STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
*Stat., Optim. Inf. Comput.*, Vol. **12**, Month 2024, pp 1149–1172.
Published online in International Academic Press (www.IAPress.org)

# A New SVM Solver Applied to Skin Lesion Classification

Jonatas Silva [1], Atécio Alves [2], Paulo Santos [3], Luiz Matioli [4]

[1]*Postgraduate Doctoral Program in Computer Science - UFMA/UFPI Association, Federal University of Piauí, Brazil*
[2]*Postgraduate Program in Mathematics, Federal University of Piauí, Brazil*
[3]*Department of Mathematics, Federal University of Delta do Parnaiba, Brazil*
[4]*Department of Mathematics, Federal University of Paraná, Brazil*

**Abstract** We present a unified framework for solving the nonlinear Support Vector Machines (SVM) training problems. The framework is based on an objective function approximation so that the Problem becomes separable, with low computational cost root-finding methods to solve the resulting subproblems. Because of the diagonalization of the objective function in the first stage of the framework, we named the new SVM solver DiagSVM. To test the performance of the DiagSVM, we reported preliminary numerical experiments with benchmark datasets. From the results, we chose the best combination used in the framework to solve the Skin Lesion Classification (SLC) problem. Since melanoma (skin cancer) is the most dangerous and deadliest disease that affects the skin, the application of the DiagSVM can be integrated into several Computer-Aided Diagnosis (CAD) systems to help them detect skin cancer and significantly reduce both morbidity and mortality associated with this disease. Machine learning (ML) and deep learning (DL) based approaches have been widely used to develop robust skin lesion classification systems. For the SLC problem, three pre-trained convolutional neural networks (CNN), Xception, InceptionResNetV2 and DenseNet201, were employed as feature extractors and their dimension was reduced using Principal Component Analysis (PCA), Kernel Principal Component Analysis (KPCA) and Independent Component Analysis (ICA). Finally, the samples were fed into two SVM solvers: DiagSVM and Libsvm. The experiment shows that using PCA, KPCA, or ICA, the SVM can perform better than without feature reduction. The classification performance of the proposed methodology is analyzed on the ISIC2017 and PH2 datasets. The benchmark and SLC results indicate a promising proposal for accuracy, specificity and sensitivity metrics.

**Keywords** Separable diagonal approximation, Support vector machines (SVMs), Root-finding methods, Skin lesion classification, Convolutional Neural Network (CNN)

**AMS 2020 subject classifications** 47N10, 68W25

**DOI:** 10.19139/soic-2310-5070-2005

## 1. Introduction

Among the state-of-the-art methods for solving machine learning problems are Support Vector Machines (SVMs) [20]. SVM is one of the most used machine learning algorithms in statistical learning problems such as spam filtering, text classification, handwriting analysis, and face and object recognition, among others [3]. It combines excellent performance with a solid mathematical foundation that involves solving a quadratic programming optimization problem, as shown by [1] and [2].

We can emphasize the two main drivers for the evolution of the SVM solvers: generalization error and large or small-scale learning [13]. They explain the compromise between the types of generalization errors and emphasize the limit on the number of training samples and the limit on the computational time, which, in some aspects,

$^*$Correspondence to: Jonatas Silva (Email: jonatas.iw@gmail.com). Postgraduate Doctoral Program in Computer Science - UFMA/UFPI Association, Federal University of Piauí, Brazil

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 2024 International Academic Press

because the computational time is always limited in practice, large-scale problems always percolate into small-scale learning problems.

The training stage for SVM involves a dense convex quadratic optimization problem (QP) at its core. Solving this optimization problem is computationally expensive, primarily due to the dense Hessian matrix. Solving the QP with a general-purpose QP solver would result in the time taken scaling cubically $O(n^3)$ with the number of data points [64, 65]. Such complexity means that general-purpose optimization solvers cannot solve large-scale SVM training problems. Several schemes have been developed in the literature where a solution is built by solving a sequence of small-scale problems. Some other popular implementations and strategies to reduce SVM training time can be seen in [5, 20, 8, 11, 24, 25, 26, 27, 10, 28];

We propose a framework to solve the QP in the dual SVM formulation to mitigate this issue. First, the framework defines a separable quadratic approximation of the QP objective function, as presented in [16] and [18]. Each subproblem is addressed in its dual form, which can be solved using root-finding methods; see, for example, [14], [9] and their references.

In the context of skin lesion classification (SLC), the survey [31] summarized from 2011 to 2022 several studies carried out on skin lesion segmentation (SLS) and classification. The survey's observations show a clear interest in applying machine learning (ML) and deep learning (DL) methods to solve SLC problems. Furthermore, most ML methods applied to SLC were performed with SVM. The survey also notes that SLS and SLC are rapidly growing areas of study and that the number of publications has increased annually for both tasks, especially after 2016.

Most of the interest in SLS and SLC comes from the fact that skin cancer incidence and mortality rates are increasing worldwide [32]. Melanoma is a severe form of skin cancer because it is more likely to spread to other parts of the body. It becomes hard to treat once melanoma spreads beyond the skin to other body parts. Thus, early diagnosis is essential for the patient's survival since detecting malignant skin lesions at the initial stage can be cured successfully [32].

Dermatologists are the dermoscopic method, a noninvasive technique, to diagnose the skin lesion. This method analyses lesion structures with the naked eye using the magnification tool. Later, this information can be used to diagnose the structure by employing popular diagnostics methods [35], for example, asymmetry, border, color, and diameter (ABCD) rule [44], seven-Point checklist [45], and Menzies technique [43]. All three methods have the same first step: identification of lesion, either as melanoma or non-melanoma, using a score-based (ABCD rule and 7-Point checklist) and features-based (Menzies) approaches. The traditional dermoscopy method can increase the melanoma detection rate from 10–27% [40]. However, it depends on the formal training of dermatologists [41].

One reliable solution to this problem is to apply computerized or digital dermoscopy analysis techniques, commonly known as computer-aided detection (CAD) systems. In dermatology, [39] describes the prospective benefits of digital imaging. Afterwards, CAD systems have put forth various efforts to improve clinical melanoma diagnostics [35]. Such image processing for automatic skin lesion classification can be divided into three main stages: lesion segmentation, features extraction and lesion classification. Segmentation and feature extraction are the key steps and significantly influence the outcome of the classification results [31].

The Convolutional neural network (CNN) can be used in two ways. The first is to design a classification model, and the second is to extract features using Transfer Learning [48]. In our work, we will use transfer learning using pre-trained networks; we extract features from three different architectures of pre-trained networks: Xception, InceptionResNetV2 and DenseNet201. Given the high number of features that will be extracted from the CNN models, it is wise to make a feature reduction to keep only the best and most relevant ones. Then, to reduce the feature dimension, we used three popular methods: Principal Component Analysis (PCA), Inference Component Analysis (ICA), and Kernel Principal Component Analysis (KPCA). After normalization and reduction, the extracted features can be used to improve the classification accuracy rate.

This paper aims to present a new SVM solver with low computational cost and compare the performance of PCA, KPCA and ICA for feature reduction in SVM applied to SLC. We extracted the dermoscopic images from this study's ISIC2017 and PH2 datasets. The CNN models were applied directly to the images without a segmentation step. In this stage, we tested three pre-trained CNN models as feature extractors: Xception, InceptionResNetV2 and DenseNet201. The original higher-dimensional inputs will be transformed into other lower-dimensional features using one of the three feature reduction methods. The new features are then used as SVM inputs to solve the SLC problem.

We can highlight some contributions of this research, such as:

- Flexibility in using different methods in the proposed SVM solver (in the diagonal approximation step and in the stage of solving the approximate subproblems);
- Combining various alternatives in DiagSVM to find good training results in the SVM problems;
- DiagSVM enables training medium-sized volumes of data, keeping the metrics competitive with the state-of-the-art method Libsvm;
- Comparison of PCA, KPCA and ICA reveals that such feature reduction performs better than without feature reduction in the context of SVM applied to SLC.

We organize this paper as follows. Section 2 presents the SVM problem, which is essential for understanding the context of our proposal. Section 3 presents the proposed framework, addressing it in the SVM training. Section 4 presents our methodology for the SLC problem. Section 5 reports numerical results for the benchmark dataset, and section 6 reports numerical results for the SLC problem. We show the conclusions in Section 7.

## 2. Support Vector Machines (SVMs)

SVM is a supervised machine-learning technique developed by [1] and then extended in [2] to solve classification and regression problems. The basic idea of an SVM in a simple binary classification problem is to search for the hyperplane that is the farthest to the closest training data points from both sides of the hyperplane. This process has two phases: training and testing. In the training phase, the machine is trained to find a hyperplane that separates the given data samples into two classes with known negative or positive labels. After the machine is trained, the training model is extracted, and the testing phase is carried out. In the testing phase, the SVM model predicts which class label a new unseen test sample should have [23].

An SVM has unique characteristics used to implement the proposed framework. One characteristic is that the solution to a classification problem is obtained by only a few samples, called *support vectors* (SVs) [1], that determine the maximum margin separating hyperplane [23]. Another characteristic of SVM is performing nonlinear mapping without knowing the mapping function using kernels' predefined functions for calculating the inner product of mapping functions [2]. Another characteristic of SVMs is the simple structure of constraints.

To clarify, given the set of labelled examples

$$D = \{(x_i, y_i),\ \ i = 1, ..., n,\ \ x_i \in \mathbb{R}^m,\ y_i \in \{-1, 1\}\},$$

the SVM algorithm performs classification of new examples $x \in \mathbb{R}^m$ by using decision function $F : \mathbb{R}^m \to \{-1, 1\}$ of the form

$$F(x) = sign\left(\sum_{i=1}^{n} \alpha_i^* y_i K(x, x_i) + b^*\right), \tag{1}$$

where $K \in \mathbb{R}^{n\times n}$ denotes a particular kernel function, and $\alpha^* = (\alpha_1^*, ..., \alpha_n^*)$ is the solution of SVM QP problem, which appears in two forms: primal form and dual form. The proposed framework is addressed to the dual formulation.

The dual optimization problem addressed by SVMs is as follows:

$$\begin{aligned} \text{minimize } f(\alpha) = & \frac{1}{2}\alpha^T P\alpha - \alpha^T e && (2)\\ \text{subject to } \quad & y^T\alpha = 0 && (3)\\ & 0 \le \alpha \le Ce, && (4) \end{aligned}$$

where $\alpha \in \mathbb{R}^n$, $C \in \mathbb{R}$, $y \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$, $e = (1, 1, \dots, 1) \in \mathbb{R}^n$ and the matrix $P \in \mathbb{R}^{n \times n}$ is symmetrical, positive semi-definite with components defined as

$$P_{i,j} = y_i y_j K(x_i, x_j), \quad i, j = 1, 2, ..., n. \tag{5}$$

Once the vector $\alpha^*$ is computed, the quantity $b^*$ in equation (1) is easily derived. A training example $x_i$ is called *support vector* (SV) if the corresponding $\alpha_i$ is nonzero and a *bound support vector* (BSV) if $\alpha* = C$. The Kernel function measures the similarity or the distance between vectors $x_i$ and $x_j$ [2]. Examples of well-known kernel functions are presented in Table 1.

Table 1. Some Kernel functions

| Name | Function(x,z) | Parameters |
|---|---|---|
| Linear | $x^T z$ | - |
| Polynomial | $(x^T z + p)^d$ | $d \in \mathbb{N},\ p \in \mathbb{R}_+$ |
| Gaussian | $\exp(\sigma\|\|x - z\|\|^2)$ | $\sigma \in \mathbb{R}_+$ |
| Exponential | $\exp(\sigma x^T z)$ | $\sigma \in \mathbb{R}_+$ |
| Sigmoid | $\tanh(kx^T z + \theta)$ | $k$ e $\theta < 0$ |

## 3. SVM framework proposal (DiagSVM)

The proposed SVM framework, DiagSVM, is motivated by the necessity of a fast and flexible SVM QP solver. Flexibility is characterized by the possibility of separable approximations in the first stage and the many existing root-finding algorithms that can be used in the second stage of the framework. The algorithms studied for implementing the framework's second stage are characterized by simplicity and low computational cost. In Figure 1, we present the SVM block diagram to summarize each stage of the algorithm.

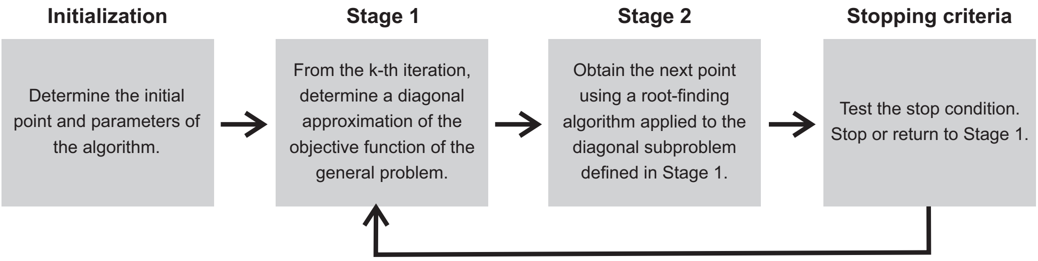

Initialization
Determine the initial point and parameters of the algorithm.
Stage 1
From the k-th iteration, determine a diagonal approximation of the objective function of the general problem.
Stage 2
Obtain the next point using a root-finding algorithm applied to the diagonal subproblem defined in Stage 1.
Stopping criteria
Test the stop condition. Stop or return to Stage 1.

Figure 1. DiagSVM block diagram

Next, we describe in more detail the characteristics of our scheme. In Subsection 3.1, we introduce the approximation structure used in the first stage of the DiagSVM and which methods we use as a separable quadratic approximation. In Subsection 3.2, we define the root-finding methods we are looking for, presenting some reference examples. In Subsection 3.3, we define the final DiagSVM formulation and present its algorithm.

### *3.1. Separable Quadratic Approximations*

Our study of finding a way to approximate the Hessian matrix of the objective function of the SVM QP problem by a diagonal matrix started with the work of [16], applied to solve unconstrained optimization problems. The approach is considered an alternative to the conjugates gradient-based methods because it requires less storage of variables and is computationally efficient. The method applies the steepest descent technique in successive spherical quadratic approximations of the objective function so that no linear search is necessary to solve the minimization problem. [16] show that the method is convergent when applied to general positive-definite quadratic functions and works well with ill-conditioned problems. More precisely, they considered the following unconstrained optimization problem:

$$\text{minimize } \{f(x) : x \in \mathbb{R}^n\}, \tag{6}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is the objective function.

Assuming the differentiability of $f$, [16] define a diagonal approximation of $f$ at the point $x^k$, named $\bar{f}_k(x)$, by:

$$\bar{f}_k(x) = \frac{1}{2}(x - x^k)^T Z_k (x - x^k) + \nabla f(x^k)^T (x - x^k) + f(x^k), \tag{7}$$

and $Z_k = diag(z_k, z_k, ..., z_k) = z_k I$. Before we show how to find $z_k$, we will first convert the Equation (7) to the format we want to solve in Equation (2).

$$\begin{array}{rcl} \bar{f}_k(x) & = & \frac{1}{2}(x - x^k)^T Z_k (x - x^k) + \nabla f(x^k)^T (x - x^k) + f(x^k) \\ & = & \nabla f(x^k)^T x - \nabla f(x^k)^T x^k + \frac{1}{2}\left[(x - x^k)^T Z_k x - (x - x^k)^T Z_k x^k\right] \\ & = & \nabla f(x^k)^T x - \nabla f(x^k)^T x^k + \frac{1}{2}\left[x^T Z_k x - x^{k^T} Z_k x - x^T Z_k x^k - x^{k^T} Z_k x^k\right] \\ & = & \nabla f(x^k)^T x - \nabla f(x^k)^T x^k + \frac{1}{2} x^T Z_k x - \frac{1}{2} x^{k^T} Z_k x - \frac{1}{2} x^T Z_k x^k - \frac{1}{2} x^{k^T} Z_k x^k. \end{array} \tag{8}$$

After removing the terms $\nabla f(x^k)^T x^k$ and $\frac{1}{2} x^{k^T} Z_k x^k$ which correspond to the constant terms in (8), the function $\bar{f}_k(x)$ is rewritten as

$$\bar{f}_k(x) = \frac{1}{2} x^T Z_k x - \left[-\nabla f(x^k) + Z_k x^k\right]^T x. \tag{9}$$

The Equation (9) together with the constraints (3) and (4), forms the separable quadratic subproblems generated in each iteration of the framework. The functions at (7) and (9) are always separable and convex as long as the terms $z_k$ are strictly positive, thus having a positive-definite and diagonal Hessian matrix.

Now, to obtain the value of $z_k$, we adopted in the framework two different approaches, the first one named Spherical diagonal approximation proposed in [16], and the second one named Quasi-Cauchy approximation proposed in [18].

- In [16], the Spherical diagonal approximation is defined following the equation (7). They declare that forcing $\bar{f}_k(x^{k-1}) = f(x^{k-1})$, is possible to obtain $z_k$ as follows:

  $$z_k = \frac{2[f(x) - f(x^k) - \nabla f(x^k)^T (x - x^k)]}{\|x - x^k\|^2}. \tag{10}$$

  Thus, the separable approximation of the objective function at (7) and (9) uses a multiple of the identity matrix.

- In [18], the Quasi-Cauchy approximation is presented, in which it is related to the ”weak” Quasi-Newton approximation of [19] where an additional constraint replaces the full matrix with a diagonal matrix. The Quasi-Cauchy approximation can be obtained through the following expression:

  $$H_k = (s_k^T y_k / s_k^T s_k) I, \tag{11}$$

  where $s_k = x^k - x^{k-1}$, $y_k = \nabla f(x^k) - \nabla f(x^{k-1})$ and $I$ an identity matrix. $H_k$ is the well-known Oren-Luenberger scaling diagonal matrix. Rewriting the equation (11) in the same format of equation (10), we can obtain $z_k$ from the Quasi-Cauchy approximation as follow:

  $$z_k = \frac{(x^k - x^{k-1})^T (\nabla f(x^k) - \nabla f(x^{k-1}))}{(x^k - x^{k-1})^T (x^k - x^{k-1})} \tag{12}$$

### *3.2. Root-finding methods*

The root-finding methods mentioned in this work are those whose purpose is to solve the Separable Quadratic Knapsack Problem (SQKP), which in turn has the following form:

$$(SQKP) : \left\{ \begin{array}{ll} \text{minimize } f(x) = & \frac{1}{2}x^T Dx - a^T x \\ \text{subject to} & b^T x = c \\ & l \leq x \leq u, \end{array} \right. \tag{13}$$

where $x \in \mathbb{R}^n$ is the decision variable; $D = \text{diag}(d)$ is a diagonal matrix with positive entries $d_i$; $a$, $b$ are vectors in $\mathbb{R}^n$, $c$ is a scalar; and $l < u$ are lower and upper bounds. In Problem (13), reformulating $x$ as a dependent function of $\lambda$, we have a problem finding the Lagrange multiplier of the equality constraint. The proof of the following three theoretical results can be found in [9].

*Theorem 1*
A vector $x^* \in \mathbb{R}^n$ is a minimum of the Problem (13), if and only if, there are Lagrange multipliers $\lambda_* \in \mathbb{R}$, $v^* \in \mathbb{R}^n_+$ and $w^* \in \mathbb{R}^n_+$ such that

$$L'_x(x^*, \lambda_*, v^*, w^*) = \nabla f(x^*) + \lambda_* \nabla g(x^*) + (\nabla r(x^*))' v^* + (\nabla s(x^*))' w^* = 0 \tag{14}$$

and beyond that,

$$v_i r_i(x_i^*) = 0, \; i = 1, \cdots, n, \tag{15}$$

$$w_i s_i(x_i^*) = 0, \; i = 1, \cdots, n, \tag{16}$$

where $f(x) = \frac{1}{2}x^T Dx - a^T x$, $g(x) = b^T x - c$, $r(x) = l - x$ and $s(x) = x - u$.

*Lemma 3.1*
According to the conditions of the KKT Theorem, (14) is equivalent to

$$\left. \begin{array}{ll} 0 \geq \dfrac{d_i u_i - a_i}{b_i} + \lambda_* & , \text{ if } x_i^* = u_i \\ 0 = \dfrac{d_i x_i^* - a_i}{b_i} + \lambda_* & , \text{ if } x_i^* \in (l_i, u_i) \\ 0 \leq \dfrac{d_i l_i - a_i}{b_i} + \lambda_* & , \text{ if } x_i^* = l_i, \end{array} \right\} \tag{17}$$

for all $i = 1, 2, \ldots, n$.

Motivated by (17), the function $x$ is defined depending on the variable $\lambda$.

$$x : \mathbb{R} \to \mathbb{R}^n, \; \lambda \mapsto x(\lambda)$$

with

$$x_i(\lambda) = \left\{ \begin{array}{ll} l_i & , \text{ if } \; \lambda \geq \dfrac{a_i - d_i l_i}{b_i} \\ \dfrac{a_i - \lambda b_i}{d_i} & , \text{ if } \; \dfrac{a_i - d_i u_i}{b_i} < \lambda < \dfrac{a_i - d_i l_i}{b_i} \\ u_i & , \text{ if } \; \lambda \leq \dfrac{a_i - d_i u_i}{b_i}. \end{array} \right. \tag{18}$$

*Theorem 2*
A vector $x^* \in \mathbb{R}^n$ is a unique solution of the optimization problem (13), if and only if, exist $\lambda_* \in \mathbb{R}$ such that $x(\lambda_*)$ defined in (18) satisfies

$$g(x(\lambda_*)) = 0. \tag{19}$$

From the above development, notice that to solve SQKP, we can use root-finding methods that determine zeros of the one-dimensional non-differentiable function $\tilde{g}\colon \mathbb{R} \to \mathbb{R}$, where

$$\tilde{g}(\lambda) = \sum_{i=1}^{n} b_i x_i(\lambda) - c.$$

Some works with possibilities of root-finding strategies for the implementation in the second stage of the proposed framework are listed:

1. In [9], a fixed-point strategy for a stratified sampling problem (non-quadratic but separable) is compared with classic methods from the literature: Bisection, Regula falsi and Secants.
2. In [14], a Newton-type method for SQKP is studied, and comparisons with Secant, Median Search and Variable Fixing methods are included.
3. In [15], a basic pegging algorithm is developed to solve SQKP and proposes several implementational choices for the solution of the reduced Problem and the updates.
4. In [29], we developed a fixed-point method to solve SQKP based on [9].

In addition to the works mentioned above, more methods and algorithms to solve the SQKP problem can be found in the surveys [33, 34].

### 3.3. *DiagSVM algorithm*

Using the results of the previous sections, we are now ready to formulate our framework algorithm to solve the SVM QP problem presented in Section 2. Below, we detailed each step of our algorithm:

Step 1: Obtain the labelled data to train the model: $X \in \mathbb{R}^{n\times m}$ and $y \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$.

Step 2: Define parameters $C > 0$ and Kernel function $K$.

Step 3: Generate the SVM QP problem:

$$\begin{array}{ll} \text{minimize } f(\alpha) = & \frac{1}{2}\alpha^T P\alpha - \alpha^T e \\ \text{subject to} & y^T\alpha = 0 \\ & 0 \le \alpha \le Ce, \end{array} \tag{20}$$

where, $P \in \mathbb{R}^{n\times n}$ according with (5), $\alpha \in \mathbb{R}^n$, $e \in \mathbb{R}^n$ is a vector of ones, $y$ is the same of the Step 1.

Step 4: Set $k = 0$ and $\alpha^k$ = $e$;
Set $\nabla f^k = P\alpha^k - e$, $z^k = \|\nabla f^k\| e$, and $a^k = -\nabla f^k + z^k * \alpha^k$;

Step 5: Use a root-finding algorithm to solve the Problem (21).

$$\begin{array}{ll} \text{minimize } \tilde{f}(\alpha^k) = & \frac{1}{2}\alpha^T Z^k\alpha^k - \alpha^k a^k \\ \text{subject to} & y^T\alpha^k = 0 \\ & 0 \le \alpha^k \le Ce \end{array} \tag{21}$$

where $Z^k = diag(z^k)$.

Step 6: Set $k = k + 1$, let $\alpha^k = \tilde{f}(\alpha^{k-1})$;
From Theorem 2, $\alpha^k$ is a unique solution of the problem (21), if $\lambda^k$ satisfies (19). Thus, we use $\lambda^k$ as stopping criteria, so that, if $\lambda^k - \lambda^{k-1} < \epsilon$ then $\alpha^* = \alpha^k$ and STOP; otherwise, continue to next step.

```
Step 7: If using SQA:
        Set $f^k = \frac{1}{2}\alpha^{kT} P\alpha^k - \alpha^{kT} e$ and $\nabla f^k = P\alpha^k - e$;
        Let $s = \|x^k - x^{k-1}\|^2$ and $v = 2(f^{k-1} - f^k - \nabla f^k s)$;
        Set $z^k = \max(10^{-20}, \frac{s}{v})$, $a^k = -\nabla f^k + z^k * \alpha^k$, and $z^k = z^k e$.
        Go to Step 5.
        If using QCA:
        Set $\nabla f^k = P\alpha^k - e$;
        Let $s = x^k - x^{k-1}$ and $v = \nabla f^k - \nabla f^{k-1}$;
        Set $z^k = \max(10^{-20}, \frac{s^T v}{s^T s})$, $a^k = -\nabla f^k + z^k * \alpha^k$, and $z^k = z^k e$.
        Go to Step 5.
```

### *3.4. DiagSVM complexity*

Consider the algorithm in section 3.3. Steps 1 and 2 define some variables necessary to run the algorithm, which can be processed in a constant time ($O(1)$). In Step 3, the SVM QP problem is generated. Since we used the RBF kernel (see Table 1), the time complexity of this step can achieve $O(n^2m)$ [23], where n is the number of data points, and m is the dimension of input space. However, in the literature, there is some fast implementation research for the RBF kernel function [67, 68]. In Step 4, the most expensive evaluation is the matrix-vector operation, which requires $O(n^2)$ time complexity. In step 5, we used $O(n)$ root-finding methods to solve each subproblem. Step 6 checks the stopping criteria, which requires only constant time. Step 7 generates the new subproblem approximation of the general objective function by the last iterate. Spherical diagonal approximation and Quasi-Cauchy approximation require $O(n^2)$ time complexity because, similarly to Step 4, the most expensive evaluation is the matrix-vector operation

## 4. Skin Lesion Analysis (SLA)

As mentioned previously, SLA plays a significant role in skin cancer prevention. A growing number of deaths due to melanoma has been noticed globally; this type of malignant in Figure 2 is deemed to be the most aggressive one compared to other lesions.

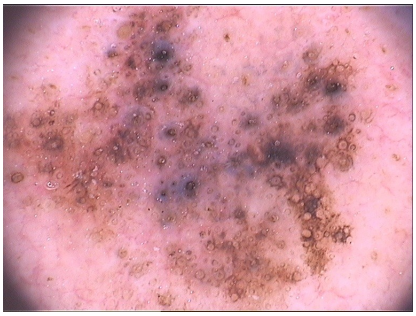


Figure 2. example of melanoma skin lesion

Many researchers proposed using image processing for skin lesion detection [42]. This process is commonly divided into three steps: image segmentation to identify the lesion, features extraction and lesion classification. Segmentation separates suspicious damage from the normal skin area to extract further features from the lesion region. The algorithms for SLS have been improved in recent years. Many papers set the OTSU thresholding technique to stand apart from other techniques due to their simplicity [31]. Algorithms based on active contour models have been continuously suggested for the segmentation of skin lesions in images where the initial curves are being moved to the object of interest through convenient deformation [42]. Region-based algorithms like K-means [46] have also been widely used to segment skin lesion images. However, CNN-based segmentation networks have been widely applied to medical images, outperforming traditional image processing methods relying on manual features [31].

The type of features used in the classification step is the key to successful classification of SLC. For that, handcrafted and deep-learning features exist [31]. ABCD rules, Gray Level Co-occurrence Matrix (GLCM), Local binary pattern (LBP), and Histogram of Oriented Gradient (HOG) are standard handcrafted features in the skin lesion classification [48]. The ABCD rule characterizes the skin lesion based on dermatology criteria: shape, color and symmetry. This acronym (ABCD) stands for Asymmetry, Border irregularity, Color variegation, and Diameter greater than 6 mm [47]. GLCM is a statistical measure that computes the joint probability of occurrence of grey levels considering two pixels spatially separated by a fixed vector. Several measures may be computed based on the GLCM, such as mean, correlation, homogeneity, contrast, energy, dissimilarity, kurtosis, variance, entropy, maximum probability, inverse difference, angular second moment, and standard deviation [50]. The local Binary Pattern (LBP) method is used for texture analysis. It has been found to be a very efficient texture operator. Edge histogram, Gabor and Histogram of Oriented Gradients (HOG) methods are used for shape feature extraction [49].

As discussed in [31], handcrafted features are often challenging due to the appearance of various intrinsic and extrinsic noises in dermoscopic images. On the other hand, the CNN-based approaches provide excellent SLC results and boost diagnostic procedure rates while being end-to-end systems. Many authors employed pre-trained CNN models and fine-tuned them with the skin lesion datasets [31]. The feature extraction step is carried out automatically by CNN, which can be effective in the classification process. A CNN model usually comprises three layers, namely convolution, pooling, and fully connected layers [54]. Convolution and pooling layers play a feature extraction role, while the fully connected layer generally acts as a classifier. Although CNN can classify images, its combination with other classifiers might increase the classification efficiency [57]. [51] proposed an automatic computerized method for skin lesion classification, which employs deep features from CNNs. They use three pre-trained deep models. The extracted features were used to train SVM classifiers and got an 83.83% accuracy rate. The database used is the ISIC challenge [52]. [55] presented a hybrid classification framework for dermoscopy image assessment by combining CNN, Fisher vector (FV) and linear SVM.

Although CNN automatically extracts the features from the images, selecting features can provide better performance and accuracy to the systems. Therefore, selecting the best subset of features for classification is essential. In [56], the authors proposed an automated system for SLC through transfer learning-based deep neural network (DCNN) features extraction and kurtosis-controlled principle component (KcPCA) based optimal features selection. They used pre-trained ResNet for feature extraction and selected the best features for training the SVM classification model. In [57], a system is proposed to detect melanoma automatically using CNNs and image texture feature extraction. The features dimension was also reduced using kernel principal component analysis (KPCA) to improve the classification performance. The method was evaluated on ISIC2016, ISIC2019, and PH2.

The classification phase consists of recognizing and interpreting information about skin lesions based on the extracted and selected characteristics. The classification process is usually performed by randomly dividing the available image samples into training and testing sets. The training stage consists of developing a classification model to be employed by one or more classifiers based on the samples of the training set. Each sample includes features extracted from a provided image and its corresponding class value, named as input data to the classifier. The testing step measures the model's metrics learned by the training step over the test set. [31] shows that from 2011 to 2022, SVM and K-nearest Neighbors (KNN) were the top ML models used for SLC.

In this study, our focus is to validate the DiagSVM presented in the previous sections but also contribute with a modern methodology to the SLA. Figure 3 illustrates a block diagram of this approach, which employs the following steps: dataset, data augmentation, feature extraction, feature engineering and classification. Each of them is described in the following subsections.

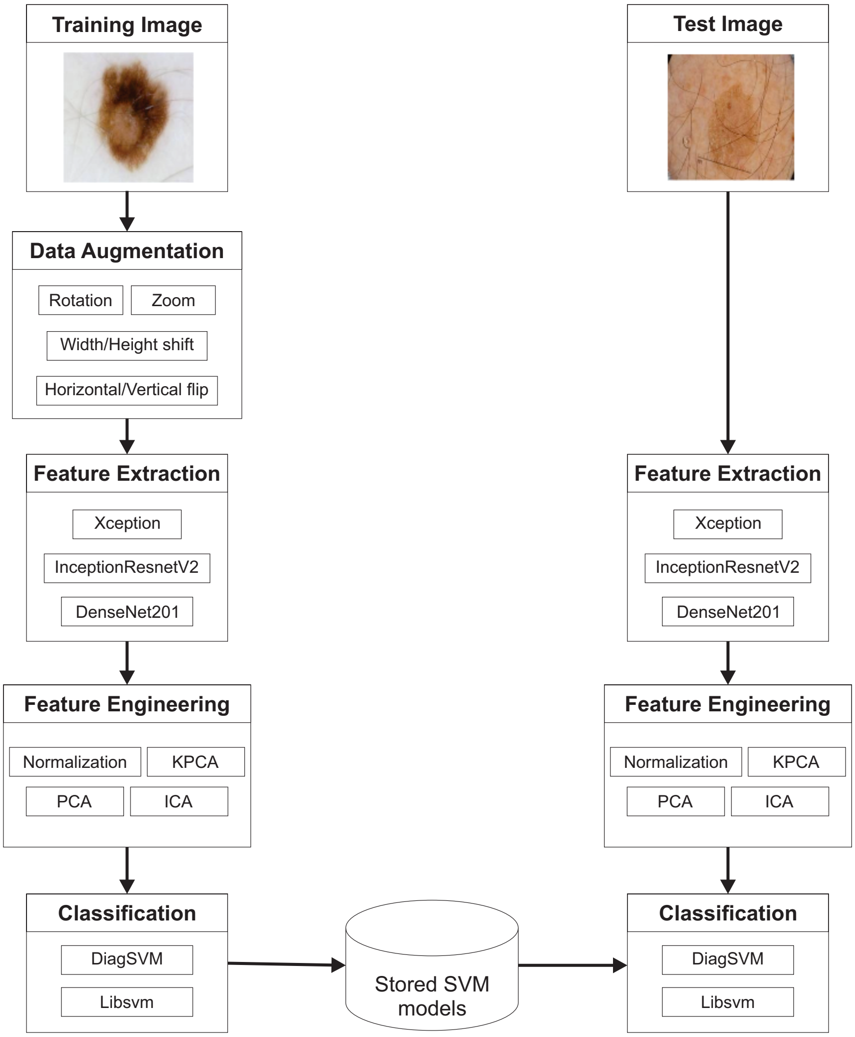


Figure 3. Block diagram of the proposed methodology

### *4.1. Dermoscopic images dataset*

The International Skin Imaging Collaboration (ISIC) datasets have become a leading repository for researchers in machine learning for medical image analysis, especially in skin cancer detection and malignancy assessment [59]. The ISIC2017 dataset (Codella et al., 2017) contains 2000 training and 600 test images, a total of 2600 images. Ground truth and patient metadata are provided for both training and test sets, indicating if the lesion is one of three class groups: melanoma (ME), benign nevi (BN), and seborrheic keratosis (SK). The patient's approximate age and gender are also provided as additional metadata. Among the training images, 374 belong to class ME, 254 to class SK and 1372 to class BN in JPG format. Among the test images, 117 belong to class ME, 90 to class SK, and 393 to class BN, also in JPG format. We defined two binary classification problems for testing our methodology using the ISIC2017 dataset according to Table 2.

In Table 2, two binary classification problems are presented; each Problem comprises the classes filled in the columns Class 1 and Class 2. For example, the problem ”ME, BN - SK” is composed of the merging of the classes ME and BN to represent the positive class (see section 2), and the class SK was used to represent the negative class. The same idea is applied to the problem ”ME, SK - BN”. The other columns correspond to the number of samples in each class used in the training and test steps.

Table 2. Description of classes used in binary classification problems for the ISIC2017 dataset before the augmentation step.

| Problem name | Class 1 | Class 2 | number of class 1 training samples | number of class 2 training samples | number of class 1 test samples | number of class 2 test samples |
|---|---|---|---|---|---|---|
| ME, BN - SK | ME, BN | SK | 1746 | 254 | 510 | 90 |
| ME, SK - BN | ME, SK | BN | 628 | 1372 | 207 | 393 |

PH2 is a dermoscopic skin lesion information database. PH2 dataset comprises many manual skin lesion segmentation images for clinical diagnosis and research. Dermatologists, or skin disease specialists, identified different skin lesion dermoscopic structures [30]. The dermoscopic images in this dataset were obtained under the same conditions through the Tuebinger Mole Analyzer system using a magnification of 20x. They are 8-bit RGB color images with a resolution of 768x560 pixels. The database contains 200 dermoscopy images, where 80 are common nevi (CN), 80 are atypical nevi (AN), and 40 are melanomas (ME). Since the PH2 dataset has only 200 samples, we split each class into 70% training samples and 30% test samples. Similarly to the problems generated by the ISIC2017 dataset, below in Table 3, we present the two binary classification problems using the PH2 dataset after splitting the data.

Table 3. Description of classes used in binary classification problems for the PH2 dataset before the augmentation step.

| Problem name | Class 1 | Class 2 | number of class 1 training samples | number of class 2 training samples | number of class 1 test samples | number of class 2 test samples |
|---|---|---|---|---|---|---|
| CN, AN - ME | CN, AN | ME | 112 | 28 | 48 | 24 |
| AN, ME - CN | AN, ME | CN | 84 | 56 | 36 | 12 |

In Table 3, we follow the same methodology used to build the problems of Table 2; two binary classification problems were formed from the PH2 dataset.

As we can see in Table 2 and 3, the number of samples in the training step for the ISIC2017 dataset is large but very imbalanced for all problems. In contrast, for the PH2 dataset, all problems have few samples in each class. In the next step, we will present the data augmentation process to obtain more samples in some problem classes to solve both issues.

### 4.2. Data augmentation

Data augmentation is commonly used to overcome the limited number of images and reduce overfitting [58]. Our methodology used horizontal/vertical flip, width/height shift, zoom and rotation augmentation to train the machine learning models. Figure 4 depicts the sample images after horizontal/vertical flip augmentation [58].

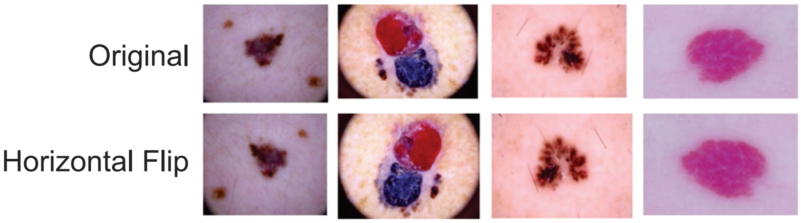


Figure 4. Horizontal Flip augmentation

As a result, models with data augmentation are more likely to learn more differentiating characteristic features than models without data augmentation. In Table 4, we present the number of samples we obtained after data augmentation for all the problems we performed. The augmentation was applied only for the training samples.

Table 4. Description of classes used in binary classification problems for the ISIC2017 and PH2 dataset after the augmentation step.

| Problem name | Class 1 | Class 2 | number of class 1 training samples | number of class 2 training samples |
|---|---|---|---|---|
| ME, BN - SK | ME, BN | SK | 2046 | 1774 |
| ME, SK - BN | ME, SK | BN | 1985 | 2023 |
| CN, AN - ME | CN, AN | ME | 1556 | 1148 |
| AN, ME - CN | AN, ME | CN | 1730 | 1170 |

### 4.3. Feature extraction

CNN is a specific type of feed-forward neural network with a stack of convolutional layers, each followed by pooling layers to extract features from the input data and produce a set of high-level feature maps at each level of convolution. The feature map information is summarized using pooling layers to reduce the parameters, followed by a fully connected layer to produce the final classification [63]. We extracted features from three different architectures of pre-trained networks: Xception, InceptionResNetV2 and DenseNet201.

- Xception: [60] proposes a deep convolutional neural network architecture inspired by Inception. Inception modules consist of a series of convolutions in parallel with different kernel sizes to extract features. Instead of Inception, Xception architecture has depth-wise separable convolutions and 36 convolutional layers to extract essential features.

- InceptionResNetV2: [61] have shown that InceptionResNetV2 significantly accelerates the training of Inception networks with the help of residual connections. InceptionResNetV2 has introduced significant simplification to the inception blocks. It is a variation of the InceptionV3 model, which borrows some ideas from ResNet models. Residual connections allow the training of much deeper neural networks, leading to better performance.

- DenseNet201: [62] introduced the Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward fashion. DenseNet201 is a complex model with 201 layers.

### 4.4. Feature engineering

Principal component analysis (PCA) is a well-known method for feature reduction. By calculating the eigenvectors of the covariance matrix of the original inputs, PCA linearly transforms a high-dimensional input vector into a low-dimensional one whose components are uncorrelated [36]. Nonlinear PCA has also been developed by using different algorithms. Kernel principal component analysis (KPCA) is one type of nonlinear PCA developed by generalizing the kernel method into PCA [37]. Specifically, KPCA first maps the original inputs into a high-dimensional feature space using the kernel method and then calculates PCA in the high-dimensional feature space. The linear PCA in the high-dimensional feature space corresponds to a nonlinear PCA in the original input space. Another linear transformation method, Independent Component Analysis (ICA), has been used [38]. Instead of transforming uncorrelated components, ICA attempts to achieve statistically independent components in the transformed vectors. ICA was initially developed for blind source separation. Later, it has been generalized for feature reduction.

### *4.5. Classification*

After extracting the features in all the images using CNN models, the proposed work uses SVM to generate a model to learn how to classify skin lesions. This step is referred to as the training step. In the test step, we used the unseen samples to validate the model and compare the predicted results with the actual outputs. The SLC task was formulated as a binary classification problem, so now we can apply the DiagSVM algorithm proposed in section 3. We also apply the Libsvm [8] in the proposed SLC problem to compare both results.

## 5. Numerical experiments for the benchmark datasets

To illustrate the numerical behaviour of the proposed SVM solver, we performed some numerical tests using a laptop Intel Core i5 1.6 GHz, 8 GB of RAM, dual-core processor, with code implemented in MATLAB 2021b.

Comparisons were made with Libsvm algorithm [8]. We used grid search to find the optimum SVM parameters. To compose the grid search, we selected four different values of $C$ (1E+0, 1E+1, 1E+2, 1E+3) and $\sigma$ (1E-1, 1E-2, 1E-3, 1E-4). Hence, for each classification problem, 16 (4 x 4) combinations were performed.

Table 5 shows the basic descriptions of the performance metrics used in this study. From Table 5, $TP$, $TN$, $FP$, and $FN$ refer to true positive, true negative, false positive and false negative, respectively.

Table 5. Performance metrics

| Performance measure | Description | Formula |
|---|---|---|
| Accuracy $(A_c)$ | It is defined as the number of correctly classified skin cancer images to the total number of images tested. | $A_c = \frac{TP+TN}{TP+TN+FP+FN}$ |
| Specificity $(S_p)$ | It is the ratio between the number of TN decisions by the classifier to the actual N cases tested. | $S_p = \frac{TN}{TN+FP}$ |
| Sensitivity $(S_e)$ | It is the ratio between the number of TP decisions by the classifier to the actual P cases tested. | $S_e = \frac{TP}{TP+FN}$ |

As mentioned briefly, we will use the results obtained in benchmark datasets to select the best DiagSVM combination and apply it to the skin cancer classification problem (Section 6).

### *5.1. Benchmark datasets description*

One of the datasets we used in this section is the MNIST from [21]. This database for handwritten digit recognition has been widely used to validate new techniques in machine learning, as in [14]. [22] presented a survey which summarizes the top state-of-the-art contributions reported on the MNIST. All the other datasets were extracted from the official Libsvm website (`https://www.csie.ntu.edu.tw/~cjlin/libsvm/`). The Libsvm authors already provide the data in a readable format, and they also offer a function to transform the data into matrices and vectors with dimensions relative to the $n$ samples and $p$ attributes.

The datasets were linearly scaled to the intervals [-1, 1] or [0, 1]. The main advantage of scaling is preventing some attributes' numerical value ranges from being much larger than the range of values of other attributes. Another advantage is to avoid numerical difficulties during calculation when using the Kernel function [17]. For these experiments, only scaled data were used. In this case, the transformation of an attribute $r$ of any dataset can be described within the range $[a, b]$ in the form

$$r_{scaled} = (a-b)\frac{r-\min(r)}{\max(r)-\min(r)} + a. \tag{22}$$

Table 6 describes the selected data, illustrating the number of samples (n) that the dataset has and the number of attributes that are considered (p). The Cross-Validation (CV) technique will be applied in the experiments performed to obtain a disjoint test set. The technique separates a percentage of the original dataset - in this work, 70% was used - for a training set of the model, while the complement is destined for the test phase.

Table 6. Benchmark datasets

| Name | n | p |
|---|---|---|
| MNIST | 1568 | 784 |
| heart | 270 | 13 |
| australian | 690 | 14 |
| fourclass | 862 | 2 |
| ionosphere | 351 | 34 |
| sonar | 208 | 60 |
| splice | 1000 | 60 |

More details about the heart, australian, fourclass, ionosphere, sonar, and splice datasets can be found in [8].

The MNIST dataset is composed of 1568 images from number 0 to 9 (see Figures 5 and 6). To use it in our experiments, a preprocessing of each image was necessary. Firstly, to have a binary classification problem, the dataset was divided into two categories:

- Number zero (0)

Figure 5. Number 0

- Other numbers (1-9)

Figure 6. Numbers from 1 to 9

According to [21], the original black and white (bi-level) images from MNIST were size normalized to fit in a 20x20 pixel box while preserving their aspect ratio. The resulting images contain grey levels due to the anti-aliasing technique used by the normalization algorithm. The images were centred in a 28x28 image by computing the centre of mass of the pixels and translating the image to position this point at the centre of the 28x28 field. After reshaping the 28x28 matrix, a vector of size 784 was obtained.

### *5.2. Results from benchmark datasets*

As part of the experiments, for the first stage of the proposed framework to solve the Problem (2) subject to (3) and (4), the spherical approximation [16] and quasi-Cauchy approximation [18] were used. For the second stage of the framework, the Fixed Point method [29], the Secant, Bisection, and Regula falsi methods according to [9] were used. Such combinations were developed to exemplify the framework's ability to enable the combination of diagonal approximation techniques in the first stage and any other algorithm in the literature capable of solving the Problem (13) in the second stage. Still based on [16], for the first subproblem we set $z_0 = |\nabla \tilde{f}_0|$ and $\alpha_0$ any feasible point.

As CV was applied, ten seeds were randomly selected on the data to generate different training and test data subsets. Thus, the results will be the average values of the ten problems generated.

In these experiments, we chose the Gaussian kernel function for the tests, whose formula is described in Table 1. As mentioned briefly, the value of the parameter C and the parameter $\sigma$, used in the Gaussian Kernel function, were chosen according to the grid search method and CV. The best combination of $\sigma$ and C for each dataset is presented next.

In the experiments, we named the algorithms so that it is easier to understand which combination was used to compose the DiagSVM in the experiment. Table 7 lists what the acronyms that make up each algorithm mean.

Table 7. Algorithm acronym meaning

| Acronym | Meaning |
|---|---|
| Sp | Spherical approximation |
| Qc | Quasi-Cauchy approximation |
| fp | Fixed Point method |
| sc | Secant method |
| rf | Regula falsi method |
| bs | Bisection method |
| Libsvm | Library for Support Vector Machines |

In Table 8, 9, 10, 11 and 12, we present the results from quasi-Cauchy approximation with fixed point and secant method, quasi-Cauchy approximation with regula falsi and bisection, spherical approximation with fixed point and secant method, spherical approximation with regula falsi and bisection and Libsvm respectively. The results indicate that the proposal obtained results similar to Libsvm for all benchmark datasets. In the tables, we added columns C and $\sigma$, indicating that from the grid search, which parameter value the framework (DiagSVM) obtained the best metrics.

Table 8. Results from quasi-Cauchy approximation with fixed point and secant method

| | Qc-fp | | | | | Qc-sc | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Name | C | $\sigma$ | $A_c$ | $S_p$ | $S_e$ | C | $\sigma$ | $A_c$ | $S_p$ | $S_e$ |
| mnist | 1E+2 | 1E-2 | 98.74 | 97.73 | 100.00 | 1E+1 | 1E-2 | 98.77 | 99.48 | 98.78 |
| heart | 1E+1 | 1E-2 | 84.32 | 82.35 | 89.36 | 1E+0 | 1E-2 | 84.57 | 91.43 | 76.09 |
| australian | 1E+1 | 1E-1 | 85.17 | 79.59 | 91.74 | 1E+0 | 1E-2 | 85.65 | 76.85 | 92.93 |
| german | 1E+1 | 1E-2 | 73.73 | 55.65 | 85.41 | 1E+0 | 1E-2 | 74.83 | 56.52 | 78.35 |
| fourclass | 1E+2 | 1E+1 | 99.73 | 98.84 | 100.00 | 1E+2 | 1E+1 | 99.77 | 98.84 | 100.00 |
| ionosphere | 1E+1 | 1E-1 | 95.43 | 93.55 | 95.35 | 1E+1 | 1E-1 | 95.43 | 93.55 | 95.35 |
| sonar | 1E+1 | 1E-1 | 90.16 | 96.30 | 82.86 | 1E+1 | 1E-1 | 90.16 | 96.30 | 82.86 |
| splice | 1E+3 | 1E-2 | 80.93 | 92.45 | 80.00 | 1E+3 | 1E-2 | 80.85 | 92.45 | 80.00 |

Table 8 presents the comparison between quasi-Cauchy approximation with fixed point (Qc-fp) and quasi-Cauchy approximation with secant method (Qc-sc). The results for the accuracy metric are very similar for both methods in all datasets. For the mnist, heart and german datasets, the Qc-fp presented a better sensitivity.

Table 9. Results from quasi-Cauchy approximation with regula falsi and bisection method

| | Qc-rf | | | | | Qc-bs | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Name | C | $\sigma$ | $A_c$ | $S_p$ | $S_e$ | C | $\sigma$ | $A_c$ | $S_p$ | $S_e$ |
| mnist | 1E+1 | 1E-2 | 98.77 | 99.48 | 98.78 | 1E+1 | 1E-2 | 98.77 | 99.48 | 98.78 |
| heart | 1E+0 | 1E-2 | 84.57 | 91.43 | 76.09 | 1E+0 | 1E-2 | 84.57 | 91.43 | 76.09 |
| australian | 1E+0 | 1E-2 | 85.65 | 76.85 | 92.93 | 1E+0 | 1E-2 | 85.65 | 76.85 | 92.93 |
| german | 1E+0 | 1E-2 | 74.83 | 56.52 | 78.35 | 1E+0 | 1E-2 | 75.00 | 56.52 | 78.35 |
| fourclass | 1E+2 | 1E+1 | 99.77 | 98.84 | 100.00 | 1E+2 | 1E+1 | 99.77 | 98.95 | 100.00 |
| ionosphere | 1E+1 | 1E-1 | 95.24 | 93.55 | 95.35 | 1E+1 | 1E-1 | 95.43 | 93.55 | 95.35 |
| sonar | 1E+1 | 1E-1 | 90.16 | 96.30 | 82.86 | 1E+1 | 1E-1 | 89.68 | 96.30 | 82.86 |
| splice | 1E+3 | 1E-2 | 80.85 | 92.45 | 80.00 | 1E+3 | 1E-2 | 80.68 | 92.16 | 77.61 |

Table 10. Results from spherical approximation with fixed point and secant method

| | Sp-pf | | | | | Sp-sc | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Name | C | $\sigma$ | $A_c$ | $S_p$ | $S_e$ | C | $\sigma$ | $A_c$ | $S_p$ | $S_e$ |
| mnist | 1E+3 | 1E-2 | 98.94 | 98.98 | 98.70 | 1E+3 | 1E-2 | 98.94 | 98.98 | 98.70 |
| heart | 1E+1 | 1E-2 | 84.81 | 90.48 | 84.62 | 1E+1 | 1E-2 | 84.81 | 90.48 | 84.62 |
| australian | 1E+0 | 1E-1 | 85.89 | 79.61 | 92.31 | 1E+1 | 1E-1 | 86.38 | 88.37 | 90.08 |
| german | 1E+0 | 1E-2 | 76.20 | 66.67 | 78.05 | 1E+0 | 1E-2 | 76.10 | 67.92 | 78.14 |
| fourclass | 1E+0 | 1E+1 | 99.88 | 100.00 | 100.00 | 1E+0 | 1E+1 | 99.88 | 100.00 | 100.00 |
| ionosphere | 1E+1 | 1E-1 | 94.10 | 95.77 | 97.06 | 1E+1 | 1E-1 | 94.00 | 95.77 | 97.06 |
| sonar | 1E+2 | 1E-1 | 90.00 | 96.67 | 84.38 | 1E+3 | 1E-1 | 88.55 | 86.67 | 87.50 |
| splice | 1E+1 | 1E-2 | 79.92 | 84.62 | 77.27 | 1E+1 | 1E-2 | 80.00 | 84.62 | 77.27 |

Table 11. Results from spherical approximation with regula falsi and bisection method

| | Sp-rf | | | | | Sp-bs | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Name | C | $\sigma$ | $A_c$ | $S_p$ | $S_e$ | C | $\sigma$ | $A_c$ | $S_p$ | $S_e$ |
| mnist | 1E+3 | 1E-2 | 98.94 | 98.98 | 98.70 | 1E+3 | 1E-2 | 98.94 | 98.98 | 98.70 |
| heart | 1E+1 | 1E-2 | 84.81 | 90.48 | 84.62 | 1E+1 | 1E-2 | 84.81 | 90.48 | 84.62 |
| australian | 1E+1 | 1E-1 | 86.33 | 89.53 | 90.91 | 1E+1 | 1E-1 | 86.52 | 89.53 | 90.91 |
| german | 1E+0 | 1E-2 | 76.13 | 67.92 | 78.14 | 1E+0 | 1E-2 | 76.13 | 67.92 | 78.14 |
| fourclass | 1E+0 | 1E+1 | 99.92 | 100.00 | 100.00 | 1E+0 | 1E+1 | 99.92 | 100.00 | 100.00 |
| ionosphere | 1E+1 | 1E-1 | 94.10 | 95.77 | 97.06 | 1E+1 | 1E-1 | 94.19 | 95.77 | 97.06 |
| sonar | 1E+2 | 1E-1 | 90.00 | 96.67 | 84.38 | 1E+2 | 1E-1 | 89.84 | 96.67 | 84.38 |
| splice | 1E+1 | 1E-2 | 80.00 | 84.62 | 77.27 | 1E+1 | 1E-2 | 80.00 | 84.62 | 77.27 |

Table 9 shows that quasi-Cauchy approximation with regula falsi (Qc-rf) and quasi-Cauchy approximation with bisection (Qc-bs) presented a very similar performance for all metrics in all datasets, except for the sensitivity in splice dataset, where Qc-rf obtained a better result than Qc-bs. As expected, the parameter C and $\sigma$ from Qc-pf, Qc-sc, Qc-rf and Qc-bs presented in Tables 8 and 9 were identical, since the diagonal approximation is the same for all algorithms.

Similarly to quasi-Cauchy approximation, when we compare the spherical approximation combined with the root-finding algorithms in Table 10 and 11, the accuracy performance was very similar in all combinations, except for the sonar dataset, where the spherical approximation with fixed point (Sp-fp), regula falsi (Sp-rf) and bisection (Sp-bs) presented a better accuracy and specificity than the spherical approximation with secant method (Sp-sc). The parameters C and $\sigma$ were identical in all spherical approximation combinations.

Table 12. Results from Libsvm

| Name | C | $\sigma$ | $A_c$ | $S_p$ | $S_e$ |
|---|---|---|---|---|---|
| mnist | 1E+3 | 1E+1 | 99.34 | 99.23 | 96.25 |
| heart | 1E+0 | 1E-2 | 83.58 | 90.91 | 72.92 |
| australian | 1E+1 | 1E+1 | 86.62 | 83.93 | 88.42 |
| german | 1E+0 | 1E+1 | 76.63 | 72.50 | 75.77 |
| fourclass | 1E+3 | 1E-2 | 99.73 | 100.00 | 100.00 |
| ionosphere | 1E+1 | 1E-2 | 95.33 | 92.96 | 97.06 |
| sonar | 1E+3 | 1E+1 | 86.29 | 88.00 | 83.78 |
| splice | 1E+1 | 1E-1 | 81.86 | 80.95 | 83.64 |

Compared to Libsvm, the most significant difference in the results appears in the sonar dataset, where Qc-fp, Qc-sc, Qc-rf, Sp-pf and Sp-rf obtained about 90% accuracy against 86.29% from Libsvm. The Libsvm was superior to the proposed SVM solver in the splice and mnist dataset with a small margin.

Finally, in Figure 7, we present a stacked bar with the accuracy summation of all algorithms for all datasets. The figure proves that the proposed SVM solver performed similarly to Libsvm regarding accuracy metric. Based on Figure 7, since there is no significant difference between the algorithms, we choose the Qc-fp algorithm to compose the DiagSVM in the numerical experiments for the SLC problem presented in the next section.

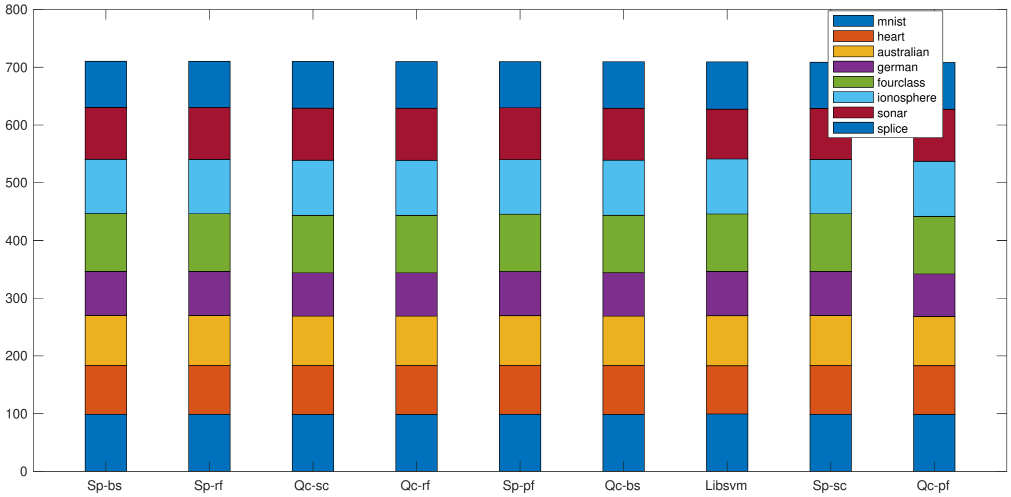


Figure 7. Stacked bar with the accuracy summation of all algorithms for all benchmark datasets.

## 6. Numerical experiments for the SLC problem

The studies and implementation presented in this section have been implemented with Python 3.8 installed on a computer Intel Core i5-9400 CPU 2.90GHz, 24 GB of RAM; Radeon RX Vega (VEGA10, DRM 3.42.0, 5.15.0-94-generic, LLVM 12.0.0), with 8 GB memory.

The proposed approach will be evaluated and validated using the databases described in Section 4: PH2 and ISIC2017 dataset. Each dataset consists of three classes of lesion images. For both datasets, we manipulated the classes to generate two binary classification problems as shown in Table 2, 3 and 4. The performance of classification models is validated by four metrics such as accuracy, specificity and sensibility. All metrics were described in Table 5.

### 6.1. *ISIC2017 Dataset*

As previously mentioned, this publicly available dataset contains a total of 2000 training JPG images, including 374 to class ME, 254 to class SK and 1372 to class BN, and 600 test images, 117 belonging to class ME, 90 to class SK, and 393 to class BN. To balance the dataset, we updated the number of training samples using augmentation. Table 4 shows the number of samples for each class after augmentation.

In Table 13, 14 and 15, we presented the results for each combination of CNN model extractor and feature reduction using DiagSVM as well as Libsvm.

Table 13. ISIC2017 classification results using Xception as a feature extractor.

| | ”ME, BN - SK” problem | | | | | | ”ME, SK - BN” problem | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ |
| DiagSVM | 1E+0 | 1E-4 | - | 85.33 | 91.05 | 51.16 | 1E+0 | 1E-4 | - | 75.0 | 63.9 | 80.76 |
| DiagSVM-PCA | 1E+1 | 1E-4 | 50 | **86.83** | 88.83 | **62.22** | 1E+1 | 1E-4 | 250 | 75.17 | 62.29 | **83.52** |
| DiagSVM-KPCA | 1E+1 | 1E-1 | 500 | **86.83** | **90.74** | 57.75 | 1E+1 | 1E-1 | 250 | 75.0 | 65.08 | 79.56 |
| DiagSVM-ICA | 1E+0 | 1E-3 | 100 | 85.5 | 89.83 | 52.17 | 1E+0 | 1E-3 | 250 | 73.17 | 60.18 | 81.02 |
| Libsvm | 1E+1 | 1E-4 | - | 86.0 | 92.43 | 53.06 | 1E+1 | 1E-4 | - | **76.5** | 64.47 | 83.87 |
| Libsvm-pca | 1E+0 | 1E-4 | 50 | 86.17 | 87.79 | 60.0 | 1E+0 | 1E-4 | 1000 | **75.5** | **65.46** | 80.3 |
| Libsvm-kpca | 1E+1 | 1E-1 | 500 | **86.83** | **90.74** | 57.75 | 1E+1 | 1E-1 | 250 | 74.67 | 64.71 | 79.18 |
| Libsvm-ica | 1E+0 | 1E-4 | 100 | 86.17 | 90.51 | 54.79 | 1E+0 | 1E-4 | 250 | 75.0 | 73.17 | 75.47 |

Table 13 shows that the algorithms DiagSVM-PCA, DiagSVM-KPCA, and Libsvm-KPCA presented the same and the better accuracy metric for the ”ME, BN - SK” problem using Xception as a feature extractor. We highlighted the best results for each CNN model in each problem. In the ”ME, SK - BN” problem, the DiagSVM-PCA obtained the best sensitivity. However, Libsvm-PCA presented better accuracy and specificity. In columns $C$ and $\sigma$, we presented the SVM parameters, and in column Nc, the number of components used in the feature reduction algorithm. The $C$, $\sigma$ and Nc parameter values were obtained from the grid search method. In Table 13, 14 and 15, the DiagSVM and Libsvm algorithms do not use feature reduction, thus, the Nc column is not filled.

Table 14. ISIC2017 classification results using InceptionResNetV2 as feature extractor.

| | ”ME, BN - SK” problem | | | | | | ”ME, SK - BN” problem | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ |
| DiagSVM | 1E+1 | 1E-4 | - | 86.17 | 92.79 | 53.47 | 1E+1 | 1E-4 | - | 76.33 | 63.6 | 84.76 |
| DiagSVM-pca | 1E+1 | 1E-4 | 50 | 86.5 | 87.43 | **66.67** | 1E+1 | 1E-4 | 1000 | 76.17 | 63.11 | **85.11** |
| DiagSVM-kpca | 1E+2 | 1E-1 | 1000 | **88.17** | 90.57 | 66.1 | 1E+2 | 1E-1 | 500 | 76.0 | **68.86** | 78.75 |
| DiagSVM-ica | 1E+3 | 1E-3 | 100 | 85.17 | **92.18** | 50.5 | 1E+3 | 1E-3 | 250 | 75.0 | 62.34 | 82.93 |
| Libsvm | 1E+1 | 1E-4 | - | 86.0 | 92.43 | 53.06 | 1E+1 | 1E-4 | - | 76.5 | 64.47 | 83.87 |
| Libsvm-pca | 1E+1 | 1E-4 | 50 | 86.5 | 87.43 | **66.67** | 1E+1 | 1E-4 | 500 | 76.67 | 64.63 | 84.1 |
| Libsvm-kpca | 1E+2 | 1E-1 | 1000 | 88.0 | 90.41 | 65.52 | 1E+2 | 1E-1 | 250 | 76.0 | 66.49 | 80.44 |
| Libsvm-ica | 1E+1 | 1E-3 | 100 | 85.0 | 85.0 | 00.00 | 1E+1 | 1E-3 | 250 | **77.33** | 65.24 | 85.01 |

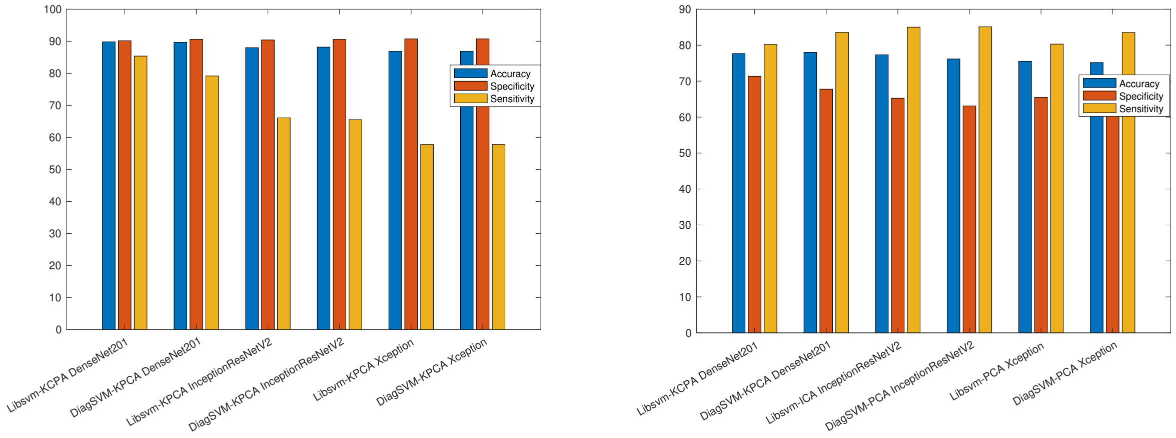
In Table 14, the algorithm DiagSVM with KPCA and ICA feature reduction presented better accuracy and specificity, respectively, for the ”ME, BN - SK” problem using InceptionResNetV2. For the ”ME, SK - BN” problem, the Libsvm-ICA algorithm performed better accuracy than all other algorithms. The results obtained from InceptionResNetV2 models were better than those obtained with Xception.

Table 15. ISIC2017 classification results using DenseNet201 as feature extractor.

| | ”ME, BN - SK” problem | | | | | | ”ME, SK - BN” problem | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ |
| DiagSVM | 1E+1 | 1E-3 | - | 88.67 | 92.34 | 64.1 | 1E+1 | 1E-3 | - | 75.33 | 62.24 | 84.12 |
| DiagSVM-pca | 1E+1 | 1E-3 | 1000 | 88.5 | 92.16 | 63.64 | 1E+1 | 1E-3 | 1000 | 75.33 | 61.94 | 84.7 |
| DiagSVM-kpca | 1E+2 | 1E-1 | 100 | 89.67 | 90.58 | 79.17 | 1E+2 | 1E-1 | 250 | **78.0** | 67.77 | 83.55 |
| DiagSVM-ica | 1E+1 | 1E-1 | 50 | 88.83 | 90.05 | 74.47 | 1E+1 | 1E-1 | 50 | 75.83 | 63.96 | 82.8 |
| Libsvm | 1E+1 | 1E-3 | - | 88.67 | 92.34 | 64.1 | 1E+1 | 1E-3 | - | 75.33 | 62.24 | 84.12 |
| Libsvm-pca | 1E+1 | 1E-3 | 500 | 88.5 | **92.65** | 62.65 | 1E+1 | 1E-3 | 1000 | 75.5 | 62.2 | **84.75** |
| Libsvm-kpca | 1E+2 | 1E-1 | 500 | **89.83** | 90.16 | **85.37** | 1E+2 | 1E-1 | 250 | 77.67 | **71.35** | 80.19 |
| Libsvm-ica | 1E+1 | 1E-1 | 50 | 89.0 | 89.93 | 77.27 | 1E+1 | 1E-1 | 50 | 75.83 | 63.96 | 82.8 |

Among all models, DenseNet201 presented the better results for the ISIC2017 dataset, as shown in Table 15. For the ”ME, BN - SK” problem, Libsvm-KPCA obtained a small advantage in terms of accuracy. However, the Libsvm-KPCA presented better accuracy for the ”ME, BN - SK” problem.

Based on the results presented in Table 13, 14 and 15, Figure 8 shows the metric results of the algorithms for each model. We chose the accuracy metric to represent the algorithm that better performed the model. In the ISCI2017 dataset, for the ”ME, BN - SK” problem, the DenseNet201 model with Libsvm-KPCA algorithm obtained the best results. For the ”ME, SK - BN” problem, the DenseNet201 model with DiagSVM-KPCA performed better than all other algorithms. In general, DiagSVM and Libsvm performed very similarly, indicating that DiagSVM is a good alternative as an SVM solver for such problems.



(a) Results for the ”ME, BN - SK” problem.

(b) Results for the ”ME, SK - BN” problem.

Figure 8. Representation of the accuracy, sensitivity and specificity to the best algorithms in each model applied ISIC2017 dataset

### 6.2. *PH2 Dataset*

The PH2 dataset contains a total of 200 images, where 80 are CN, 80 are AN, and 40 are ME. In Table 3, we described the class composition after splitting the data into training and test datasets. In Table 4, we present the training samples after augmentation, which we use to perform the experiments in this section.

From the PH2 dataset, we built two binary classification problems, ”CN, AN - ML” and ”AN, ML - CN”. In Table 16, 17 and 18, we present the experiment results of all algorithms combining each one with a CNN model.

Table 16. PH2 classification results using Xception as a feature extractor.

| | ”CN, AN - ML” problem | | | | | | ”AN, ML - CN” problem | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ |
| DiagSVM | 1E+1 | 1E-4 | - | 88.33 | 90.2 | 77.78 | 1E+1 | 1E-4 | - | 73.33 | 77.78 | 66.67 |
| DiagSVM-pca | 1E+2 | 1E-4 | 500 | 90.0 | 92.0 | 80.0 | 1E+2 | 1E-4 | 50 | 80.0 | 81.58 | **77.27** |
| DiagSVM-kpca | 1E+3 | 1E-2 | 1000 | 88.33 | 87.27 | **100.0** | 1E+3 | 1E-2 | 100 | 80.0 | 83.33 | 75.0 |
| DiagSVM-ica | 1E+1 | 1E-4 | 50 | **91.67** | **93.88** | 81.82 | 1E+1 | 1E-4 | 50 | 76.67 | **86.67** | 66.67 |
| Libsvm | 1E+2 | 1E-4 | 2048 | 90.0 | 90.38 | 87.5 | 1E+2 | 1E-4 | 2048 | 73.33 | 79.41 | 65.38 |
| Libsvm-pca | 1E+0 | 1E-4 | 100 | 90.0 | 92.0 | 94.44 | 1E+0 | 1E-4 | 50 | 80.0 | 81.58 | **77.27** |
| Libsvm-kpca | 1E+3 | 1E-2 | 1000 | 88.33 | 87.27 | **100.0** | 1E+3 | 1E-2 | 100 | 78.33 | 79.49 | 76.19 |
| Libsvm-ica | 1E+2 | 1E-3 | 50 | **91.67** | **93.88** | 94.44 | 1E+2 | 1E-3 | 100 | **81.67** | 85.71 | 76.0 |

In Table 16, using the Xception model, for the ”CN, AN - ML” problem, DiagSVM-ICA and Libsvm-ICA obtained the same accuracy and specificity, and DiagSVM-KPCA and Libsvm-KPCA the same sensitivity. Now, for the ”AN, ML - CN” problem, the Libsvm-ICA presented the best accuracy, while the DiagSVM-ICA presented the best specificity.

Table 17. PH2 classification results using InceptionResNetV2 as feature extractor.

| | ”CN, AN - ML” problem | | | | | | ”AN, ML - CN” problem | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ |
| DiagSVM | 1E+2 | 1E-4 | 1536 | 93.33 | 95.83 | 83.33 | 1E+2 | 1E-4 | 1536 | 83.33 | 84.21 | 81.82 |
| DiagSVM-pca | 1E+0 | 1E-4 | 50 | **96.67** | **100.0** | 85.71 | 1E+0 | 1E-4 | 50 | 83.33 | 82.5 | 85.0 |
| DiagSVM-kpca | 1E+2 | 1E-2 | 500 | 91.67 | 92.16 | 88.89 | 1E+2 | 1E-2 | 1000 | 83.33 | 82.5 | 85.0 |
| DiagSVM-ica | 1E+3 | 1E-4 | 50 | 95.0 | 95.92 | 90.91 | 1E+3 | 1E-4 | 100 | 83.33 | 82.5 | 85.0 |
| Libsvm | 1E+2 | 1E-4 | - | 93.33 | 95.83 | 83.33 | 1E+2 | 1E-4 | - | 81.67 | 80.49 | 84.21 |
| Libsvm-pca | 1E+1 | 1E-4 | 50 | 91.67 | 95.74 | 91.89 | 1E+1 | 1E-4 | 250 | **85.0** | **84.62** | **85.71** |
| Libsvm-kpca | 1E+2 | 1E-2 | 250 | 90.0 | 90.38 | **97.14** | 1E+2 | 1E-2 | 1000 | 83.33 | 82.5 | 85.0 |
| Libsvm-ica | 1E+1 | 1E-3 | 50 | 95.0 | 95.92 | **97.14** | 1E+1 | 1E-3 | 100 | 83.33 | 82.5 | 85.0 |

From Table 17, using the InceptionResNetV2 model, DiagSVM-PCA obtained the best accuracy and specificity results. For the ”AN, ML - CN” problem, the Libsvm-PCA presented the best accuracy, specificity and sensitivity. The results obtained after using InceptionResNetV2 as a feature extractor were better than those using Xception.

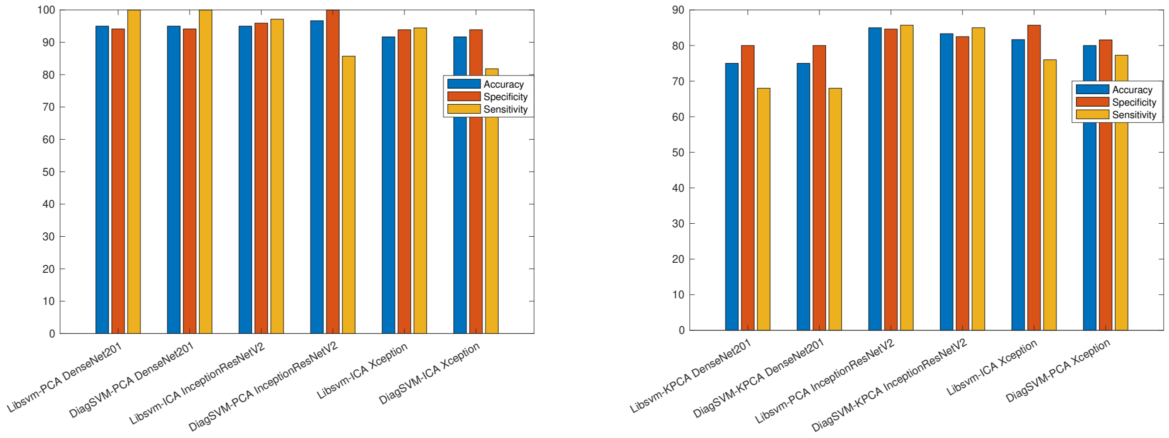
Table 18. PH2 classification results using DenseNet201 as feature extractor.

| | ”CN, AN - ML” problem | | | | | | ”AN, ML - CN” problem | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Algorithm | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ | C | $\sigma$ | Nc | $A_c$ | $S_p$ | $S_e$ |
| DiagSVM | 1E+3 | 1E-4 | - | 93.33 | 92.31 | 100.0 | 1E+3 | 1E-4 | - | 66.67 | 73.53 | 57.69 |
| DiagSVM-pca | 1E+3 | 1E-4 | 1000 | **95.0** | **94.12** | **100.0** | 1E+3 | 1E-4 | 250 | 68.33 | 79.31 | 58.06 |
| DiagSVM-kpca | 1E+2 | 1E-1 | 100 | 93.33 | 92.31 | **100.0** | 1E+2 | 1E-1 | 50 | **75.0** | 80.0 | **68.0** |
| DiagSVM-ica | 1E+1 | 1E-4 | 50 | **95.0** | **94.12** | **100.0** | 1E+1 | 1E-4 | 100 | 71.67 | 82.76 | 61.29 |
| Libsvm | 1E+0 | 1E-4 | 1920 | 95.0 | 94.12 | 100.0 | 1E+0 | 1E-4 | 1920 | 66.67 | 73.53 | 57.69 |
| Libsvm-pca | 1E+0 | 1E-4 | 50 | **95.0** | **94.12** | **100.0** | 1E+0 | 1E-4 | 50 | 70.0 | 78.12 | 60.71 |
| Libsvm-kpca | 1E+2 | 1E-1 | 100 | 93.33 | 92.31 | **100.0** | 1E+2 | 1E-1 | 50 | **75.0** | 80.0 | **68.0** |
| Libsvm-ica | 1E+1 | 1E-2 | 50 | **95.0** | **94.12** | **100.0** | 1E+1 | 1E-2 | 50 | 73.33 | **83.33** | 63.33 |

Table 18 shows that Libsvm and DiagSVM using PCA and ICA performed identically to the ”CN, AN - ML” problem using the DenseNet201 model. For the ”AN, ML - CN” problem, DiagSVM-KPCA and Libsvm-KPCA presented the same accuracy result.

Among all algorithms applied to the ”CN, AN - ML” problem, the DiagSVM-PCA was the one that performed better. And for the ”AN, ML - CN” problem”, Libsvm-PCA obtained the better results. Both of them using the InceptionResNetV2 model as a feature extractor.

Similarly to the chart presented in Section 6.1, we present in Figure 9 the metric results of the best algorithms for each problem and model. The Libsvm and DiagSVM results for the ”CN, AN - ML” problem were similar for the Xception and DenseNet201 models, but for the InceptionResNetV2 model, the DiagSVM obtained the best result from all other algorithms and models.



(a) Results for the ”CN, AN - ML” problem.

(b) Results for the ”AN, ML - CN” problem.

Figure 9. Representation of the accuracy, sensitivity and specificity to the best algorithms in each model applied to PH2 dataset

## 7. Conclusions

This work presented a new framework, named DiagSVM, for training nonlinear Support Vector Machines (SVM). The framework consists of a combination of two approaches. In the first stage, a separable quadratic approximation technique of the problem’s objective function is used, resulting in a new subproblem characterized as a separable quadratic knapsack problem (SQKP). According to our literature search, such an approach has not yet been used to solve a nonlinear optimization problem in the format that SVM training has. Numerical results indicate satisfactory results compared to the Libsvm solver for the tested datasets.

We tested our approach using benchmark datasets and the Skin Lesion Classification (SLC) problem. In Section 5, all DiagSVM combinations demonstrate robustness in solving different types and size problems. In Section 6, the DiagSVM solved all proposed SLC problems. Comparing the DiagSVM and Libsvm, Figure 8 and 9 reveals that DiagSVM obtained competitive performance for all of the formulated SLC problems.

Due to the imbalance class distributions of skin lesions, we used various augmentation approaches such as horizontal/vertical flip, width/height shift, zoom and rotation. Augmenting data enabled us to increase the size of the training set and reduce the overfitting problem. To improve the accuracy of the SLC problems, we used PCA, KPCA and ICA for feature reduction. We employed standard evaluation metrics such as accuracy, specificity, and sensitivity to evaluate the results.

Our experiments show that DenseNet201 outperforms the Xception and InceptionResNetV2 architectures for the ISIC2017 dataset, with classification accuracy, specificity, and sensitivity equal to 89.83%, 90.16% and 85.37%, respectively. For the PH2 dataset, the InceptionResNetV2 outperformed DenseNet201 and Xception, with accuracy, specificity, and sensitivity equal to 96.67% 100.00% and 85.71%, respectively. Regarding accuracy, specificity and sensitivity, the proposal presented satisfactory results compared with the well-established method in the literature: Libsvm.

For future research, we aim to expand the study of separable quadratic approximation methods to apply in the first stage of the framework, develop new root-finding methods, and improve the existing ones. For example, the simplicity of the Fixed-point [29] method can be explored and combined with other approaches in the literature to improve the performance of the DiagSVM. In the context of the SLC problem, feature reduction can play more roles, and other features, like the shape and color of the lesion, can be used before the feature reduction.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgement

In this research, the author Paulo Santos was supported by the CNPq Grant 309552/2022-2.

## REFERENCES

1. Boser, B.E., Guyon, I.M. and Vapnik, V.N., 1992, July. A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (pp. 144-152).
2. Cortes, C. and Vapnik, V., 1995. Support-vector networks. Machine learning, 20, pp.273-297.
3. Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L. and Lopez, A., 2020. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, pp.189-215.
4. Bhavsar, H. and Panchal, M.H., 2012. A review on support vector machine for data classification. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 1(10), pp.185-189.
5. Joachims, T., 1999. Svmlight: Support vector machine. SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund, 19(4), p.25.
6. Collobert, R. and Bengio, S., 2001. SVMTorch: Support vector machines for large-scale regression problems. Journal of machine learning research, 1(Feb), pp.143-160.
7. Shalev-Shwartz, S., Singer, Y. and Srebro, N., 2007, June. Pegasos: Primal estimated sub-gradient solver for svm. In Proceedings of the 24th international conference on Machine learning (pp. 807-814).
8. Chang, C.C. and Lin, C.J., 2011. LIBSVM: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST), 2(3), pp.1-27.
9. Münnich, R.T., Sachs, E.W. and Wagner, M., 2012. Numerical solution of optimal allocation problems in stratified sampling under box constraints. AStA Advances in Statistical Analysis, 96, pp.435-450.
10. Khemchandani, R. and Chandra, S., 2007. Twin support vector machines for pattern classification. IEEE Transactions on pattern analysis and machine intelligence, 29(5), pp.905-910.
11. Suykens, J.A. and Vandewalle, J., 1999. Least squares support vector machine classifiers. Neural processing letters, 9, pp.293-300.
12. Haftka, R.T. and Gürdal, Z., 2012. Elements of structural optimization (Vol. 11). Springer Science & Business Media.
13. Bottou, L. and Lin, C.J., 2007. Support vector machine solvers. Large scale kernel machines, 3(1), pp.301-320.
14. R. Cominetti, W. F. Mascarenhas, P. J. Silva, A newton's method for the continuous quadratic knapsack problem, Mathematical Programming Computation 6 (2) (2014) 151–169.
15. K. Kiwiel, Variable fixing algorithms for the continuous quadratic knapsack problem, Journal of Optimization Theory and Applications 136 (3) (2008) 445–458.
16. J. Snyman, A. Hay, The spherical quadratic steepest descent (sqsd) method for unconstrained minimization with no explicit line searches, Computers & Mathematics with Applications 42 (1-2) (2001) 169–178.
17. C.-W. Hsu, C.-C. Chang, C.-J. Lin, et al., A practical guide to support vector classification (2003).
18. M. Zhu, J. L. Nazareth, H. Wolkowicz, The quasi-cauchy relation and diagonal updating, SIAM Journal on Optimization 9 (4) (1997) 1192– 1204.
19. J. E. Dennis, Jr, H. Wolkowicz, Sizing and least-change secant methods, SIAM Journal on Numerical Analysis 30 (5) (1993) 1291–1314.
20. J. Shawe-Taylor, S. Sun, A review of optimization methodologies in support vector machines, Neurocomputing 74 (17) (2011) 3609–3618.
21. Y. LeCun, C. Cortes, C. Burges, Mnist handwritten digit database (2010).

22. A. Baldominos, Y. Saez, P. Isasi, A survey of handwritten character recognition with mnist and emnist, Applied Sciences 9 (15) (2019) 3169.
23. N. Cristianini, J. Shawe-Taylor, et al., An introduction to support vector machines and other kernel-based learning methods, Cambridge university press, 2000.
24. Y.-J. Lee, O. L. Mangasarian, Rsvm: Reduced support vector machines, in: Proceedings of the 2001 SIAM International Conference on Data Mining, SIAM, 2001, pp. 1–17.
25. G. Fung, O. L. Mangasarian, Incremental support vector machine classification, in: Proceedings of the 2002 SIAM International Conference on Data Mining, SIAM, 2002, pp. 247–260.
26. B. Sch olkopf, A. J. Smola, R. C. Williamson, P. L. Bartlett, New support vector algorithms, Neural computation 12 (5) (2000) 1207–1245.
27. C.-F. Lin, S.-D. Wang, Fuzzy support vector machines, IEEE transactions on neural networks 13 (2) (2002) 464–471.
28. E. G. Birgin, J. M. Martınez, M. Raydan, Spectral projected gradient methods: review and perspectives, Journal of Statistical Software 60 (2014) 1–21.
29. A. Alves, J. Silva, P. Santos, L. Matioli, S. Souza, An accelerated fixed-point algorithm applied to quadratic convex separable knapsack problems, Journal of Control Science and Engineering, pp. 1687-5249 2023.
30. T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. Marcal, J. Rozeira, Ph 2-a dermoscopic image database for research and benchmarking, in: 2013 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC), IEEE, 2013, pp. 5437–5440.
31. M. K. Hasan, M. A. Ahamad, C. H. Yap, G. Yang, A survey, review, and future trends of skin lesion segmentation and classification, Computers in Biology and Medicine (2023) 106624.
32. R. Marks, An overview of skin cancers, Cancer 75 (S2) (1995) 607–612.
33. M. Patriksson, A survey on the continuous nonlinear resource allocation problem, European Journal of Operational Research 185 (1) (2008) 1–46.
34. M. Patriksson, C. Stromberg, Algorithms for the continuous nonlinear resource allocation problem—new implementations and numerical studies, European Journal of Operational Research 243 (3) (2015) 703–722.
35. M. A. Khan, T. Akram, M. Sharif, T. Saba, K. Javed, I. U. Lali, U. J. Tanik, A. Rehman, Construction of saliency map and hybrid set of features for efficient segmentation and classification of skin lesion, Microscopy research and technique 82 (6) (2019) 741–763.
36. Cao, L.J. and Chong, W.K., 2002, November. Feature extraction in support vector machine: a comparison of PCA, XPCA and ICA. In Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02. (Vol. 2, pp. 1001-1005). IEEE.
37. Schölkopf, B., Smola, A. and Müller, K.R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural computation, 10(5), pp.1299-1319.
38. Hyvarinen, A., 1999. Fast and robust fixed-point algorithms for independent component analysis. IEEE transactions on Neural Networks, 10(3), pp.626-634.
39. A. Saez, B. Acha, C. Serrano, Pattern analysis in dermoscopic images, Computer vision techniques for the diagnosis of skin Cancer (2014) 23–48.
40. J. Mayer, Systematic review of the diagnostic accuracy of dermatoscopy in detecting malignant melanoma., The Medical Journal of Australia 167 (4) (1997) 206–210.
41. Z. J. Wolner, O. Yelamos, K. Liopyris, T. Rogers, M. A. Marchetti, A. A. Marghoob, Enhancing skin cancer diagnosis with dermoscopy, Dermatologic clinics 35 (4) (2017) 417–437.
42. Y. Filali, S. Abdelouahed, A. Aarab, An improved segmentation approach for skin lesion classification, Statistics, Optimization & Information Computing 7 (2) (2019) 456–467.
43. S. W. Menzies, C. Ingvar, K. A. Crotty, W. H. McCarthy, Frequency and morphologic characteristics of invasive melanomas lacking specific surface microscopic features, Archives of dermatology 132 (10) (1996) 1178–1182.
44. R. J. Friedman, D. S. Rigel, A. W. Kopf, Early detection of malignant melanoma: the role of physician examination and self-examination of the skin., CA: a cancer journal for clinicians 35 (3) (1985) 130–151.
45. G. Argenziano, G. Fabbrocini, P. Carli, V. De Giorgi, E. Sammarco, M. Delfino, Epiluminescence microscopy for the diagnosis of doubtful melanocytic skin lesions: comparison of the abcd rule of dermatoscopy and a new 7-point checklist based on pattern analysis, Archives of dermatology 134 (12) (1998) 1563–1570.
46. N. C. Lynn, Z. M. Kyu, Segmentation and classification of skin cancer melanoma from skin lesion images, in: 2017 18th international conference on parallel and distributed computing, applications and technologies (PDCAT), IEEE, 2017, pp. 117–122.
47. S. Jain, N. Pise, et al., Computer aided melanoma skin cancer detection using image processing, Procedia Computer Science 48 (2015) 735–740.
48. Y. Filali, H. EL Khoukhi, M. A. Sabri, A. Aarab, Efficient fusion of handcrafted and pre-trained cnns features to classify melanoma skin cancer, Multimedia Tools and Applications 79 (41-42) (2020) 31219–31238.
49. R. Seeja, A. Suresh, Deep learning based skin lesion segmentation and classification of melanoma using support vector machine (svm), Asian Pacific journal of cancer prevention: APJCP 20 (5) (2019) 1555.
50. S. Kumar, C. J, Skin cancer classification using machine learning for dermoscopy image, International Journal of Innovative Technology and Exploring Engineering (IJITEE) (2019).
51. A. Mahbod, G. Schaefer, C. Wang, R. Ecker, I. Ellinge, Skin lesion classification using hybrid deep neural networks, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 1229–1233.
52. N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, et al., Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic), in: 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018), IEEE, 2018, pp. 168–172.
53. S. M. Alizadeh, A. Mahloojifar, Automatic skin cancer detection in dermoscopy images by combining convolutional neural networks and texture features, International Journal of Imaging Systems and Technology 31 (2) (2021) 695–707.

54. J. Schmidhuber, Deep learning in neural networks: An overview, Neural networks 61 (2015) 85–117.
55. Z. Yu, D. Ni, S. Chen, J. Qin, S. Li, T. Wang, B. Lei, Hybrid dermoscopy image classification framework based on deep convolutional neural network and fisher vector, in: 2017 IEEE 14th international symposium on biomedical imaging (ISBI 2017), IEEE, 2017, pp. 301–304.
56. M. A. Khan, M. Y. Javed, M. Sharif, T. Saba, A. Rehman, Multi-model deep neural network based features extraction and optimal selection approach for skin lesion classification, in: 2019 international conference on computer and information sciences (ICCIS), IEEE, 2019, pp. 1–7.
57. S. M. Alizadeh, A. Mahloojifar, Automatic skin cancer detection in dermoscopy images by combining convolutional neural networks and texture features, International Journal of Imaging Systems and Technology 31 (2) (2021) 695–707.
58. B. Shetty, R. Fernandes, A. P. Rodrigues, R. Chengoden, S. Bhattacharya, K. Lakshmanna, Skin lesion classification of dermoscopic images using machine learning and convolutional neural network, Scientific Reports 12 (1) (2022) 18134.
59. Cassidy, B., Kendrick, C., Brodzicki, A., Jaworek-Korjakowska, J. and Yap, M.H., 2022. Analysis of the ISIC image datasets: Usage, benchmarks and recommendations. Medical image analysis, 75, p.102305.
60. Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).
61. Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A., 2017, February. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).
62. Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q., 2017. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
63. Kassani, S.H. and Kassani, P.H., 2019. A comparative study of deep learning architectures on melanoma detection. Tissue and Cell, 58, pp.76-83.
64. Ye, Y. and Tse, E., 1989. An extension of Karmarkar's projective algorithm for convex quadratic programming. Mathematical programming, 44, pp.157-179.
65. Kang, Z., Zhou, W., Zhao, Z., Shao, J., Han, M. and Xu, Z., 2020, April. Large-scale multi-view subspace clustering in linear time. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 04, pp. 4412-4419).
66. Abdiansah, A. and Wardoyo, R., 2015. Time complexity analysis of support vector machines (SVM) in LibSVM. Int. J. Comput. Appl, 128(3), pp.28-34.
67. Yokota, R., Barba, L.A. and Knepley, M.G., 2010. PetRBF—A parallel O (N) algorithm for radial basis function interpolation with Gaussians. Computer Methods in Applied Mechanics and Engineering, 199(25-28), pp.1793-1804.
68. Qiu, S. and Lane, T., 2005, April. Parallel computation of RBF kernels for support vector classifiers. In Proceedings of the 2005 SIAM International Conference on Data Mining (pp. 334-345). Society for Industrial and Applied Mathematics.