Thai Journal of **Math**ematics
Volume 21 Number 2 (2023)
Pages 399–412

http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

International Conference on Digital Image Processing and Machine Learning 2022

Application of a Twin Parametric Support Vector Machine and Deep Learning Techniques for Abnormal Retinopathy Detection from Fundus Photographs

Kittikorn Sriwichai, Jessada Tanthanuch and Panu Yimmuang*

School of Mathematics, Institute of Science, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand
e-mail : nicesri59@outlook.com (K. Sriwichai); jessada@g.sut.ac.th (J. Tanthanuch); panu.y@sut.ac.th (P. Yimmuang)

Abstract In today's world, the detection of abnormal retinopathy from fundus photographs plays a significant role in non-destructive ocular diagnosis. Machine learning applications in ophthalmic imaging aid in the prescreening of ophthalmologists. This research proposes the utilization of twin parametric support vector machine and deep learning techniques for classifying normal and abnormal retinopathy from fundus photographs. The dataset used is an open source of fundus photographs from Shanggong Medical Technology (SMT) Co., Ltd. available on Kaggle.com. The procedure begins with the image processing phase, where all photographs undergo processing using five image processing techniques: ridge detection, ridge detection by Robert edge detection and hysteresis thresholding method, ridge detection by Sobel edge detection and hysteresis thresholding method, adaptive thresholding, and color contrast enhancement. The processed and non-processed images are then resized, and features are extracted using five algorithms of a convolutional neural network: ResNet50, InceptionV3, MobileNetV2, VGG19, and DenseNet201. For classification, both a classical support vector machine method (SVM) and a stochastic gradient method twin parametric support vector machine with a generalized pinball loss function (SG-TPSVM) are employed. It was observed that the classical SVM with the MobileNetV2 feature extraction algorithm achieved the highest accuracy, 92.98%, in a computational time of 4,764.4732 seconds. On the other hand, classification using the SG-TPSVM combined with all five feature selection algorithms resulted in slightly lower accuracy but significantly faster computational time. The accuracy of the models created by SG-TPSVM ranged from 84.74% to 87.64%. The model created by SG-TPSVM in conjunction with ResNet50 achieved the fastest creation time of only 2,142.0821 seconds, while maintaining an accuracy of 87.64%. Overall, SG-TPSVM provided a model with approximately 5.3% lower accuracy but was 55% faster than the classical approach.

MSC: 68U10; 92C50
Keywords: twin parametric support vector machine; abnormal retinopathy; fundus photographs

Submission date: 30.12.2022 / Acceptance date: 14.06.2023

*Corresponding author.

Published by The Mathematical Association of Thailand.

1. Introduction

In the past, the analysis of retinal lesions required the expertise of a medical professional to make a diagnosis. This analysis provides information on various diseases, such as diabetes, glaucoma, cataracts, macular degeneration, hypertension, pathological myopia, and other abnormalities. However, advanced artificial intelligence (AI) technology now enables computers to learn and assist doctors in identifying abnormalities in retinal photographs. This technology helps reduce expenses and time for patients while supporting doctors in improving the efficiency and accuracy of diagnoses. Numerous research projects are currently focused on utilizing image processing and AI techniques to detect eye diseases. In 2012, Ravudu, Jain, and Kunda conducted a review of image processing techniques for the automatic detection of eye diseases [1]. The techniques identified included image enhancement, image registration, image fusion, image segmentation, feature extraction, pattern matching, data classification, image morphology, and statistical measurements and analysis. Subsequently, Grinsven, Ginneken, Hoyng, Theelen, and Snchez utilized a fast convolutional neural network (CNN) method for hemorrhage detection in color fundus images [2]. In 2019, Pratt, Coenen, and Zheng applied a convolutional neural network to create a feature visualization for the classification of diabetic retinopathy [3]. A year later, Alyoubi, Shalash, and Abulkhair reviewed 33 articles related to the application of deep learning techniques in manipulating the diabetic retinopathy screening system [4]. They claimed that diabetic retinopathy can be detected based on the appearance of four lesion types on fundus images: microaneurysms, hemorrhages, soft exudates, and hard exudates. Although various deep learning methods, such as restricted Boltzmann machines, convolutional neural networks, autoencoders, and sparse coding, can be applied for diabetic retinopathy detection and classification, most articles reviewed by Alyoubi preferred methods based on CNNs. However, these methods can be resource-intensive and time-consuming on low-performance computers. This research proposes a faster method for the classification of normal and abnormal retinopathy from fundus photographs.

Support vector machine (SVM) is a popular classification algorithm that operates based on statistical learning theory, as introduced by Vapnik in the early 1990s [5]. SVMs have found wide application in real-world problems like image classification [6], text categorization [7], and more. The fundamental concept of SVM is to identify parallel hyperplanes that maximize the distance between two groups of data using optimization techniques. Twin support vector machine (TSVM) [8] is an extension of the traditional SVM method that creates two non-parallel hyperplanes positioned in the middle of the two most relevant data points. A TSVM requires approximately four times less computation time than SVM. Various types of TSVM have been proposed, including twin support vector machine regression (TSVR) [9, 10] and twin parametric support vector machine (TPSVM) [11, 12]. Wang [13] introduced the stochastic gradient twin support vector machine (SG-TSVM), which demonstrated faster computation speed than TSVM. Makmuang [14] proposed the stochastic gradient method twin parametric support vector machine with a generalized pinball loss function (SG-TPSVM), which exhibited less sensitivity to noise and faster performance compared to SG-TSVM.

Today, deep learning techniques are widely discussed and can be applied to solve problems in various fields. One such application is to uncover hidden image properties that are not visible in the original image. Convolutional Neural Network (CNN) is the

most popular deep learning architecture for analyzing medical images. CNN simulates human vision by examining smaller areas and aggregating clusters of sub-regions to infer visual features. Notably, in this research [15], CNN demonstrated its ability to discover remarkable features.

This research proposes the application of twin parametric support vector machine and deep learning techniques for the classification of normal and abnormal retinopathy from fundus photographs. The primary objective is to screen for normal and abnormal retinopathy using feature extraction techniques, support vector machine, stochastic gradient method twin parametric support vector machine with generalized pinball loss function, and deep learning techniques.

2. Background

In this chapter, we will review the fundamental knowledge of basic mathematics and machine learning pertaining to classification and deep learning techniques. The contents of this chapter will cover various topics, including feature extraction techniques, classification methods, and deep learning techniques.

2.1. Feature Extraction Techniques

In this research, five image processing techniques are employed to extract features from fundus photographs. The application of these techniques involves utilizing the following methods.

2.1.1. Edge Detection

Edges in digital images are considered as curves of discontinuities point in intensity values [16]. First and second-order derivatives are efficient methods to detect discontinuities. Here, the first-derivative is *gradient* of a 2-D function, $f(x, y)$, defined as

$$\nabla \mathbf{f} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

The magnitude of the gradient vector is $\|\nabla \mathbf{f}\| = \left[(\partial f/\partial x)^2 + (\partial f/\partial y)^2\right]^{1/2}$. The first-order derivatives can be approximated digitally by using different approximation. In the case that we have image neighborhood as

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

.

We call *Sobel edge detection*, if the magnitude $\|\nabla \mathbf{f}\|$ is approximated by

$$\begin{aligned} \|\nabla \mathbf{f}\| &= \Big\{[(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)]^2 \\ &\quad + [(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)]^2\Big\}^{1/2}. \end{aligned}$$

Also, we call *Robert edge detection*, if the magnitude is approximated by

$$\|\nabla \mathbf{f}\| = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2\right]^{1/2}.$$

2.1.2. Ridge Detection

In the image processing, one defines curves of points allocated on the local maxima of intensity data as a *ridge* [17]. However, there are several approaches to identifying the ridge from images. In this article, we consider Eigen values from the second-order partial derivative matrix, known as Hessian matrix, of the input image as the ridge detection.

2.1.3. Thresholding Method

Given two thresholds, T_1 and T_2, with $T_1 < T_2$. The thresholding method is the finding of ridge pixels, which are thresholded by so-called Hysteresis thresholding [16]. The Hysteresis thresholding is defined by

- Pixels with intensity greater than T_2 are said to be *edge pixels*;
- Pixels with intensity less than T_1 are said to be *non-edge pixels*;
- Pixels with intensity between T_1 and T_2 are classified to be *edge pixels* if they are connected with an *edge pixel*, otherwise they are classified as *non-edge pixels*.

On the other hand, if T_1 and T_2 are not fixed for whole digital image, the thresholding method is called the *adaptive thresholding method*.

2.1.4. Color Contrast Enhancement

If $I(x, y)$ represents an intensity of a pixel at coordinate (x, y) of an image. The image can be reconstructed by

$$I_{cce}(x, y; \sigma) = \alpha I(x, y) + \beta G(x, y; \sigma) * I(x, y) + \gamma,$$

where $*$ represents the convolution operator, $G(x, y; \sigma)$ is Gaussian filter with parameter σ, α, β and γ are adjustable parameters [2]. The function $I_{cce}(x, y; \sigma)$ introduced provides modified color fundus photographs, which may help in the observation of lesion.

2.2. Twin Support Vector Machine (TSVM)

To be easier to understand twin parametric support vector machine, support vector machine and twin support vector machine are presented.

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points. SVM model is generally formulated as a convex quadratic programming problem (QPP). Let us consider a training set $T = \left\{(x_i^T, y_i) \in \mathbb{R}^n \times \{1, -1\} : i = 1, 2, \ldots, m\right\}$, where $x_i^T \in \mathbb{R}$ is the vector attribute of training dataset with associated class labels, y_i for $i = 1, 2, 3, \ldots, m$. Each y_i can take one of two values, either 1 or -1. SVM find a hyperplane which is defined by

$$f(x) = w^T x + b.$$

The distance between the separating hyperplane and the training data sample nearest to the hyperplane is called margin, the goal of SVM is to find the optimal hyperplane with the maximum margine.

The SVM model is expressed in the form

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i, \\ \text{subject to } \; &y_i(x_i^T w + b) \geq 1 - \xi_i, \\ &\xi_i \geq 0, \quad i = 1, 2, 3 \ldots, m, \end{aligned} \tag{2.1}$$

where, $w \in \mathbb{R}^n$, $b \in \mathbb{R}$, C is the penalty parameter and ξ_i is slack variables. The decision function of the above formulation is based on the sign of $w^T x + b$ where x is assigned to class +1 if the value is positive otherwise it is assigned to class 1.

Loss Function or Cost Function is to calculate the error that the model predicted is different from the real then find the mean. In order to find the gradient of loss based on the different weights with backpropagation. Then use the gradient descent algorithm to make the loss smaller in the next trend.

The first loss function in SVM is Hinge loss, which is defined as follows

$$L_{hinge}(1 - y_i(x_i^T w + b)) = \max\{0, 1 - y_i(x_i^T + b)\},$$

where L_{hinge} is Hinge loss function. This loss function is a well known and commonly and defined by calculating constant of SVM. In 2014, Shwartz [18] used hinge loss in SVM and they showed that SVM problem can rewrited into unconstrained optimization problem as follows

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} L_{hinge}(1 - y_i(x_i^T w + b)),$$

where $C > 0$ is a constant parameter. Hinge loss is sensitive to noise and instability for resampling. To resolve this problem, Huang [19] used a pinball loss function to the SVM classifier. They showed that this method brings noise insensitivity. Pinball loss has been well studied in regression. However, the pinball loss has not been used for classification yet. Pinball loss function, which is defined as follows

$$L_\tau(u) = \begin{cases} u, & \text{if } u \geq 0, \\ -\tau u, & \text{if } u < 0, \end{cases}$$

where $u = 1 - y_i(x_i^T w + b)$ and $\tau \geq 0$ is a hyperparameter. Though the support vector machine using a pinball loss function is claimed to be more noise-insensitive than the support vector machine using a hinge loss function, in the process, it still loses sparsity, because the pinball loss functions sub-gradient is non-zero almost everywhere. To resolve this problem, in the same research, they introduced an ϵ-insensitive zone to Pin-SVM. They showed that ϵ-insensitive zone pinball SVM does not lose sparsity, because this loss function turns out to be zero in the range $[\tau, \epsilon]$. ϵ-insensitive zone pinball loss function, which is defined as follows

$$L_\tau^\epsilon(u) = \begin{cases} u - \epsilon, & \text{if } u \geq \epsilon, \\ 0, & \text{if } -\frac{\epsilon}{\tau} \leq u \geq \epsilon, \\ -\tau(u + \frac{\epsilon}{\tau}), & \text{if } u < -\frac{\epsilon}{\tau}, \end{cases}$$

where $u = 1 - y_i(x_i^T w + b)$ and $\epsilon, \tau \geq 0$ are hyperparameters.

In 2018, Rastogi [20] introduced a new loss function that was inspired by generalized pinball loss. This loss function is called generalized pinball loss that have noise insensitivity, sparseness, is stable for resampling and generalized over other loss function. generalized pinball loss function, which is defined as follows :

$$L_{\tau_1,\tau_2}^{\epsilon_1,\epsilon_2}(u) = \begin{cases} \tau_1(u - \frac{\epsilon_1}{\tau_1}), & \text{if } u > \frac{\epsilon_1}{\tau_1}, \\ 0, & \text{if } -\frac{\epsilon_2}{\tau_2} \leq u \geq \frac{\epsilon_1}{\tau_1}, \\ -\tau_2(u + \frac{\epsilon_2}{\tau_2}), & \text{if } u < -\frac{\epsilon_2}{\tau_2}, \end{cases}$$

where $u = 1 - y_i(x_i^T w + b)$ and $\tau_1, \tau_2, \epsilon_1, \epsilon_2 \geq 0$, are hyperparameters.

In 2007, Khemchandani [8] introduced a twin support vector machine (TSVM) that creates two non-parallel hyperplanes such that each hyperplane is closer to one of two

classes and is at least one far from the other. They showed that TSVM reduces the computational complexity of SVM. TSVM find a two non parallel planes hyperplane which is defined as

$$f_1(x) = w_1^T x + b_1 \quad \text{and} \quad f_2(x) = w_2^T x + b_2,$$

where, $w_1, w_2 \in \mathbb{R}^n$ and $b_1, b_2 \in \mathbb{R}$ are the weight vectors and biases of the hyperplane $f_1(x)$ and the hyperplane $f_2(x)$, respectively.

Let us consider a training set $T = \left\{(x_i^T, y_i) \in \mathbb{R}^n \times \{1, -1\} : i = 1, 2, \ldots, m\right\}$, where $x_i \in \mathbb{R}$ is the vector attribute of training dataset with associated class labels. Let m_1 be number of training data in class $+1$ and m_2 be number of training data in class -1, with $m_1 + m_2 = m$. Let A and B be $(m_1 \times n)$ and $(m_2 \times n)$ matrix, respectively. The optimization problems of TSVM with constraints are written as

$$\begin{aligned} \min_{w_1, b_1, \xi_2} \ & \frac{1}{2} \|Aw_1 + e_1 b_1\|^2 + c_1 e_2^T \xi_2, \\ \text{subject to } & -(Bw_1 + e_2 b_1) + \xi_2 \geq e_2, \\ & \xi_2 \geq 0, \end{aligned} \tag{2.2}$$

and

$$\begin{aligned} \min_{w_2, b_2, \xi_1} \ & \frac{1}{2} \|Bw_2 + e_2 b_2\|^2 + c_2 e_1^T \xi_1, \\ \text{subject to } & -(Aw_2 + e_1 b_2) + \xi_1 \geq e_1, \\ & \xi_1 \geq 0, \end{aligned} \tag{2.3}$$

where $c_1, c_2 > 0$ are the penalty parameters, $\xi_1 \in \mathbb{R}^{m_1}$, $\xi_2 \in \mathbb{R}^{m_2}$ are slack vectors, and $e_1 \in \mathbb{R}^{m_1}, e_2 \in \mathbb{R}^{m_2}$.

In the real world, the probability distribution of training data is not always distributed uniformly and SVM and TSVM assume that the probability distribution of training data is distributed uniformly. So, to solve this problem, Peng [12] introduced TPSVM. TPSVM generates two nonparallel hyperplanes similar to the TSVM. The TPSVM classifier is obtained by solving the following pair of quadratic programming problems as follows :

$$\begin{aligned} \min_{w_1, b_1, \xi_1} \ & \frac{1}{2} \|w_1\|^2 + \frac{\nu_1}{m_2} e_2^T (Bw_1 + e_2 b_1) + \frac{c_1}{m_1} e_1^T \xi_1, \\ \text{subject to } & Aw_1 + b_1 e_1 \geq 0 - \xi_1, \\ & \xi_1 \geq 0, \end{aligned} \tag{2.4}$$

and

$$\begin{aligned} \min_{w_2, b_2, \xi_2} \ & \frac{1}{2} \|w_2\|^2 + \frac{\nu_2}{m_1} e_1^T (Aw_2 + e_1 b_2) + \frac{c_2}{m_2} e_2^T \xi_2, \\ \text{subject to } & Bw_2 + b_2 e_2 \geq 0 - \xi_2, \\ & \xi_2 \geq 0, \end{aligned} \tag{2.5}$$

where $c_1, c_2, \nu_1, \nu_2 > 0$ are the penalty parameters, and ξ_1, ξ_2 are slack vectors.

In machine learning training, we want to know how much a change in weight gain / loss affects the gain / loss. But by using mathematics to help, we do not have to gradually move to check weight, we can use calculus to find the derivative slope of weight / loss out that the loss will increase when we move the weight. Gradient Descent (GD) means that we will gradually move every weight in the direction of the slope that is negative so that the loss is lowest.

Let us consider the unconstrained optimization problem of SVM with hinge loss

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} L_{hinge}(1 - y_i(x_i^T w + b)).$$

The idea of Gradient Descent method is to start at $w^{(k)}$ and move in the direction of the negative of $\eta_k \bigtriangledown f(w^{(k)})$, where η_k is a positive scalar called the step size. The procedure leads to the following iterative algorithm:

$$w^{(k+1)} = w^{(k)} - \eta_k \bigtriangledown f(w^k).$$

The gradient method is to requires that the function f be differentiable. For the function f is nondiferentiable. The gradient method using a subgradient of f at $w^{(k)}$, instead of the gradient.

Stochastic Gradient Descent (SGD) is an algorithm similar to GD, but it updates parameters only once per training cycle, which is computed quickly. Every time there is an update. The updated parameters have a high variance and result in loss function values varying in different intensities, but this method has a problem the lower we converge the resulting value becomes more variable and complex. On each iteration Stochastic Gradient Descent method as follow. Initially $w^{(1)}$ and positive parameters η, T. For $t = 1, \ldots, T$, on iteration t of the algorithm, we choose a random training sample (x_{i_t}, y_{i_t}) by picking an index $i_t \in \{i | i = 1, \ldots, m\}$. We then replace the objective function of unconstrained optimization problem of SVM with hinge loss with an approximation based on the training example (x_{i_t}, y_{i_t}), yielding:

$$f(w; i_t) \ := \frac{1}{2} \|w\|^2 + C L_{hinge}(1 - y_i(x_{i_t}^T w + b)).$$

Then consider the sub-gradient of the above approximate objective and update

$$w^{(t+1)} = w^{(t)} - \eta \bigtriangledown_w f(w; i_t)$$

Eventually, after T iterations, the stochastic gradient descent algorithm outputs the averaged vector $w = \frac{1}{T} \sum_{t=1}^{T} w^{(t)}$.

In 2020, Makmuang [14] proposed SG-TPSVM, inspired by stochastic gradient method, generalized pinball loss function, and twin parametric support vector machine. They showed that the accuracy performance of SG-TPSVM is better than the existing classifiers for different data scenarios and SG-TPSVM approach is leading to insensitivity from noise. The optimization problems of SG-TPSVM with constraints are written as

$$\min_{\omega_1} \frac{1}{2} \|\omega_1\|^2 + \frac{\nu_1}{m_2} \sum_{j=1}^{m_2} (\omega_1^T x_j^-) + \frac{c_1}{m_1} \sum_{i=1}^{m_1} L_{\tau_1,\tau_2}^{\epsilon_1,\epsilon_2}(0 - y_i^+(\omega_1^T x_i^+)), \tag{2.6}$$

$$\min_{\omega_2} \frac{1}{2} \|\omega_2\|^2 - \frac{\nu_2}{m_1} \sum_{i=1}^{m_1} (\omega_2^T x_i^+) + \frac{c_2}{m_2} \sum_{j=1}^{m_2} L_{\tau_3,\tau_4}^{\epsilon_3,\epsilon_4}(0 - y_j^-(\omega_2^T x_j^-)), \tag{2.7}$$

where $\omega_1 = [w_1^T, b_1]^T, \omega_2 = [w_2^T, b_2]^T, x = [x^T, 1]^T$ and w_1, w_2 are column vectors in $\mathbb{R}^n$, $b_1, b_2 \in \mathbb{R}, y^+ = 1, y^- = -1$, and $c_1, c_2, \nu_1, \nu_2 > 0$ are positive parameters.

2.3. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is multilayer structure network, and each layer produces a response. The layers extract the essential image feature and pass it to the next layer. CNN are suitable for 2 or 3-dimensional inputs, where the structure of a

convolutional neural network can take many forms by switching layers or adding different layers to it. The steps of a convolutional neural network are described below.

A convolutional layer is a layer that brings out the outstanding features of an image, such as borders, colors, shapes, etc. In this layer, filter matrices are used to extract elements of the image, and what we get from this step will be a matrix called Feature map. A convolutional layer comes in many forms, but the two most popular forms are narrow convolution and wide convolution. The Narrow convolution process is, if we have an input matrix of size $N \times N$ and a filter matrix of size $m \times m$, with this process we get a feature map of size $(N-m+1) \times (N-m+1)$. Wide convolution has the advantage of preventing the loss of data features at the edge of the input matrix. The process is, if we have an input matrix of size $N \times N$ and a filter matrix of size $m \times m$, with this process we get a feature map of size $(N+m-1) \times (N+m-1)$.

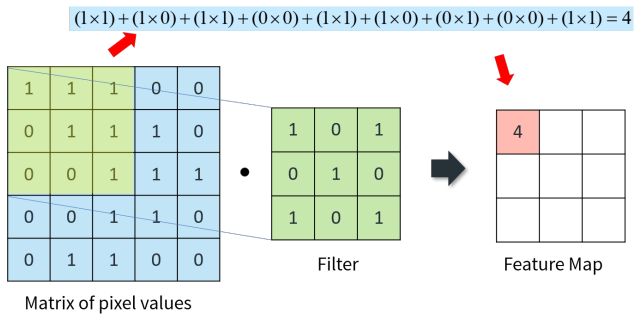


FIGURE 1. Convolutional Layer.

After the convolution is performed, the activation function will be implemented. ReLu is considered the most popular and widely used activation function. ReLU, full name Rectified Linear Unit, is used for non-linear operations, where ReLu's equation is very simple $f(x) = \max(0, x)$. The purpose of applying ReLu in the CNN process is to add non-linearity to the CNN.

The pooling layer is to reduce the size of the data that has been convoluted and is commonly used in conjunction with the convolution layer. There are two popular methods of pooling, maximum pooling and average pooling. Max pooling is a very popular method in today's research on neural network convolution. The method is to select the largest values from the submatrix and reconstruct them from these values to reduce the size of the matrix to be used in the next layer. Average pooling same as max pooling, but the result will be the mean of values in submatrix.

The fully connected layer is the layer in which the extracted data from different layers is connected to determine the correlation and weighting of values. Then go back and update your weight and loss calculations to get the best results. After the assembly of the convolution layer and the pooling layer, This layer is made up of sub-layers with many perceptions, where each perception has a line connecting all the perceptions in the previous layer and all the perceptions in the next layer.

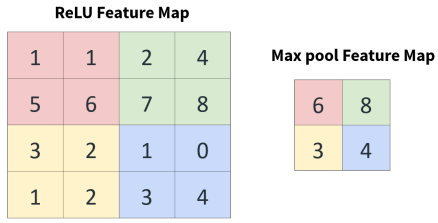


FIGURE 2. Pooling layer.

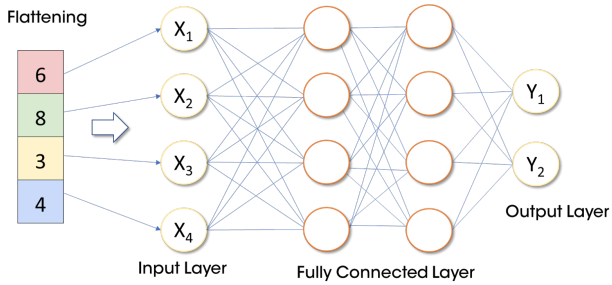


FIGURE 3. Fully connected layer.

3. METHODOLOGY

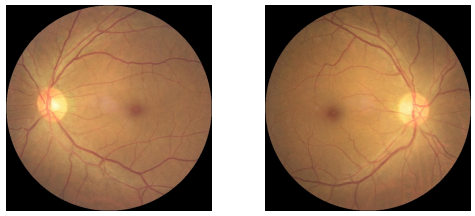
In this research, we obtained the dataset from user andrewmvd's Kaggle website. The dataset, known as Ocular Disease Intelligent Recognition, comprises eye images of 31,960 patients, including both left and right eyes. It is designed to represent a patient dataset collected by Shanggong Medical Technology Co., Ltd. from various Chinese hospitals and medical centers. The fundus images were captured using market-available cameras such as Canon, Zeiss, and Kowa, resulting in images with varying resolutions. Table 1 provides further details about the dataset. Figure 4 and Figure 5 depict examples of fundus photographs classified as normal and abnormal, respectively.

TABLE 1. Details of dataset.

Type of fundus photographs		Total
Normal	Abnormal	
14,370	17,590	31,960

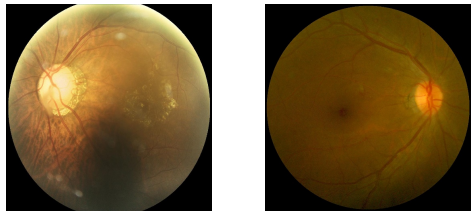
In the first step, we employed five feature extraction techniques, namely ridge detection, ridge detection by Robert edge detection and hysteresis thresholding method, ridge detection by Sobel edge detection and hysteresis thresholding method, adaptive thresholding, and color contrast enhancement. These techniques were utilized to augment the dataset. In the second step, we utilized the deep features extracted from the five most commonly used deep convolutional neural network (CNN) models: ResNet50, InceptionV3, VGG19, DenseNet201, and MobileNetV2. These models were applied to extract features from the images obtained in the first step. For the classification process, we employed SVM and

SG-TPSVM models. In the final step, we compared the performance of these methods based on five evaluation criteria: accuracy, time, precision, false recall, and F1 score. The results of the abnormal retinopathy detection from fundus photographs using deep learning techniques and SVM/SG-TPSVM are illustrated in Figure 6.



(A) Left. (B) Right.

FIGURE 4. The fundus photographs of normal.



(A) Left. (B) Right.

FIGURE 5. The fundus photographs of abnormal.

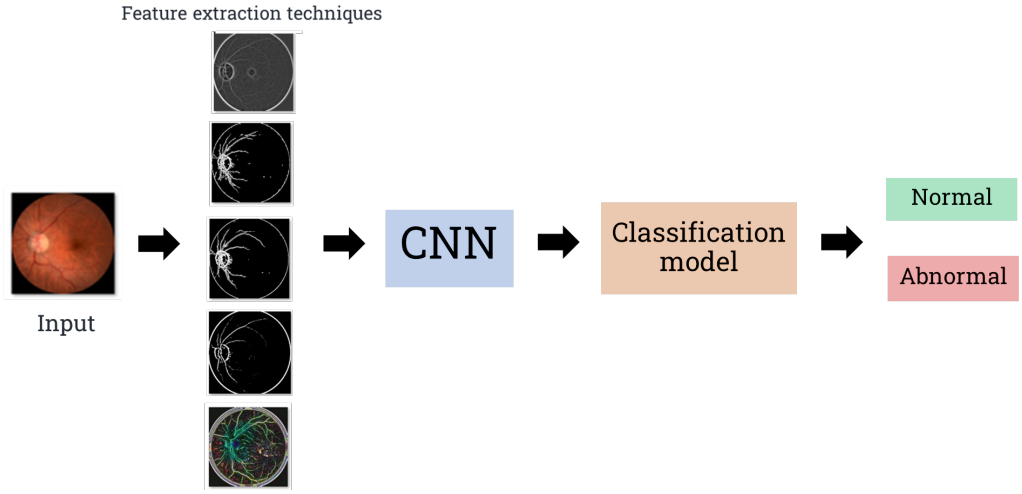
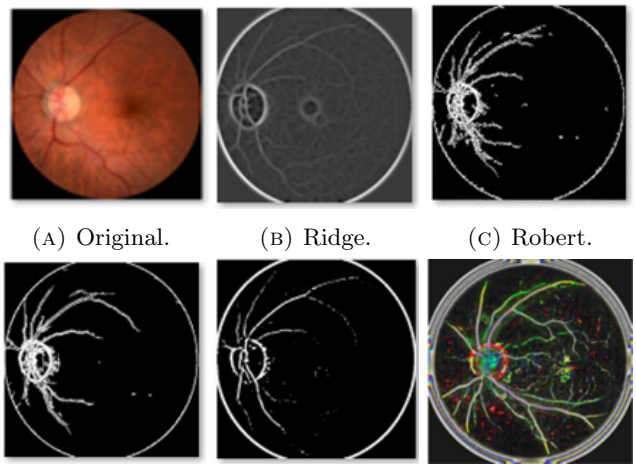


FIGURE 6. The abnormal retinopathy detection from fundus photographs based on deep learing techniques and SVM/SG-TPSVM.

4. Results and Discussion

In this research, the experiments were conducted using Python3 on a laptop running Windows 10 OS with an AMD Ryzen 7 4800HS processor, 16 GB of memory, and an NVIDIA GeForce GTX 2060 graphics card. To evaluate the performance of the models, we employed a 10-fold cross-validation technique for all experiments. The primary focus of this work was on achieving both computational speed and accuracy. In the first step, the number of datasets used in this research was increased by applying the five feature extraction techniques, resulting in a five-fold increase in the dataset size. An example of an image augmented using these five techniques is illustrated in Figure 7. The computational speed and total parameters of various CNN models in the feature extraction stage are displayed in Table 2. The accuracy and other performance measures of different deep CNN models, including the SG-TPSVM model, are recorded in Table 3 and Table 4, respectively.



(A) Original. (B) Ridge. (C) Robert.

(D) Sobel. (E) Thresholding. (F) Enhanced.

Figure 7. An example of an image augmented by the five techniques.

Table 2. Details of various CNN models.

CNN models	Feature vectors	Total parameters	Time (s)
ResNet50	2048	23,587,712	**2019.4497**
InceptionV3	2048	21,802,784	5216.7037
MobilenetV2	1280	2,257,984	3387.4330
VGG19	4096	139,570,240	4858.5858
DenseNet201	1920	18,321,984	7943.4544

TABLE 3. Performance measures (%) of different classification models based on SVM using deep features of various CNN models.

CNN models	Classification Model (SVM)				
	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Time (s)
ResNet50	92.49	91.81	**92.73**	92.27	1398.8622
InceptionV3	91.74	91.29	91.29	91.29	1445.9424
MobilenetV2	**92.98**	**92.14**	90.78	**92.43**	1377.0402
VGG19	91.61	91.04	91.51	91.27	**1344.0234**
DenseNet201	92.29	91.67	90.76	91.21	1743.387

TABLE 4. Performance measures (%) of different classification models based on SG-TPSVM using deep features of various CNN models.

CNN models	Classification Model (SG-TPSVM)				
	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Time (s)
ResNet50	**87.64**	**87.81**	86.73	**87.26**	122.6324
InceptionV3	84.74	84.31	84.31	84.31	122.7536
MobilenetV2	86.98	84.14	**86.78**	85.44	**111.5658**
VGG19	85.61	85.04	85.51	85.27	144.9792
DenseNet201	86.29	86.67	86.76	86.71	123.4142

From Table 2, it is evident that ResNet50 is the fastest among the five CNN models, with a computational speed of 2,019.4497 seconds. Looking at Table 3, we can see that the combination of MobilenetV2 with the SVM model achieves higher accuracy compared to other classification models, with an accuracy of 92.98%. Additionally, Table 3 reveals that the MobilenetV2-SVM combination exhibits higher precision and F1 Score than other classification models, with values of 92.14% and 92.43% respectively. The ResNet50-SVM model demonstrates the highest recall, with a recall rate of 92.73%. However, the SVM model computes the fastest when using the features obtained from the VGG19 feature selection technique, with a time of 1,344.0234 seconds. Referring to Table 4, it can be observed that the ResNet50-SG-TPSVM model delivers the best performance, with an accuracy of 87.64%, precision of 87.81%, and F1 Score of 87.26%. Furthermore, it is observed from Table 4 that the SG-TPSVM model exhibits the fastest computational speed when using the features from the MobilenetV2 techniques, taking only 111.5658 seconds.

5. CONCLUSION

In summary, this research applies five CNN models for feature selection and then utilizes two machine learning methods for the classification of abnormal retinopathy from fundus photographs. The CNN models used are ResNet50, InceptionV3, MobileNetV2, VGG19, and DenseNet201, with ResNet50 being the fastest feature selection technique. The two methods employed for classification are SVM and SG-TPSVM. Although the SVM method performs slightly better in classification, it takes more than 10 times longer to process than SG-TPSVM. When considering Tables 2-4, it is evident that using the ResNet50 technique for feature selection in combination with the SG-TPSVM method

yields the best results. This approach has a total processing time of only 2,142.0821 seconds, while delivering the highest accuracy and precision compared to using SG-TPSVM in conjunction with other feature selection methods. However, if the highest classification performance is desired, it is necessary to utilize the MobileNetV2 technique for feature selection in conjunction with the SVM method, despite the longer total processing time of 4,764.4732 seconds (2,622.3911 seconds longer). Although there is a slight decrease of approximately 5.3% in accuracy and precision, the overall processing time is reduced by approximately 55%. In reality, in the future, there may be an increase in the amount of data, including new techniques for extracting additional features from fundus photographs. This would enhance the efficiency of classification. However, having more data and more features may also result in increased processing time for classification, significantly so. Therefore, SG-TPSVM would be a suitable choice for processing in cases where there is a large volume of data and features.

Acknowledgements

This work was supported by the SUT Scholarships for Graduate Students (Kittibundit) at Suranaree University of Technology (SUT), Thailand. We would like to give many thanks for support from the School of Mathematics at SUT and Department of Mathematics at Naresuan University.

References

[1] M. Ravudu, V. Jain, M.M.R. Kunda, Review of image processing techniques for automatic detection of eye diseases, 2012 Sixth International Conference on Sensing Technology (ICST) (2012) 320–325.

[2] M.J.J.P. Grinsven, B. Ginneken, C.B. Hoyng, T. Theelen, C.I. Snchez, Fast convolutional neural network training using selective data sampling: application to hemorrhage detection in color fundus images, In IEEE Transactions on Medical Imaging 35 (5) (2016) 1273–1284.

[3] H. Pratt, F. Coenen, S.P. Harding, D.M. Broadbent, Y. Zheng, Feature visualisation of classification of diabetic retinopathy using a convolutional neural network, In CEUR Workshop Proceedings 2429 (2019) 23–29.

[4] L.W. Alyoubi, W.M. Shalash, M.F. Abulkhair, Diabetic retinopathy detection through deep learning techniques: A review, Informatics in Medicine Unlocked 20 (2020) 100377.

[5] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[6] E. Osuna, R. Freund, F. Girosit, Training support vector machines: an application to face detection, In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1997) 130–136.

[7] D. Isa, L.H. Lee, V.P. Kallimani, R. Rajkumar, Text document preprocessing with the Bayes formula for classification using the support vector machine, IEEE Transactions on Knowledge and Data Engineering 20 (9) (2008) 1264–1272.

[8] R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (5) (2007) 905–910.

[9] X. Peng, TSVR: an efficient twin support vector machine for regression, Neural Networks 23 (3) (2010) 365–372.

[10] Y. Xu, L. Wang, A weighted twin support vector regression, Knowledge-Based Systems 33 (2012) 92–101.

[11] P.Y. Hao, New support vector algorithms with parametric insensitive/margin model, Neural Networks 23 (1) (2010) 60–73.

[12] X. Peng, TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition, Pattern Recognition 44 (10–11) (2011) 2678–2692.

[13] Z. Wang, Y.H. Shao, L. Bai,, C.N. Li, L.M. Liu, N.Y. Deng, Insensitive stochastic gradient twin support vector machines for large scale problems, Information Sciences 462 (2018) 114–131.

[14] D. Makmuang, R. Wangkeeree, N. Khamsemanan, C. Nattee, A novel twin parametric support vector machine for large scale problem, Thai Journal of Mathematics 18 (3) (2020) 2107–2127.

[15] S. Asif, Y. Wenhui, H. Jin, Y. Tao, S. Jinhai, Classification of COVID-19 from chest X-ray images using deep convolutional neural networks, medRxiv (2020).

[16] R.C. Gonzalez, Digital Image Processing, Pearson Education India, 2019.

[17] T. Lindeberg, Edge detection and ridge detection with automatic scale selection, International Journal of Computer Vision 30 (2) (1998) 117–156.

[18] S. Shalev-Shwartz, S. Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.

[19] X. Huang, L. Shi, J.A. Suykens, Support vector machine classifier with pinball loss, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (5) (2013) 984–997.

[20] R. Rastogi, A. Pal, S. Chandra, Generalized pinball loss SVMs, Neurocomputing 322 (2018) 151–165.