre-trained models can be seen in Figure 8.

Based on the visualization of fern-like lines detection results in Figure 8 using a patch size of 34×34 pixels, it can be explained as follows: first, in Figure 8(a), Figure 8(b), Figure 8(c), Figure 8(d), Figure 8(e), Figure 8(f), Figure 8(g), Figure 8(h) and Figure 8(i) show that there is overlap because the fern-like line area and the non-fern-like line area are all detected in the figure salivary ferning. Second, Figure 8(j) using the ResNet50 model cannot detect the shape of the fern-like lines pattern. Third, Figure 8(k), Figure 8(l), Figure 8(m), Figure 8(n) and Figure 8(o) can detect the presence of fern-like lines, but the ResNet34 model in Figure 8(k) has resulted from the best visualization in detecting the shape of the fern-like lines pattern.

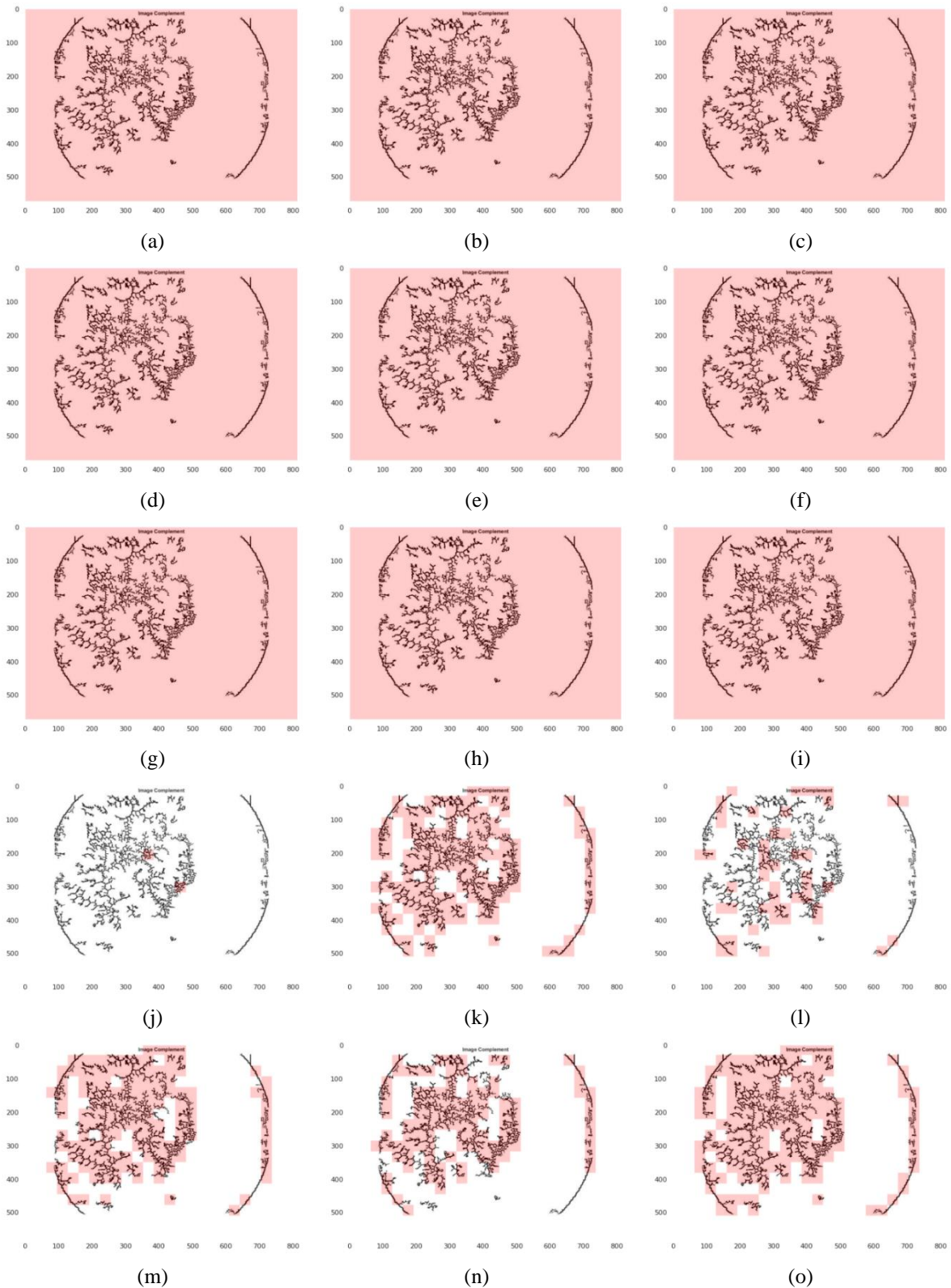


Figure 8. Visualization of fern-like lines detection results: (a) ResNet18, (b) VGG11_BN, (c) XresNet18, (d) XresNet18_Deep, (e) XresNet18_Deeper, (f) XresNet34, (g) XresNet34_Deeper, (h) XresNetx18, (i) XresNext34, (j) ResNet50, (k) ResNet34, (l) AlexNet, (m) SqueezeNet1.0, (n) SqueezeNet1.1, and (o) XresNet34_Deep

4. CONCLUSION

By combining the pre-trained network model and the frangi filter, this research succeeded in detecting transparent, hidden, irregular fern-like lines with uneven lighting on the surface of the salivary ferning image. In addition, this study also compared fifteen pre-trained convolutional neural network models to determine which pre-trained network model produces the best performance in detecting and visualizing the shape of the fern-like lines pattern in salivary ferning images with a size of 578×814 pixels. Using a fixed learning rate ($1e - 3$), the pre-trained network model that gave the best performance is the ResNet50, with an error rate of 4.37% and an accuracy of 95.63%.

The pre-trained network model was configured using an auto-learning rate ranging between two values (example: $1e - 3$ up to $1e - 4$ (10^{-3} up to 10^{-4})). The experimental results empirically showed that the ResNet18 model achieved the best performance with an error rate of 1.99% and an accuracy of 98.01%. The results of the measurement value parameters on the independent dataset in the test directory show that the fifteen pre-trained network models produce parameter values of the confusion matrix table, precision values, recall, *F1-score* have high-performance results with accuracy above 98%.

The results of the visualization of fern-like lines detection in the salivary ferning image using a patch size or windows size of 34×34 pixels showed that the ResNet34 model has the best results in detecting the shape of the fern-like lines pattern. The results of this study can potentially develop insights and considerations for future researchers in the application of CNN's variants to detect fern-like lines in salivary ferning. Salivary ferning line pattern detection in this work has the main benefit of assisting medical practitioners or individuals in estimating women's ovulatory cycle. Understanding the almost exact time of the women's ovulatory cycle might help medical practitioners guide and suggest women's mate time preferences to conceive or avoid pregnancy.




REFERENCES

- [1] C. F. Özgenel and A. G. Sorguç, "Performance comparison of pretrained convolutional neural networks on crack detection in buildings," in *35th International Symposium on Automation and Robotics in Construction (ISARC 2018)*, 2018, pp. 693-700, doi: 10.22260/isarc2018/0094.
- [2] W. R. L. da Silva and D. S. de Lucena, "Concrete cracks detection based on deep learning image classification," *Proceedings*, vol. 2, no. 8, p. 489, 2018, doi: 10.3390/ICEM18-05387.
- [3] Z. Qu, J. Mei, L. Liu, and D. Zhou, "Crack Detection of Concrete Pavement With Cross-Entropy Loss Function and Improved VGG16 Network Model," in *IEEE Access*, vol. 8, pp. 54564-54573, 2020, doi: 10.1109/ACCESS.2020.2981561.
- [4] X. Feng *et al.*, "Pavement crack detection and segmentation method based on improved deep learning fusion model," *Mathematical Problems in Engineering*, pp. 1-22, 2020, doi: 10.1155/2020/8515213.
- [5] H. M. Eissa, A. M. Ahmed, and E. A. Elsehely, "Implementation of smart ovulation detection device," *Recent Advances in Biomedical and Chemical Engineering Materials Science*, pp. 82-86, Oct. 2014. [Online]. Available: <http://inase.org/library/2014/venice/bypaper/BICHE/BICHE-14.pdf>
- [6] H. -C. Wu, C. -Y. Lin, S. -H. Huang, and M. -H. Tseng, "An intelligent saliva recognition system for women's ovulation detection," *Asian Conference on Intelligent Information and Database Systems*, 2015, pp. 614-623, doi: 10.1007/978-3-319-15702-3_59.
- [7] R. Ravinder *et al.*, "Saliva ferning, an unorthodox estrus detection method in water buffaloes (*Bubalus bubalis*)," *Theriogenology*, vol. 86, no. 5, pp. 1147-1155, Sep. 2016, doi: 10.1016/j.theriogenology.2016.04.004.
- [8] A. Kubátová and T. Fedorova, "Saliva crystallization occurs in female bornean orangutans (*Pongo pygmaeus*): Could it be a new option for monitoring of menstrual cycle in captive great apes?," *PLOS ONE*, vol. 11, no. 7, pp. 1-11, 2016, doi: 10.1371/journal.pone.0159960.
- [9] V. Potluri *et al.*, "An inexpensive smartphone-based device for point-of-care ovulation testing," *Lab on a Chip*, vol. 19, no. 1, pp. 59-67, 2019. [Online]. Available: <https://europepmc.org/backend/ptpmcrender.fcgi?accid=PMC6321627&blobtype=pdf>
- [10] L. Luo, X. She, J. Cao, Y. Zhang, Y. Li, and P. X. K. Song, "Detection and Prediction of Ovulation From Body Temperature Measured by an In-Ear Wearable Thermometer," in *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 2, pp. 512-522, Feb. 2020, doi: 10.1109/TBME.2019.2916823.
- [11] C. Yang, J. Chen, Z. Li, and Y. Huang, "Structural crack detection and recognition based on deep learning," *Applied Sciences*, vol. 11, no. 6, 2021, doi: 10.3390/app11062868.
- [12] J. Deng, W. Dong, R. Socher, L. -J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.
- [13] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Understanding Digital Image Processing Using MATLAB*, Gatesmark Publishing, 2009.
- [14] R. Klette, *Concise computer vision: An introduction into theory and algorithms*, U.K.: Springer, 2014.
- [15] N. M. Salem, S. A. Salem, and A. K. Nandi, "Segmentation of retinal blood vessels based on analysis of the hessian matrix and Clustering Algorithm," *2007 15th European Signal Processing Conference*, 2007, pp. 428-432. [Online]. Available: <https://eurasip.org/Proceedings/Eusipco/Eusipco2007/Papers/a31-f01.pdf>
- [16] M. Rudzki, "Vessel detection method based on eigenvalues of the Hessian matrix and its applicability to airway tree segmentation," in *XI International PhD Workshop OWD 2009*, 2009, pp. 100-105. [Online]. Available: http://delibra.bg.polsl.pl/Content/30844/BCPS_34691_-_Vessel-detection-met_0000.pdf
- [17] C. L. McCarthy, N. H. Hancock, and S. R. Raine, "Automated internode length measurement of cotton plants under field conditions," *American Society of Agricultural and Biological Engineers*, vol. 52, no. 6, pp. 2093-2103, 2009, doi: 10.13031/2013.29198.
- [18] Z. M. Amean, T. Low, C. McCarthy, and N. Hancock, "Automatic plant branch segmentation and classification using Vesselness measure," in *Proceedings of Australasian Conference on Robotics and Automation*, 2013. [Online]. Available: https://www.researchgate.net/publication/286576522_Automatic_plant_branch_segmentation_and_classification_using_Vesselness_measure




- [19] S. J. Leon, *Linear algebra with applications*, 2010. [Online]. Available: <https://www.umassd.edu/media/umassdartmouth/mathematics/faculty/steve-leon/Leon8Errata.pdf>
- [20] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2006 vol. 1496, pp. 130-137, doi: 10.1007/bfb0056195.
- [21] Y. Sato *et al.*, "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images," *Medical Image Analysis*, vol. 2, no. 2, pp. 143-168, 1998, doi: 10.1016/s1361-8415(98)80009-1.
- [22] T. Lindeberg, "Edge detection and ridge detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 117-156, 1998, doi: 10.1023/a:1008097225773.
- [23] W. J. Niessen, C. M. van Bommel, A. E. Frangi, M. J. A. Siers, and O. Wink, "Model-based segmentation of cardiac and vascular images," *Proceedings IEEE International Symposium on Biomedical Imaging*, 2002, pp. 22-25, doi: 10.1109/ISBI.2002.1029183.
- [24] D. -J. Kroon, *Numerical optimization of kernel based image derivatives*, Enschede, N.L: University of Twente , 2009, [Online]. Available: http://www.k-zone.nl/Kroon_DerivativePaper.pdf
- [25] T. Jerman, F. Pernuš, B. Likar, and Ž. Špiclin, "Beyond Frangi: an improved multiscale vesselness filter," in *Proceedings Volume 9413, Medical Imaging 2015: Image Processing*, 2015, vol. 9413, pp. 623-633, doi: 10.1117/12.2081147.
- [26] T. Coye, *A Novel Retinal Blood Vessel Segmentation Algorithm for Fundus Images*. MATLAB Central File Exchange, 2015.
- [27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
- [28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, Jun. 2017, doi: 10.1145/3065386.
- [29] F. N. Iandola, S. Han , M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," *Conference paper at ICLR 2017*, Feb. 2016. [Online]. Available: <https://arxiv.org/pdf/1602.07360.pdf>
- [30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Conference paper at ICLR 2015*, 2015. [Online]. Available: <https://arxiv.org/pdf/1409.1556.pdf>
- [31] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of Tricks for Image Classification with Convolutional Neural Networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 558-567, doi: 10.1109/CVPR.2019.00065.
- [32] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5987-5995, doi: 10.1109/CVPR.2017.634.

BIOGRAPHIES OF AUTHORS






Heri Pratikno    obtained Magister Teknik (M.T.) in the electrical engineering study program - smart multimedia network from the Institut Teknologi Sepuluh Nopember (ITS), Indonesia. Currently, he is working at Universitas Dinamika, Indonesia, and is a PhD. Candidate at the Faculty of Electrical and Electronic Engineering Technology of University Malaysia Pahang, Malaysia. His research interests are Computer Vision, Image Processing, Deep Learning, IoT, Computer Networking, and Network Security. He can be contacted at email: heri@dinamika.ac.id.



Mohd Zamri Ibrahim    obtained both B.Eng and M.Eng in electrical engineering from Universiti Teknologi Malaysia, Malaysia, and a PhD in electrical and electronics engineering from Loughborough University, United Kingdom. His research interests are in the area of computer vision, internet of thing, embedded system programming, brain-computer interaction, image processing, intelligent system, and speech recognition. He is a senior lecturer at the Faculty of Electrical and Electronic Engineering Technology of University Malaysia Pahang, Malaysia. He had received more than 10 awards at national and international research exhibition and currently a principal investigator on TERAJU RM500K project on the Portable Vein Finder imaging device. Before joining University Malaysia Pahang, He was a procurement engineer at Hewlett-Packard (HP) Malaysia where he worked as main technical interface with suppliers and HP design center (Vancouver and San Diego) to drive cost reduction, quality improvement and assurance of supply. He can be contacted at email: zamri@ump.edu.my.



Jusak    received B.S. degree in electrical engineering from Brawijaya University, Malang, Indonesia in 1996 and PhD degree in electrical engineering from Royal Melbourne Institute of Technology (RMIT) University, Melbourne, Australia in 2006. From 2009 to 2011, he was a Postdoctoral Research with Massey University, Palmerston North, New Zealand working in a Next Generation Networks research project supported the Telecom New Zealand. Between 2011 and 2021, he was a Senior Lecturer with Computer Engineering Department, Dinamika University, Surabaya, Indonesia. He is currently a Senior Lecturer IoT with James Cook University Singapore. His research interest includes signal processing for wireless communication networks, biomedical signal processing, and Internet of Things for medical applications and its security. Dr. Jusak was a recipient of PhD research award in 2004. He received several national-level research competitive grants from 2011 to 2020 and the best paper award in the International Conference on Information Technology Applications and Systems in 2018. He can be contacted at email: jusak@dinamika.ac.id.