axioms

MDPI

Article

PSEV-BF Methodology for Object Recognition of Birds in Uncontrolled Environments

Lucía J. Hernández-González [1,†], Juan Frausto-Solís [1,*,†], Juan J. González-Barbosa [1,†], Juan Paulo Sánchez-Hernández [2], Deny Lizbeth Hernández-Rabadán [2] and Edgar Román-Rangel [3]

[1] Graduate & Research Division, Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Madero City 89440, Mexico
[2] Information of Technology Division, UPEMOR, Jiutepec Cty 62574, Mexico
[3] Department of Computer Science, Instituto Tecnológico Autónomo de México, México City 01080, Mexico
* Correspondence: juan.frausto@gmail.com
† These authors contributed equally to this work.

Abstract: Computer vision methodologies using machine learning techniques usually consist of the following phases: pre-processing, segmentation, feature extraction, selection of relevant variables, classification, and evaluation. In this work, a methodology for object recognition is proposed. The methodology is called PSEV-BF (pre-segmentation and enhanced variables for bird features). PSEV-BF includes two new phases compared to the traditional computer vision methodologies, namely: pre-segmentation and enhancement of variables. Pre-segmentation is performed using the third version of YOLO (you only look once), a convolutional neural network (CNN) architecture designed for object detection. Additionally, a simulated annealing (SA) algorithm is proposed for the selection and enhancement of relevant variables. To test PSEV-BF, the repository commons object in Context (COCO) was used with images exhibiting uncontrolled environments. Finally, the APIoU metric (average precision intersection over union) is used as an evaluation benchmark to compare our methodology with standard configurations. The results show that PSEV-BF has the highest performance in all tests.

Keywords: pre-segmentation; simulated annealing; YOLOV3; COCO; semantic segmentation

Citation: Hernández-González, L.J.; Frausto-Solís, J.; González-Barbosa, J.J.; Sánchez-Hernández, J.P.; Hernández-Rabadán, D.L.; Román-Rangel, E. PSEV-BF Methodology for Object Recognition of Birds in Uncontrolled Environments. *Axioms* **2023**, *12*, 197. https://doi.org/10.3390/axioms12020197

Academic Editor: Cesar Rego

Received: 28 December 2022
Revised: 30 January 2023
Accepted: 6 February 2023
Published: 13 February 2023

1. Introduction

Environmental scientists often use birds to understand ecosystems because birds are sensitive to environmental changes [1]. As a result, there are many protected areas around the world dedicated to the conservation of bird species. However, identifying and classifying birds using conventional artificial vision is a difficult task. This is a particularly complicated problem for images where occlusion is present in uncontrolled environments.

Computer vision is a field of artificial intelligence that attempts to extract meaningful information by analyzing and processing image patterns. Additionally, this field, has several branches: classification, object localization, object detection, object recognition, and segmentation.

Object recognition is a task that identifies an object present in images or videos. It is one of the most important applications of machine learning and deep learning. The purpose of this field is to recognize the content of an image using machine learning techniques or deep learning architecture.

Figure 1a shows the classical computational vision methodology for object recognition in computer vision. The other alternative is to use a deep learning architecture (Figure 1b). As we can see, the latter is less interpretable because it is equivalent to a black box where the main processes of feature extraction and selection are hidden. This paper proposes a methodology that combines elements of the two methodologies.

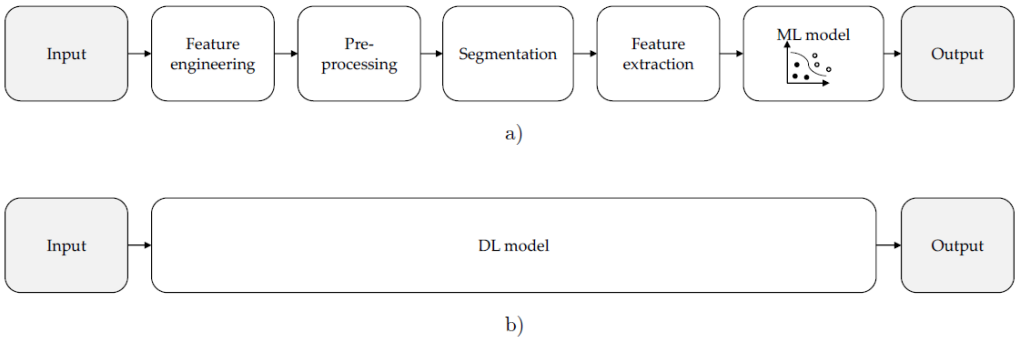


Figure 1. (**a**) Traditional computer vision methodology with ML. (**b**) Deep learning computer vision methodology.

Object recognition is considered one of the critical problems because there are several challenges to deal with images, such as:

A. Occlusion, i.e., obscuring part of the object by equal or unequal elements of the scene, Figure 2a.
B. Environmental artifacts, such as rain and fog, which can affect the quality of the image, Figure 2b.
C. Uncontrolled environments are caused by the lack of a protocol for image captures, such as the object's contrasting background, height, distance from the camera, and light correction, Figure 2c.

Figure 2. Challenge examples: (**a**) the bird appears occluded. (**b**) The appearance of haze in the image softens the color of the background. (**c**) The image appears with blur to the camera angle [2].

Therefore, in order to recognize an object, the methodology considers tasks, such as object detection and segmentation. Segmentation is the principal problem and is our focus in this work.

The segmentation goal is to identify the pixels belonging to the target object or region of interest (ROI). However, determining the optimal number of regions per image is very time-consuming and computationally expensive. Segmentation methods based on pixel-by-pixel classification can be broadly divided into two families: semantic segmentation and instance segmentation. The first type separates all pixels that belong to the same object class. The second identifies each of the objects present in the image as an individual.

Traditionally, variable or feature selection is performed using composite variables, such as the principal component analysis technique (PCA) [3–5] and other classification methods [6]. Composite variables are methods that simplify the sample space of variables by normalizing linear combinations of them. However, in recent years, there have been published methods for improving feature selection by incorporating combinatorial optimization methods [7–12] and model selections for machine learning [13]. For this reason, in this paper, we propose including an enhanced method for feature selection using the simulated annealing (SA) algorithm, a metaheuristic for combinatorial optimization, which is used to improve the feature set selected with the PCA technique.

We present a new methodology for object recognition called PSEV-BF (pre-segmentation and enhanced variables for bird features) that uses the pre-segmentation information before segmentation to refine the delimited area. This methodology has the phases of pre-processing, pre-segmentation, segmentation, ROI feature extraction, enhancement of relevant variables, and classification.

The rest of the paper is organized as follows. Section 2 presents related work with a qualitative comparison of object recognition. Section 3 presents the formulation and description of all phases of the proposed methodology. Section 4 defines the data, performance metrics, and tools used in this work. In Section 5, we present the proposed algorithms and their tuning method and show the application of the methodology to the dataset presented in the paper. Finally, we compare the results with the classical methodology. Section 6 presents our conclusions.

2. Related Works

In this section, several works related to the problems of computer vision phases are discussed. For instance, in the work of [14], a feature selection algorithm based on genetic programming (GP) is proposed. In [14], the segmentation and classification of horses and airplane images were implemented using parsimony GP features selection (PGP-FS), nondominated sorting GP feature selection (NSGP-FS), and strength Pareto GP feature selection (SPGT-FS) algorithms. These features were subjected to the decision tree, naive Bayes, and multilayer perceptron classifiers from the Weka tool. A total of 52 features were extracted in terms of Gabor filter, color, and statistical values based on a grayscale. The accuracy, F1, precision, and recall metrics were used. The selection method shows that, on average, 15 features are selected from the original 52.

There are works related to the segmentation and classification of images of skin lesions. For instance, in [15], the authors used the PCA technique and the Boltzmann entropy method to select a set of features. Feature selection was performed by considering the score (variance explained) of each PCA component. The features considered were color, texture, and shape, resulting in a total of 3849 features. By using PCA and Boltzmann methods, the number of features was reduced to 449. The selection of features was validated using the metrics DICE, Jaccard index, Jaccard distance, and Seg diameter. The selected features were classified using the following machine learning models: support vector machine (SVM), decision trees (DT), bagged trees (BT), subspace discriminant analysis (SDA), weighted-K nearest neighbor (W-KNN), fine-K nearest neighbor (F-KNN), subspace-K nearest neighbor (S-KNN), linear discriminant analysis (LDA), quadric discriminant analysis (QDA), cubic-support vector machine (C-SVM), and quadric-support vector machine (Q-SVM). The classifiers were validated using the metrics of sensitivity, specificity, accuracy, and F-score.

In 2018, Sharif and collaborators proposed a methodology for citrus disease detection using optimized weighted segmentation and feature selection [16]. The processing phase consists of a top-hat filter to eliminate noise elements and a Gaussian filter to soften the image and eliminate high-intensity fluctuations. In the segmentation phase, they used a combination of segmentation techniques with weight assignment and relevance map, which allow for retaining the elements of the image with high contrast. The extracted features are related to color, texture, and geometric features, giving a total of 270 features. PCA is used to obtain a score corresponding to the explained variance of the components. Entropy and skewness are calculated for each component to select a vector of 100 features with the highest percentages. These features were obtained by training K-nearest weighted (KNN), ensemble boosted trees (EBT), DT, and LDA classifiers and then evaluating them by 10-fold. Validation of the methodology was performed using the metrics positive false rate, negative false rate, positive true rate, negative false rate, positive predictive value, false detection rate, area under the curve, and accuracy. The authors showed that their results can keep up with the current state-of-the-art methods.

Rehman et al., in 2018, applied a feature selection for image segmentation to detect glaucoma in the optic disk region using several parameters [17]. Additionally, in pre-

processing, a bilateral filter was applied to allow the removal of noise, a clipping that allows the activation of a threshold criterion to keep objects with high intensity and discard unwanted background noise, and finally, the normalization of the red (R) channel of the image to obtain information about the exciters searched. Statistical, text on the map, and fractal features were used in the segmentation phase. Then, a selection process was performed according to the method of minimal redundancy $\left(M_{I\ (A,\ B)}\right)$. These features were trained using SVM, random forest (RF), AdaBoostM1, and rus boost classifiers. The model was validated using the metrics of sensitivity, specificity, similarity coefficient DICE, precision, and area overlap based on the confusion matrix, with results competing with other state-of-the-art methods.

More recently, deep-learning-based methods were used for bird detection [18–22], classifying bird images [23], and recognizing birds [24]. For instance, [21] used the convolutional neural network (CNN) and you only look once V3 (YOLOV3) for the detection of birds from images. In this work, the authors propose a CNN with similar architecture to the Darknet-53 network. The model was validated using the accuracy metric and comparing it with similar architectures, such as region-based convolutional neural network (R-CNN), VGG-16 + SVM, and YOLO. Additionally, Q. Ou et al., in a previous work in 2020, used you only look once (YOLO) architecture to identify birds. Other works propose hybrid methods to improve bird detection and identification [25]. Kumar and Das, in 2018, proposed a R-CNN [26], which was used for obtaining binary masks of the ROI, and it was trained with instances from the Commons Object in Context (COCO) database.

Table 1 shows a summary of the most important aspects of the related works compared with our proposal. The first column has the name of the method used and its reference. The second column determines whether birds are the object of interest. The third and fourth columns indicate whether the images used are occluded and if they are in uncontrolled environments. The fifth through seventh columns indicate whether the methodology of the work was subjected to pre-processing, pre-segmentation, or segmentation. The eighth column indicates the number of selected features. The ninth column, named "Enhanced features", indicates whether specific methods for improving the variable selected by PCA were used or not. Finally, the last column indicates whether classification techniques were used. Table 1 shows the topics considered in different methodologies and related to this work. We observe that pre-segmentation and enhanced features are not commonly used.

Table 1. A comparison between the principal topics of the methodology of the related works.

Method	Birds	Occlusion	UE	Pre-P	Pre-S	Seg.	Feature Selection	Enhanced Feature	Classifiers ML
Genetic Programing [14]	✗	✗	✓	✗	✗	✓	✓	✓	✓
Classical- PCA/SVM [15]	✗	✗	✗	✓	✗	✓	✓	✗	✓
Classical-PCA/SVM [16]	✗	✗	✗	✓	✗	✓	✓	✓	✓
Classical- Minimum Redundancy/SVM [17]	✗	✓	✗	✓	✗	✓	✓	✗	✓
Deep CNN-53 [21]	✓	✓	✓	✓	✗	✗	✗	✗	✓
Deep CNN-19 [25]	✓	✓	✓	✗	✗	✗	✗	✗	✓
CNN-Transfer Learning [26]	✓	✓	✓	✗	✗	✓	✓	✗	✓
CNN-16 [1]	✓	✗	✓	✓	✗	✗	✓	✗	✓
PSEV-BF (proposal)	✓	✓	✓	✓	✓	✓	✓	✓	✓

UE: uncontrolled environment, Pre-P: pre-processing, Pre-S: pre-segmentation, Seg: segmentation.

3. Proposed Methodology

The proposed PSEV-BF methodology (Figure 3) consists of seven phases for training and six for testing: pre-processing, pre-segmentation, segmentation, ROI feature extraction, optimal variable selection, classification, and evaluation. In this section, all phases of this work are described in detail. 4

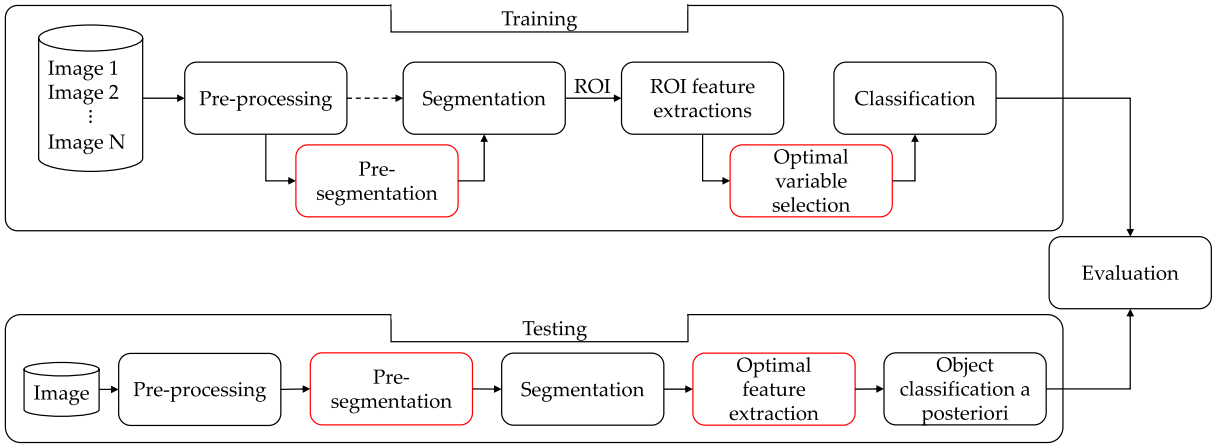


Figure 3. Proposed PSEV-BF methodology with pre-processing, pre-segmentation, segmentation, ROI feature extraction, optimal variable selection, classification, and evaluation.

3.1. Pre-Processing

The pre-processing of images is used to enhance their visual quality where several problems could be eliminated, such as brightness effects, illumination problems, and blurring due to poor contrast [16,27]. An image with low contrast affects the accuracy of segmentation and, hence, the rest of the phases. In this paper, a contrast enhancement technique based on the Gaussian smoothing function and histogram equalization is applied. First, the image contrast is increased by adding a histogram equalization filter. Then, the Gaussian smoothing filter is applied. The enhancement procedure is described in the following steps:

Step 1. Histogram equalization of an image is a transformation that aims to obtain a uniform distribution for each intensity level of an image. Said simply, it adjusts the image intensities to enhance contrast, as well as Equations (1) and (2). An image histogram is formed by tabulating the number of times that each intensity occurs throughout the image [28].

$$p_r(r_k) = \frac{n_k}{MN}; \tag{1}$$

$$T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) \mid k = 0,\ 1,\ 2,\ \ldots,\ L-1, \tag{2}$$

where p_r is the probability density function of f; n_k denotes the number of pixels that have intensity k; MN is the total number of pixels in the image; and L is the number of pixel intensity levels in the image. The application of this operation transforms the histogram into a histogram with a perfectly uniform shape across all gray levels. During the transformation, all pixels of one gray level are converted to another gray level, and the histogram is distributed over the entire available area, separating the occupations of the individual levels as much as possible.

Step 2. Applying the Gaussian smoothing function. Let $I(x, y, z)$ be the original image in a RGB, and $G(x, y)$ is a Gaussian function defined as:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3}$$

where x is the distance from the origin of the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

3.2. Pre-Segmentation

Pre-segmentation is a stage where different techniques can be applied to approximate the coordinates where the object of interest is roughly located within an image. There are

several works in the literature that use bounding boxes to determine the position of objects of interest, with YOLO being one of the most used methods. YOLO [29] is a convolutional neural network for object localization, very fast for real-time applications, and has several versions. YOLOV3 architecture [30] is composed of two main principal processes: a feature extractor called Darknet-53 and a convolutional method of the detection itself. Figure 4 shows a block diagram for YOLOV3.

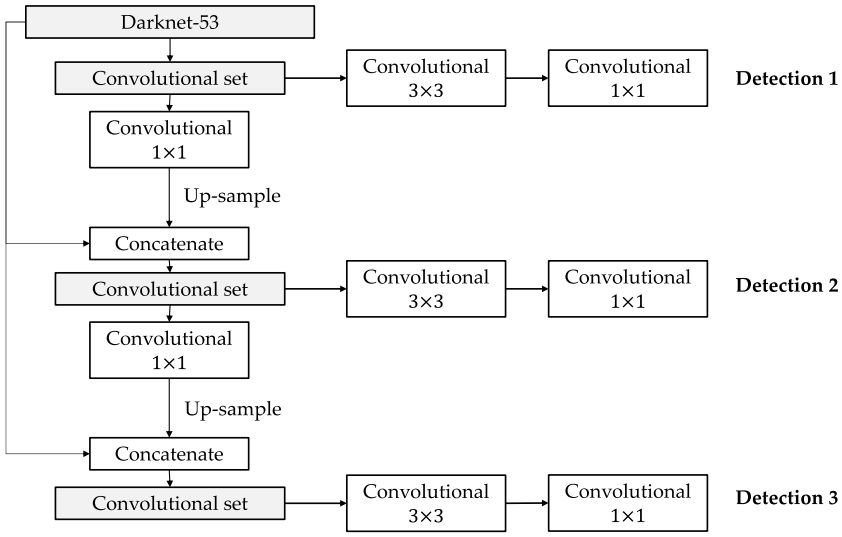


Figure 4. Architecture YOLOv3.

Figure 5 presents Darknet-53, which is a CNN with 53 layers of depth organized in five blocks of convolution layers, where each layer is a feature extractor. The last block of convolution layers contains the most important information obtained from this CNN, which is used to extract three detections of different scales.

	Type	Filters	Size	Output	
	Convolutional	32	3 × 3	256 × 256	
	Convolutional	64	3 × 3 / 2	128 × 128	
1×	Convolutional	32	1 × 1		
	Convolutional	64	3 × 3		
	Residual			128 × 128	
	Convolutional	128	3 × 3 / 2	64 × 64	
2×	Convolutional	64	1 × 1		
	Convolutional	128	3 × 3		
	Residual			64 × 64	
	Convolutional	256	3 × 3 / 2	32 × 32	
8×	Convolutional	128	1 × 1		
	Convolutional	256	3 × 3		
	Residual			32 × 32	Detection 1
	Convolutional	512	3 × 3 / 2	16 × 16	
8×	Convolutional	256	1 × 1		
	Convolutional	512	3 × 3		
	Residual			16 × 16	Detection 2
	Convolutional	1024	3 × 3 / 2	8 × 8	
4×	Convolutional	512	1 × 1		
	Convolutional	1024	3 × 3		
	Residual			8 × 8	Detection 3
	Avgpool		Global		
	Connected		1000		
	Softmax				

Figure 5. Architecture of Darknet53.

A convolutional set is a process to change the dimensionality of the outputs of Darknet-53, which come from the last three blocks. Figure 6 shows a convolutional set flow, which consists of a sequence of two convolutional filters: 1×1 and 3×3. A 1×1 convolutional filter allows one to obtain a feature map with a single dimension (WidthxHightx1). Usually, this filter is applied before an expansion filter: 3×3 convolution or 5×5 convolution.

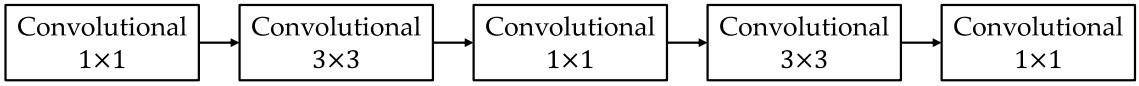


Figure 6. Convolutional set.

YOLOV3 [31] predicts a target value for each bounding box using logistic regression. The bounding box prediction consists of five components, as we see in Equation (4).

$$y = (x_1,\ y_1,\ x_2,\ y_2,\ confidence) \tag{4}$$

where $(x_1,\ y_1,\ x_2,\ y_2)$ coordinates represent the center of the box concerning the location of the grid cell. These coordinates are normalized between 0 and 1. The confidence value indicates how likely it is that the box contains an object and how accurate the bounding box is. The pre-segmentation phase is a critical stage for obtaining accurate segmentation results. For comparison purposes, it should be included in the results stage.

3.3. Segmentation

The segmentation phase aims to refine or adjust the region bounded by the coordinates from pre-segmentation to select a region of bird and no-bird. Figure 7a shows an example of the segmentation phase proposed to delineate regions for birds and non-birds based on the coordinates obtained by pre-segmentation. The adjustment of the pre-segmentation coordinates is defined by two configurations:

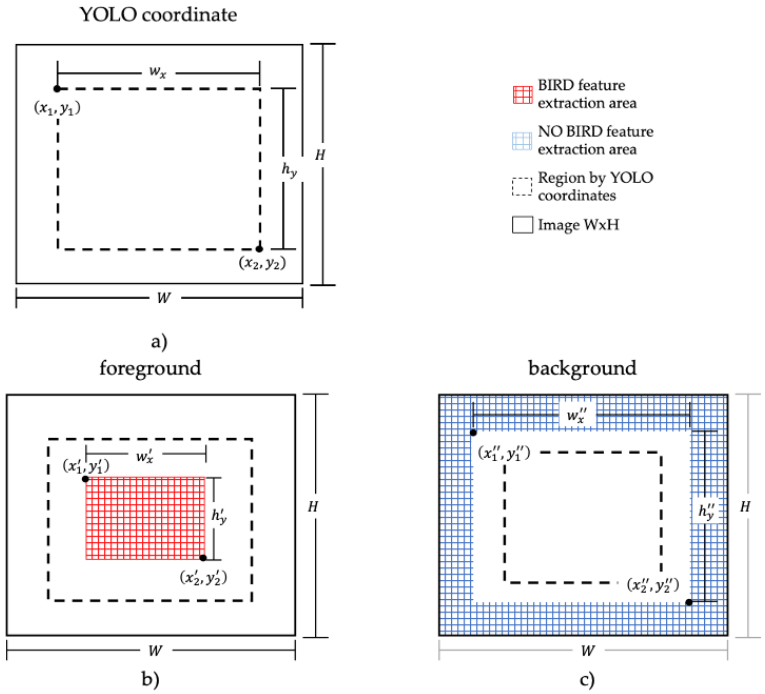


Figure 7. The region selected as bird and non-bird: (**a**) YOLOV3 coordinates; (**b**) the provisional region as a bird; and (**c**) the provisional region as non-bird.

Configuration 1: the pre-segmentation coordinates are reduced by 50%. Pixels within the range of Configuration 1 are classified as birds (Figure 7b). The coordinates of Configuration 1 are described below:

Given a coordinate vector, Equation (4), with values $[x_1,\ y_1,\ x_2,\ y_2]$, the width of the region, $w_x = x_2 - x_1$, and the height of the region, $h_y = y_2 - y_1$, are determined, and the region of the bird is defined in Equations (5)–(7):

$$x_1^{'} = x_1 + \frac{w_x}{4}\ ; x_2^{'} = x_2 - \frac{w_x}{4} \tag{5}$$

$$y_1^{'} = y_1 + \frac{h_y}{4} \; ; \; y_2^{'} = y_2 - \frac{y_x}{4}, \tag{6}$$

$$w_x^{'} = \frac{w_x}{2} \; ; \; h_y^{'} = \frac{h_y}{2} \tag{7}$$

where $(x_1,\ x_2)$ and $(y_1,\ y_2)$ are the coordinates from the origin $(0, 0)$ in the horizontal and vertical axis, respectively; $(x'_1,\ x'_2)$ and $(y'_1,\ y'_2)$ are the new coordinates in the horizontal and vertical axis.

Configuration 2: pre-segmentation coordinates are increased by 20%. Pixels outside Configuration 2 are classified as non-birds, Figure 7c. The coordinates of Configuration 1 are described below.

Given a coordinate vector, Equation (4), with values $[x_1,\ x_2,\ y_1,\ y_2]$, the width of the region $w_x = x_2 - x_1$, and the height of the region $h_y = y_2 - y_1$, are determined, and the region of the non-bird is defined in Equations (8)–(10) with the coordinates $(x''_1,\ x''_2)$ and $(y''_1,\ y''_2)$:

$$x_1^{''} = x_1 + \frac{w_x}{4} \; ; x_2^{''} = x_2 - \frac{w_x}{4} \tag{8}$$

$$y_1^{''} = y_1 + \frac{h_y}{4} \; ; \; y_2^{''} = y_2 - \frac{y_x}{4}, \tag{9}$$

$$w_x^{''} = w_x + \frac{w_x}{2} \; ; \; h_y^{''} = h_y + \frac{h_y}{2} \tag{10}$$

where $(x_1,\ x_2)$, $(y_1,\ y_2)$, $(x'_1,\ x'_2)$ and $(y'_1,\ y'_2)$ were previously defined for Equations (5)–(7).

The regions defined in Equations (5)–(10) are the pixels that are activated for feature extraction. The pixels between the bird and non-bird regions are not considered in the feature extraction phase. The label of a feature vector is assigned according to the region in which it is located.

3.4. Feature Extraction

Color features are extracted from 15 × 15 pixel regions, called super pixels, which represent a set of smoothed and enhanced images. The color features refer to the statistical behavior of the regions in each channel of the color models. The color models were selected according to the current state-of-the-art methods, and they are are HSI, CMYK, LAB, and XYZ. The variance and standard deviation are the features extracted for each channel.

Haralick texture features [32] are common texture descriptors in image analysis based on the concept that texture and hue are related. The features are determined using a correlation matrix of the intensity levels of an image, the gray-level co-occurrence matrix (GLCM). The number of gray levels in the image determines the size of the GLCM. Figure 8 shows an example of how the GLCM is determined [33].

GLCM starts with the transformation of an original image in RGB to a grayscale image, represented in Figure 8a,b. In the second step, an occurrence matrix $M(i, j)$ is created, i.e., the values at the positions in (i, j) represent the number of times the gray level intensity value i is a neighbor of the gray level intensity value j, as we showed in Figure 8c. After obtaining the occurrence matrix, $M(i, j)$, the values (i, j) are normalized, as shown in Figure 8d. Finally, the resulting matrix $p(i, j)$ is suitable for the application of the Haralick texture features (Figure 8e). Table 2 shows the notation of the variables involved in the calculation of the Haralick texture features. The first column represents the variable number; the second column shows the notation of the variables; and the third column describes the meaning of the notation.

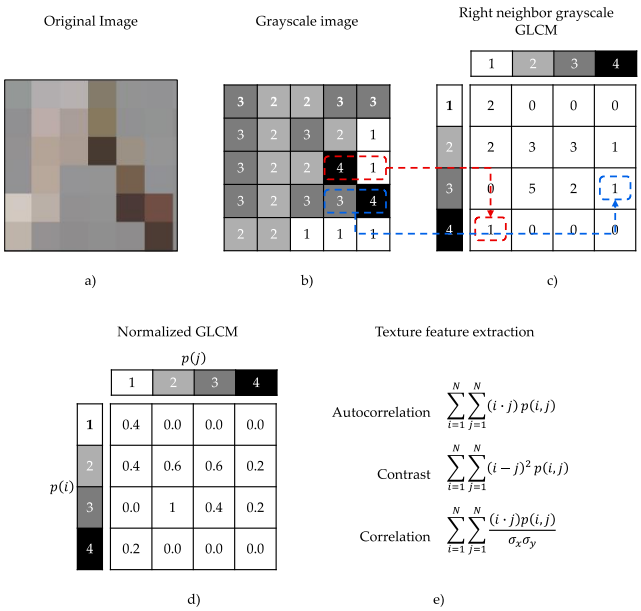


Figure 8. GLCM procedure to determine the co-occurrence matrix of gray intensity levels. (**a**) RGB image, (**b**) gray intensity levels of the RGB image, (**c**) GLCM co-occurrence matrix of the gray intensity levels, (**d**) normalized GLCM matrix between 0 and 1, (**e**) texture equations extracted from the normalized GLCM matrix.

Table 2. Notation for the calculation of Haralick texture features. Source: [33].

Num.	Notation	Meaning	Description
1	$p(i,j)$	Values i,j in the normalized GLCM	
2	N	Number of gray levels	
3	$p_x(i)$	$\sum_{j=1}^{N} p(i,j)$	
4	$p_y(j)$	$\sum_{i=1}^{N} p(i,j)$	
5	μ_x	$\sum_{i=1}^{N} i\cdot p_x(i)$	
6	μ_y	$\sum_{j=1}^{N} j\cdot p_y(j)$	
7	σ_x^2	$\sum_{i=1}^{N} (i-\mu_x)^2\cdot p_x(i)$	
8	σ_y^2	$\sum_{j=1}^{N} (j-\mu_y)^2\cdot p_y(j)$	
9	$p_{x+y}(k)$	$\sum_{i=1}^{N}\sum_{j=1}^{N} p(i,j)^2 \Big\vert i+j=k$	
10	$p_{x-y}(k)$	$\sum_{i=1}^{N}\sum_{j=1}^{N} p(i,j)^2 \Big\vert \lvert i-j\rvert=k$	
11	HX	$-\sum_{i=1}^{N} p_x(i)\cdot log p_x(i)$	Used for determining 12 and 13 equations in this work.
12	HY	$-\sum_{i=1}^{N} p_y(i)\cdot log p_y(i)$	
13	HXY	$-\sum_{i=1}^{N} p(i,j)\cdot log p(i,j)$	
14	$HXY1$	$-\sum_{i=1}^{N}\sum_{j=1}^{N} p(i,j)\cdot log\lvert p_x(i)\cdot p_y(j)\rvert$	
15	$HXY2$	$-\sum_{i=1}^{N}\sum_{j=1}^{N} p_x(i)\cdot p_y(j)\cdot log\lvert p_x(i)\cdot p_y(j)\rvert$	

Table 3 lists all the texture features used in this work. The first column represents the number of features; the second column is a name feature; in the third column, we have given an equation for the name feature.

Table 3. Haralick texture features are used in this paper.

Num.	Feature Name	Equation
1	Autocorrelation [34]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i \cdot j)p(i,j)$
2	Cluster prominence [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i+j-2\mu)^3 p(i,j)$
3	Cluster shadow [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i+j-2\mu)^4 p(i,j)$
4	Contrast [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i-j)^2 p(i,j)$
5	Correlation [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{(i \cdot j)p(i,j)}{\sigma_x \sigma_y}$
6	Difference entropy [32]	$-\sum_{k=0}^{N-1} p_{x-y}(k) log p(k)$
7	Difference variance [32]	$\sum_{k=0}^{N-1}\left(k-\mu_{x-y}\right)^2 p_{x-y}(k)$
8	Dissimilarity [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\lvert i-j\rvert \cdot p(i,j)$
9	Energy [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}p(i,j)^2$
10	Entropy [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}p(i,j) log p(i,j)$
11	Homogeneity [34]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{p(i,j)}{1+(i+j)^2}$
12	Information measure of correlation 1 [32]	$\frac{HXY-HXY1}{\max(HX,\, HY)}$
13	Information measure of correlation 2 [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\sqrt{1-exp[-2(HXY2-HXY)]}$
14	Inverse difference [35]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{p(i,j)}{1+\lvert i-j\rvert}$
15	Maximum probability [32]	$max p(i,j)$
16	Sum average μ_{x+y} [32]	$\sum_{k=2}^{2N} k p_{x+y}(k)$
17	Sum entropy [32]	$-\sum_{k=2}^{2N} p_{x+y}(k) log p_{x+y}(k)$
18	Sum square [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i-\mu)^2 p(i,j)$
19	Sum variance [32]	$\sum_{k=2}^{2N}\left(k-\mu_{x+y}\right)^2 p_{x+y}(k)$

3.5. Variable Feature Selector

The SA algorithm was proposed by Kirkpatrick in 1983 [36]. SA represents the thermodynamic process of heating and cooling metal to increase its ductility and is an optimization method to find near-optimal solutions to non-deterministic polynomial-time hardness (NP-hardness) combinatorial problems [37].

According to related work, the PCA technique is often used as a selector of relevant variables. This technique consists of describing the data in terms of new variables, called components. The components are ordered according to their explained variance, which represents the percentage of retention of the original information. However, each compo-

nent is composed of a linear combination of all the original variables. Therefore, it can be said that PCA is a dimensionality reducer and not a variable selection method.

To solve the problem, a hybrid algorithm based on the simulated annealing (SA) technique and principal component analysis (PCA) was developed, which is called SA-PCA. SA-PCA has, as its solution, a binary vector with a length of 43, which is the number of descriptors that color and texture present in this work. The initial solution is established by PCA from the percentage contribution of the component variables, with the highest percentage of variance being explained. Figure 9 shows an example of the representation of the initial solution for this work. The values 0 and 1 indicate whether a variable has been selected.

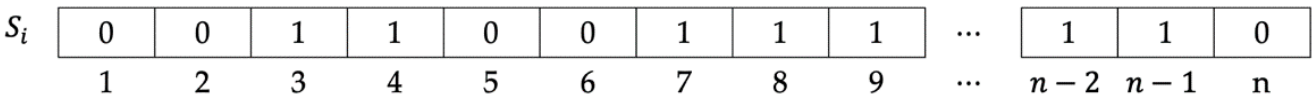


Figure 9. Representation of the initial solutions S_i in SA-PCA.

The definition of the SA-PCA parameters was subject to a tuning process [37], which is discussed in more detail in Section 4.4. Algorithm 1 shows the proposed SA-PCA algorithm, which is based on Kirkpatrick simulated annealing [36]. First, lines 2 to 5 define the initial solution, S_i, and the objective function, E_{new}, which is associated with this solution. It is defined as the best solution found so far. Line 6 verifies that the best solution found so far has reached the minimum value.

SA-PCA is defined with two principal cycles (lines 7 and 8). Here, it is traditionally checked whether the initial temperature, T_i, has reached the final temperature, T_f, and whether the metropolis cycle has reached its maximum length, L_{max}, or if there is a state of convergence. The temperature, T_i, is to be adjusted by the parameter α, line 29. The inner cycle performs a search for a new solution, X_{new}, until a stochastic equilibrium, $L_{\{max\}}$, is reached at each low temperature by the parameter β. L_{max} is adjusted by parameter β in line 30. This algorithm allows the acceptance of bad solutions by the Boltzmann acceptance criterion in line 25.

The SA-PCA algorithm has a perturbation phase, called $perturbation_{roullete}$, which includes a roulette method with the purpose of increasing the probability of selection on those variables that have been part of good solutions in the past cycles (line 9). The acceptance criterion of a solution is given by the change in the value of the objective function between the actual solution, E_{old}, and the new solution, E_{new}, i.e., $\Delta E = E_{new} - E_{old}$, where it is accepted if $\Delta E \leq 0$, as seen in lines 11, 18 to 24. Otherwise, the Boltzmann-Gibbs distribution [38], which is a decision or probability mechanism, and it is applied to determine if the bad solution is randomly accepted.

The convergence of the algorithm is defined in two cases of stable states: reaching the minimum value of the objective function, as well as stagnation. Stagnation is defined by r successive repetitions of the value of the objective function of the new solution, E_{new}, and the parameter $r = 5$. The convergence criterion in lines 33–35 means that convergence exists if the initial temperature T_i is within 5% of the final temperature T_f.

3.6. Classification

A random forest (RF) is an algorithm usually used for classification, which is composed of several decision tree classifiers, and it uses the average performance of the ensemble of classifiers to improve prediction accuracy, with the goal of optimizing the ensemble. Since the individual trees are randomly perturbed, the forest benefits from a wider exploration of the space of all possible predictors in the tree, which, in practice, result in better predictive performance [39]. The most important aspects to consider in a RF are the number of decision trees in forest, M, the function to measure the quality of the prediction, and the maximum depth of the decision trees. A decision tree with M leaves divides the feature

space into M regions, $R_m, 1 \leq m \leq M$ [40]. For each tree, the prediction function $f(x)$ is defined as:

$$f(x) = \sum_{m=1}^{M} c_m \prod(x, R_m) \tag{11}$$

$$\prod(x, R_m) = \begin{cases} 1, & if x \in R_m \\ 0, & otherwise \end{cases} \tag{12}$$

where M is the number of regions in the feature space, R_m is a region appropriate to m, and c_m is a constant suitable to m in Equation (12). The hyperparameters of the classifier RF are discussed in more detail in Section 4.3.

Algorithm 1 SA-PCA based on Kirkpatrick [36].

function SimulatedAnnealing $\left(T_i, T_f, \beta, \alpha, L_{max}, \varepsilon\right)$
$X_{old} \leftarrow solution()$
$X_{best} \leftarrow X_{old}$
$E_{old} \leftarrow objFunction()$
$E_{best} \leftarrow E_{old}$
if $(E_{best} \neq 0)$ **then**
while$(T_i > T_f$ *and* $\neg converge)$
while$(L < L_{max}$ *and* $\neg converge)$
$X_{new} \leftarrow perturbation_{roullete}(X_{old})$
$E_{new} \leftarrow objFunction(X_{new})$
$\Delta E \leftarrow E_{new} - E_{old}$
if$(E_{new} = \varepsilon)$
converge
end if
if(converge(metropoly))
converge
end if
if$(\Delta E \leq 0)$
$X_{old} \leftarrow X_{new}$
$E_{old} \leftarrow E_{new}$
if$(E_{old} < E_{best})$
$X_{best} \leftarrow X_{old}$
$E_{best} \leftarrow E_{old}$
endif
elseif $(random(0,1) > e^{\frac{-\Delta E}{T_i}})$
$X_{old} \leftarrow X_{new}$
$E_{old} \leftarrow E_{new}$
endif
$T_i \leftarrow \alpha\, T_i$
$L_{max} \leftarrow \beta\, L_{max}$
endwhile
if$\left(T_i \geq 0.95\, T_f\right)$
if(converge(Temp))
converge
endif
endif
endwhile
return X_{best}, E_{best}
endif
endfunction

4. Experimental Setup

4.1. Data

For this work, 263 images of birds were used. The set of images was divided into 193 images for training and 70 images for testing.

The image set was categorized by large and medium bird objects. The classification is based on the evaluation criteria in the competencies of the COCO [2] database, in which the object area in pixels is defined:

- Medium objects: (32×32, 96×96) pixels
- Large objects: greater than 96×96 pixels

4.2. Metric

The metric used to evaluate the performance of the model is average precision intersection over union ($APIoU$), shown in Equation (13).

$$APIoU = \sum_{i=1}^{m} \frac{TP_i}{FP_i + TP_i} \tag{13}$$

where m is the number of images, TP are the true positives, and FP are the false positives for image i. The first APIoU threshold is from 0.05 to 0.95; and the second APIoU is 0.75 to 0.95, called APIoU^{75}.

4.3. Classifier Setup

Classifier selection was performed to compare random forest and multi-perceptron classification performance using the WEKA tool. The classifiers were selected based on related work. The observations used consist of two vector sizes. The first consists of 43 features and one label, and the second consists of 14 features and one label. The latter is obtained by the selection phase of the relevant variables.

Table 4 shows the results obtained from the correct and incorrect classification of the observations. The set of observations was divided into three different sets: training set used, cross-validation, and 70–30% split. The used training set builds the classifier from all observations and re-applies all those observations to the classifier. Cross-validation splits the data into 10 sets (usually) of equal size, and each set is split into training and testing. Construct a classifier using the training data from each set, which is applied to the test data from each set to obtain an average performance. Split 70–30% is to divide the data into training and test, build the classifier with the training data, and measure performance with the test data. Table 4 shows that random forest obtained better classification performance on the three ensembles with data splitting.

Table 4. WEKA results of classifiers.

Classify	Split Data	Total Instances	Correctly Classified Instances	Incorrectly Classified Instances
Random Forest	Use training set	16,988	16,978 (99.94%)	10 (0.05%)
	Cross-validation	16,988	12,892 (75%)	4096 (24.11%)
	Split 70–30%	5096 (30%)	3853 (75.6%)	1243 (24.4%)
MLP	Use training set	16,988	11,963 (70.4%)	5025 (29.5%)
	Cross-validation	16,988	11,649 (68.5%)	5339 (31.4%)
	Split 70–30%	5096 (30%)	3494 (68.5%)	1602 (31.4%)

RF is one of the machine learning classification and regression methods. Table 5 shows the parameters of the random forest classifier. The RF method was executed using the Sklearn library. The tuning was subjected to the random grid search method. The first column lists the parameters obtained that were assigned a configuration different from the default values. The second column shows the values assigned to the parameters. The third column briefly describes each of the parameters.

Table 5. Hyperparameters of the random forest classifier.

Parameters	Value	Description
n_estimators	1400	The number of trees in the forest
max_depth	80	The maximum depth of the tree.
max_features	Auto	The number of features to consider when looking for the best split: max_features=sqrt(n_features)

4.4. Tuned SA Enhancement Algorithm

The SA algorithm is used as a randomized optimization method to find a subset of features (variables) that performs better than the original 43 features.

The parameters of this model are the following: initial temperature T_i = 1570.29, final temperature $T_f = 0.01$, metropolis length L_{max} = 198, α = 0.95, β = 1.02, and perturbation rate = 0.10.

4.5. Color and Texture Features

In this work, a total of 43 features were extracted, 26 corresponding to color and 17 corresponding to texture. The color characteristics are two measures of central tendency: standard deviation and variance. The color models are HSI, CMYK, LAB, and XYZ, which are the color characteristics being extracted for each channel. Texture characteristics are obtained by generating the GLCM matrix.

Table 6 lists the features selected by the PCA and SA techniques, the latter being the proposed enhancement algorithm. The first column shows the method, the second and third columns list the color and texture characteristics selected by each method, and the fourth column shows the total characteristics of each method.

Table 6. Features selected by the PCA and SA.

Method	Color Feature	Texture Feature	Total Feature
PCA technique	std_S, var_S, std_Y_cmyk, std_K, var_K, std_L, var_L, var_A, var_B_lab, std_X, std_Z, var_X, var_Y_xyz, var_Z	-	14
SA Enhancement Algorithm	std_H, std_S, var_I, std_M, var_C, var_K, std_L, std_A, var_A, var_Y_xyz, var_z	Correlation, diference_entropy, diference_variance	14

5. Results

We test our PSEV-BF methodology with a dataset from the COCO database. We evaluate our proposed methodology with APIoU's mean semantic segmentation metrics for medium and large objects. In this section, we show the results obtained in the tuning phase and the selection of relevant variables for SA, as well as the performance of the model's pre-segmentation and classification processes.

In the study of the better model, two distinct phases in computer vision were implemented and tested. The first phase, pre-segmentation, was a CNN architecture presented in Section 3.2, consisting of locating the regions where ROI is found; YOLOV3 was used for this purpose. The coordinates provided by YOLOV3 allow the determination of the region where the object of interest could be located, which is performed during the segmentation phase.

We used the simulated annealing (SA) algorithm as a selector for relevant variables to improve the training phase in the classification process. SA-PCA used a random forest classifier. SA configures the initial solution using the variables obtained by PCA. It also uses a perturbation method using a roulette wheel. In Table 7, we observe the solutions of

SA-PCA that were proposed. The first column indicates the number of runs, the second column shows the number of features selected in each run, and the third column is the objective function associated with the number of variables selected. The choice of a solution, i.e., a set of variables, obtained by the SA-PCA, is defined by the size of the solution and the objective function, the latter being the most important. Thus, Table 7 shows that, by using 14 variables, an objective function with a lower error rate is obtained. Therefore, these characteristics are used as relevant variables.

Table 7. Results of the SA-PCA algorithm with RF.

Number	Number Variable Selected	Objective Function
1	16	28.02
2	18	28.14
3	14	25.12
4	11	27.85
5	13	29.30
6	12	39.03
7	14	32.06
8	10	32.97
9	10	36.97
10	17	32.36
11	16	30.89
12	18	27.96
13	18	28.38
14	17	28.39
15	11	34.38
Minimal	10	25.12
Maximal	18	39.03
Standard Deviations	3.02	3.83

The classification performances for two groups of bird sizes are given in Table 4. We observe the comparative performance of the proposed methodology with different configurations: Methodology 1 (M1), Methodology 2 (M2), and Methodology 3 (M3). The M1 configuration applies the traditional processes of pre-processing, classification, evaluation, and a superpixel technique. M2 involves the same traditional process but does not use superpixels, although a variable selection method is implemented. M3 only implements a pre-segmentation phase with YOLOV3. Additionally, finally, our proposal PSEV-BF includes all the configurations proposed in this work. It is important to clarify that all methodologies use pre-segmentation with YOLOV3 for comparison purposes.

In Table 8, the first column indicates the size of the birds used: large or medium. The second column lists the different methodologies, labeled M1, M2, M3, and our PSEV-BF. The third and fourth columns indicate, with ✓, whether the superpixel technique is used in the pre-segmentation, segmentation, or enhanced feature phase, and it indicates ✗ otherwise. Finally, the last two columns are the results from APIoU with two thresholds: 0.5 to 0.95 and 0.75 to 0.95.

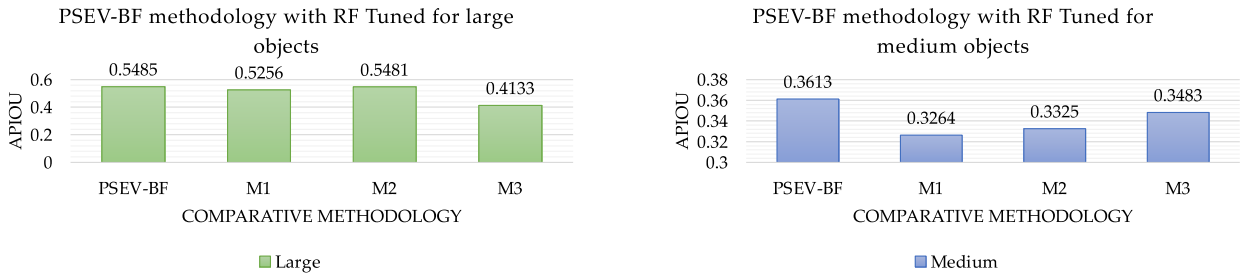
Table 8. Performance of PSEV-BF methodology vs alternative configurations.

Size Object	Method	SuperP.	Pre-S.	Seg.	Enhanced Feature	Metric	
						APIoU	$APIoU^{75}$
Large	PSEV-BF	✓	✓	✓	✓	0.5485	0.8614
	M1	✓	✓	✗	✗	0.5256	0.8235
	M2	✗	✓	✓	✓	0.5481	0.8347
	M3	✗	✓	✗	✗	0.4133	0.8459
Medium	PSEV-BF	✓	✓	✓	✓	0.3613	0.8097
	M1	✓	✓	✗	✗	0.3264	-
	M2	✗	✓	✓	✓	0.3325	0.8097
	M3	✗	✓	✗	✗	0.3483	0.8097

Super P.: super pixels; Pre-S: pre-segmentation; Seg: segmentation.

In Table 8, the results show that the proposed PSEV-BF methodology for large objects has values around 50% for the APIoU metric and 80% for the $APIoU^{75}$ metric. On the other hand, medium-sized objects have values around 36% for the APIoU metric and 80% for the $APIoU^{75}$ metric for the M2 and M1 methodologies. However, M1 for medium-sized objects did not obtain images with a higher value than the 75% threshold of the $APIOU^{75}$ metric, and they were not calculated. The average processing times obtained by the PSEV-BF in large and medium objects were 78.03 and 3.07 s, respectively. Additionally, the processing time for the M1 methodology for large objects was 90.09 s, and, for medium objects, it was 9.01. In the case of M2 and M3 methodologies, the superpixel is not included, and the time processing is not reported because the time exceeds the maximum time allowed for each image.

In Figure 10, we present the performance of the different configurations of the proposed method based on the APIoU metric with a threshold starting at 0.5 for large and medium size objects and reaching values around 50%. In Figure 10a, we show that our proposal achieves a performance of 54% accuracy for large objects, with a difference of about 12% with the M3 methodology, which achieves the lowest accuracy. The M2 and M3 methodologies obtained APIoU values very close to those of the PSEV-BF methodology. They are involved in at least two of the proposed processes: superpixel and pre-segmentation.



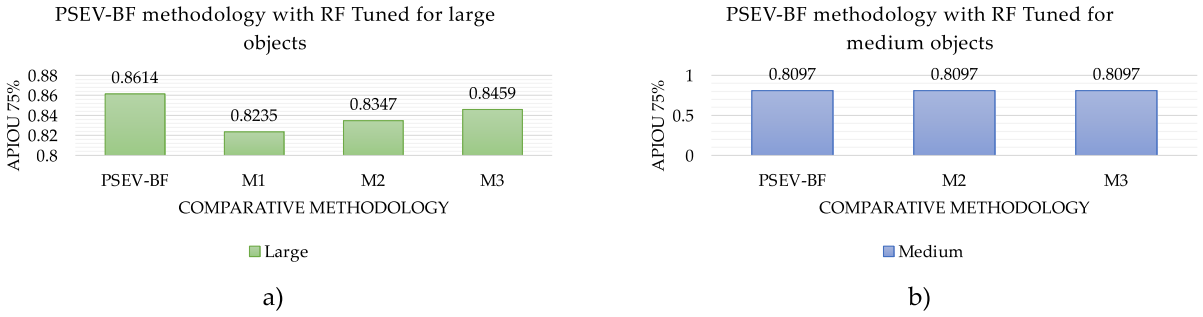
a) b)

Figure 10. Results of the methodology compared with different configurations based on the APIoU metric for (**a**) large objects and (**b**) medium objects.

In Figure 10b, we observed that the PSEV-BF achieves a performance of 36% accuracy for medium size objects, with a difference of about 4% from the M1 methodology, which achieves the lowest accuracy. The M3 methodology shows values close to those of the proposed method, and these are involved in pre-segmentation.

In Figure 11, we can observe the performance of the different configurations of the proposed method based on the APIoU metric, with a threshold starting at 0.75 for large- and medium-sized objects and reaching values around 80%. Figure 11a shows that our proposal achieves a performance of 86% accuracy for large objects, with a difference of about 4% with the M1 methodology, which achieves the lowest accuracy. The M3 methodology shows

values very close to those of the PSEV-BF methodology; these are involved in at least two of the proposed processes: superpixel and pre-segmentation.



a) b)

Figure 11. Results of the comparison of methodologies with different configurations based on the APIoU^{75} metric using (**a**) large objects and (**b**) medium objects.

In Figure 11, we present the performance of the methods for medium- and large-sized objects, considering those images that reach a threshold equal or greater than 75% (APIoU^{75}). In Figure 11b, we observe that PSEV-BF methodology, as well as M2 and M3, achieve an accuracy of 80% for medium-sized objects. On the contrary, the methodology M1 fails to obtain an accuracy above a threshold of 75%.

Figure 12 shows some examples of objects with large sizes processed. Figure 12a shows the images segmented by COCO; Figure 12b shows the adaptation resulting from the segmentation phase. Finally, Figure 12c shows some of the cases obtained using the PSEV-BF methodology. We observe that the pixels corresponding to non-birds (black blocks) are part of the background. Likewise, about 86% of the pixels corresponding to birds were correctly classified.

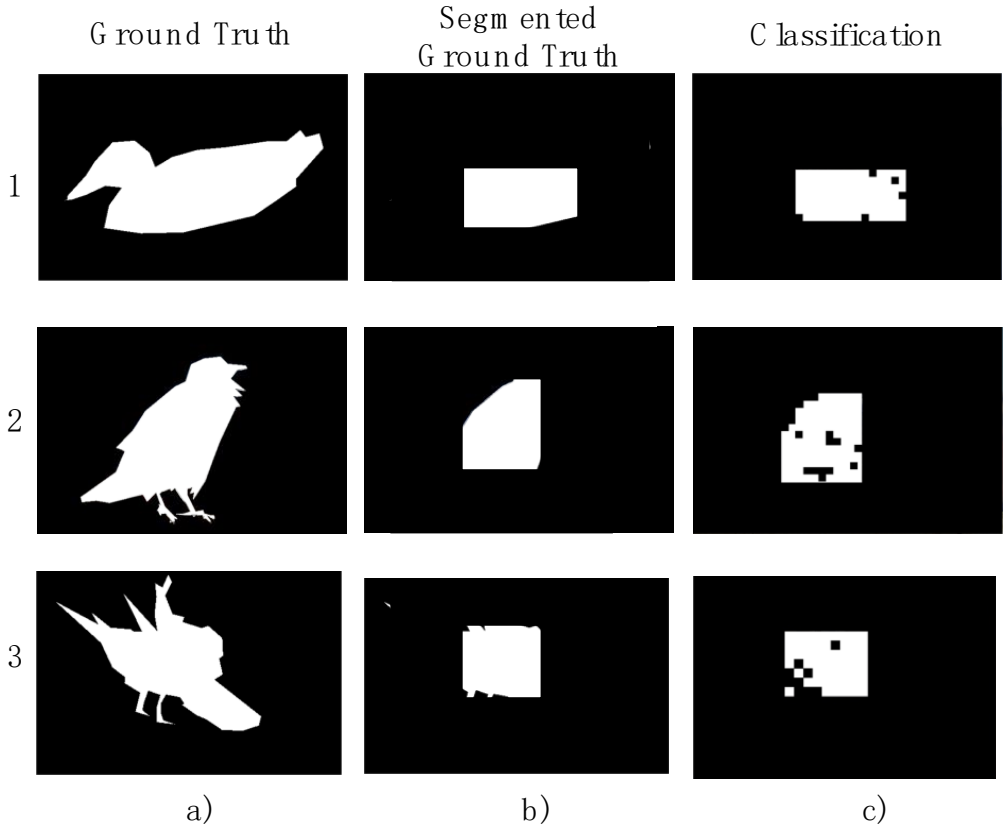


Figure 12. Comparative methodology results for three large images. (**a**) original segmented image, (**b**) segmented image by proposed segmentation, (**c**) 15 × 15-pixel window classification.

Figure 13 shows some examples of objects with large sizes that are processed with occlusion. Figure 13a shows that the white pixels correspond to birds, and the black pixels correspond to non-birds (or wrongly classified pixels). Likewise, about 86% of the pixels corresponding to birds were correctly classified.

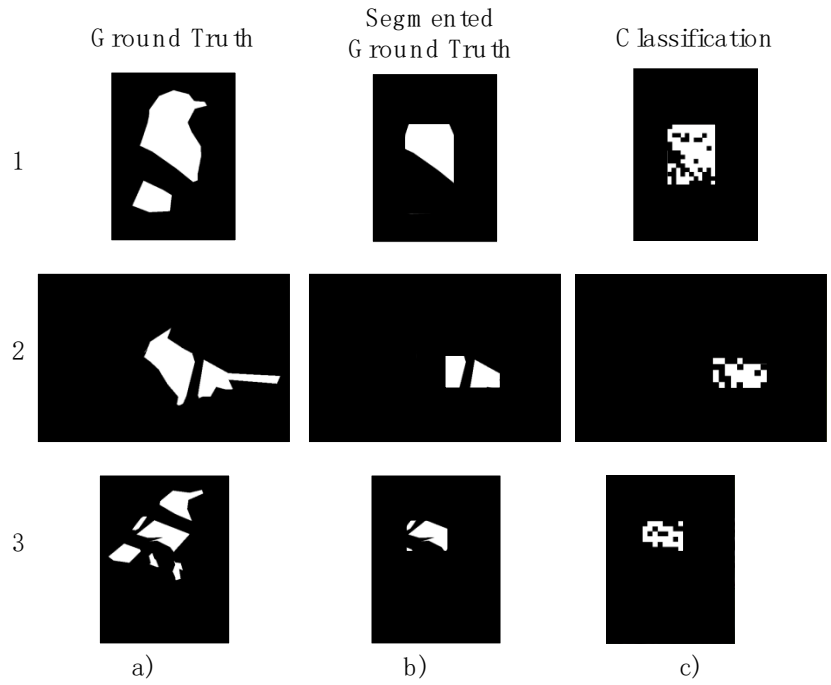


Figure 13. Comparative methodology results for three large images with occlusion. (**a**) original segmented image, (**b**) segmented image by proposed segmentation, (**c**) 15 × 15-pixel window classification.

Figure 14 shows some examples of objects with medium sizes processed by the PSEV-BF methodology. We observe, in the last column, the pixels corresponding to birds (white pixels) which were correctly classified in the first row.

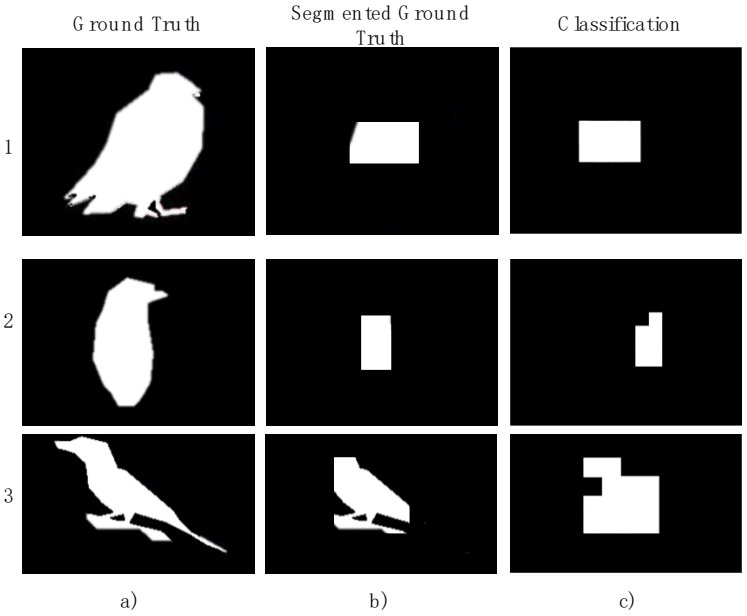


Figure 14. Comparative methodology results for three medium images. (**a**) original segmented image, (**b**) segmented image by proposed segmentation, (**c**) 15 × 15-pixel window classification. Note: the images were amplified for better illustration.

Finally, Figure 15 shows some examples of objects with medium sizes processed with occlusion by the PSEV-BF methodology. We observe, in the last column, that the pixels corresponding to birds were correctly classified.

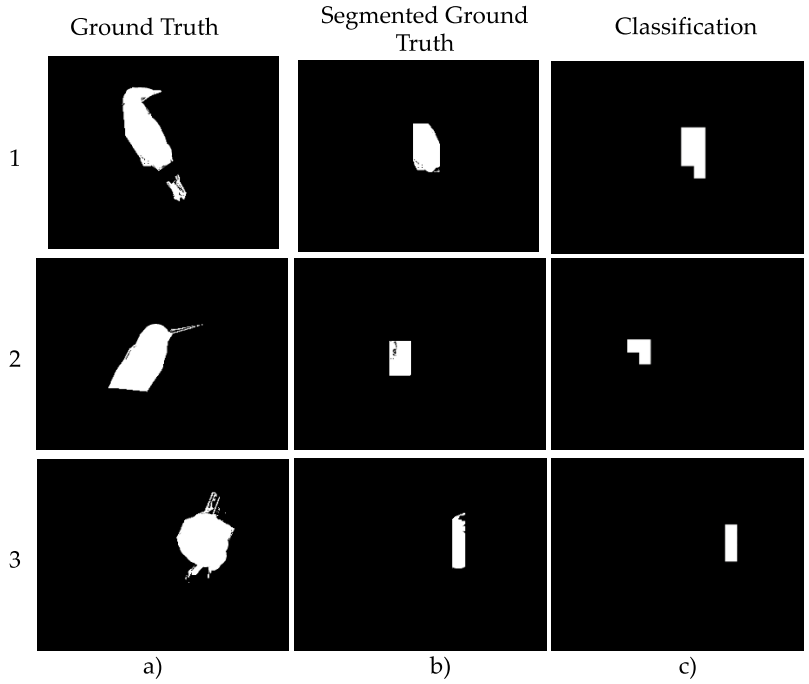


Figure 15. Comparative methodology results for three medium images with occlusion. (**a**) original segmented image, (**b**) segmented image by proposed segmentation, (**c**) 15 × 15-pixel window classification. Note: the images were amplified for better illustration.

6. Conclusions

In this paper, we presented a bird detection and classification methodology called PSEV-BF (pre-segmentation and enhanced variables for bird features), which uses pre-segmentation and a simulated annealing algorithm with principal component analysis called SA-PCA, proposed to enhance variables. PSEV-BF incorporates a new methodology compared to modern methods. Moreover, it can be applied to images with occlusions and uncontrolled environments.

The methodology of PSEV- BF consists of the phases of preprocessing, pre-segmentation, segmentation, feature extraction, relevant variables selection, and classification. Preprocessing includes histogram equalization and Gaussian filtering for image enhancement and smoothing. For pre-segmentation, a CNN detection technique, YOLOV3, was used to provide a vector of coordinates. The coordinates delineate a region that has a high probability of belonging to a bird.

Segmentation refines the coordinates obtained from pre-segmentation by redefining the given region. The inner region of the coordinates is reduced by 50% and catalogued as foreground pixels. The outer region of the coordinates is increased by 20% and cataloged as background pixels. A superpixel technique was used in feature extraction to obtain a 43-feature vector with color and texture. The superpixel technique covers an area of 15 × 15 pixels.

We compare our methodology with the traditional methodology. The methodology was tested with bird category images from the COCO database. The images were classified according to the size of the desired object: large and medium. A total of 193 images were used for training and validation of the classifier, and 70 images were used for testing. The test images are divided into large and medium groups, which correspond to 35 images per group. A total of 16,988 feature vectors were used as samples for the training and validation of the random forest classifier.

PSEV-BF was compared with the M1, M2, and M3 methodologies. These methodologies differ in configuration from the proposed methodology. For large objects, PSEV-BF and M2 show values with an approximate accuracy of 54% with the APIoU metric, whereas M2 does not have the superpixel phase. First, M1 and M2 use at least two of the proposed methods in the methodology. However, M3 does not use the proposed phases, resulting in 41% accuracy of the APIoU metric, which is the lowest value among the compared methodologies. Secondly, M2 does not use the superpixel method, which leads to a very similar accuracy value compared to PSEV-BF, while M1 has a difference of 2% compared to M2. We can say that using the proposed processes for large objects improves the accuracy of the methodology.

For objects classified as medium size, the methodology of PSEV-BF shows values with an approximate accuracy of 36% of the APIoU metric. First, M1 shows 32% accuracy, which is the lowest value among the compared methodologies. This means that the effects are very large when segmentation and enhanced variables are not used. PSEV-BF and M1 differ by 4%, and the difference is due to the use of a superpixel method. We find that, in PSEV-BF in medium-sized objects, prediction accuracy is improved.

This paper presents a methodology for pre-segmentation, the variable selection method, and feature extraction employing superpixels. Once the methodology is tuned, it can be used to solve object identification problems, for example, classification by type of bird or other objects faster than traditional methods.

For future work, we propose using similar techniques for supervised image segmentation. PSEV-BF was not designed for recognizing species of birds. We plan to incorporate other strategies for pre-segmentation, enhanced feature variables, and classification for recognizing different species.

Author Contributions: Conceptualization L.J.H.-G., J.F.-S. and J.J.G.-B.; methodology L.J.H.-G., J.F.-S., E.R-R. and J.J.G.-B.; investigation L.J.H.-G., J.F.-S. and J.J.G.-B.; Software L.J.H.-G., J.F.-S. and J.J.G.-B.; validation, J.F.-S., J.P.S.-H., D.L.H.-R. and E.R-R.; formal analysis J.F.-S., J.P.S.-H., E.R.-R. and D.L.H.-R.; writing original draft L.J.H.-G. and J.F.-S.; writing review and editing, J.F.-S., J.J.G.-B., E.R.-R. and J.P.S.-H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are openly available in Common Object in Context (COCO) at https://doi.org/10.48550/arXiv.1405.0312, in reference [2].

Acknowledgments: The authors acknowledge, with appreciation and gratitude, CONACYT and TecNM/Instituto Tecnológico de Ciudad Madero. Additionally, we acknowledge Laboratorio Nacional de Tecnologóas de la Información (LaNTI) for access to the cluster. We also thank the Asociación Mexicana de Cultura A.C.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Islam, S.; Khan, S.I.A.; Abedin, M.M.; Habibullah, K.M.; Das, A.K. Bird species classification from an image using VGG-16 network. In Proceedings of the 2019 7th International Conference on Computer and Communications Management, Bangkok, Thailand, 27–29 July 2019; pp. 38–42. [CrossRef]
2. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
3. Malhi, A.; Gao, R.X. PCA-based feature selection scheme for machine defect classification. *IEEE Trans. Instrum. Meas.* **2004**, *53*, 1517–1525. [CrossRef]
4. Lu, Y.; Cohen, I.; Zhou, X.S.; Tian, Q. Feature selection using principal feature analysis. In Proceedings of the 15th ACM international conference on Multimedia, Augsburg, Germany, 25–29 September 2007; pp. 301–304.
5. Song, F.; Guo, Z.; Mei, D. Feature selection using principal component analysis. In Proceedings of the 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization, Yichang, China, 12–14 November 2010; Volume 1, pp. 27–30.
6. Uçar, M.K. Classification performance-based feature selection algorithm for machine learning: P-Score. *IRBM* **2020**, *41*, 229–239. [CrossRef]

7. Gu, S.; Cheng, R.; Jin, Y. Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Comput.* **2018**, *22*, 811–822. [CrossRef]
8. Adhao, R.; Pachghare, V. Feature selection using principal component analysis and genetic algorithm. *J. Discret. Math. Sci. Cryptogr.* **2020**, *23*, 595–602. [CrossRef]
9. Koutanaei, F.N.; Sajedi, H.; Khanbabaei, M. A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring. *J. Retail. Consum. Serv.* **2015**, *27*, 11–23. [CrossRef]
10. Kavitha, R.; Kannan, E. An efficient framework for heart disease classification using feature extraction and feature selection technique in data mining. In Proceedings of the 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), Pudukkottai, India, 24–26 February 2016; pp. 1–5. [CrossRef]
11. Gárate-Escamila, A.K.; el Hassani, A.H.; Andrès, E. Classification models for heart disease prediction using feature selection and PCA. *Inform. Med. Unlocked* **2020**, *19*, 100330. [CrossRef]
12. Abiodun, E.O.; Alabdulatif, A.; Abiodun, O.I.; Alawida, M.; Alabdulatif, A.; Alkhawaldeh, R.S. A systematic review of emerging feature selection optimization methods for optimal text classification: The present state and prospective opportunities. *Neural Comput. Appl.* **2021**, *33*, 15091–15118. [CrossRef]
13. Chen, R.-C.; Dewi, C.; Huang, S.-W.; Caraka, R.E. Selecting critical features for data classification based on machine learning methods. *J. Big Data* **2020**, *7*, 52. [CrossRef]
14. Liang, Y.; Zhang, M.; Browne, W.N. Image feature selection using genetic programming for figure-ground segmentation. *Eng. Appl. Artif. Intell.* **2017**, *62*, 96–108. [CrossRef]
15. Nasir, M.; Khan, M.A.; Sharif, M.; Lali, I.U.; Saba, T.; Iqbal, T. An improved strategy for skin lesion detection and classification using uniform segmentation and feature selection based approach. *Microsc. Res. Tech.* **2018**, *81*, 528–543. [CrossRef]
16. Sharif, M.; Khan, M.A.; Iqbal, Z.; Azam, M.F.; Lali, M.I.U.; Javed, M.Y. Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection. *Comput. Electron. Agric.* **2018**, *150*, 220–234. [CrossRef]
17. Rehman, Z.U.; Naqvi, S.S.; Khan, T.M.; Arsalan, M.; Khan, M.A.; Khalil, M.A. Multi-parametric optic disc segmentation using superpixel based feature classification. *Expert Syst. Appl.* **2019**, *120*, 461–473. [CrossRef]
18. Rath, S.; Kumar, S.; Guntupalli, V.S.K.; Sourabh, S.M.; Riyaz, S. Analysis of deep learning methods for detection of bird species. In Proceedings of the 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 23–25 February 2022; pp. 234–239. [CrossRef]
19. Hong, S.-J.; Han, Y.; Kim, S.-Y.; Lee, A.-Y.; Kim, G. Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery. *Sensors* **2019**, *19*, 1651. [CrossRef] [PubMed]
20. Xiang, W.; Song, Z.; Zhang, G.; Wu, X. Birds detection in natural scenes based on improved faster RCNN. *Appl. Sci.* **2022**, *12*, 6094. [CrossRef]
21. Mashuk, F.; Sattar, A.; Sultana, N. Machine Learning Approach for Bird Detection. In Proceedings of the 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 4–6 February 2021; pp. 818–822. [CrossRef]
22. Akçay, H.G.; Kabasakal, B.; Aksu, D.; Demir, N.; Öz, M.; Erdoğan, A. Automated bird counting with deep learning for regional bird distribution mapping. *Animals* **2020**, *10*, 1207. [CrossRef] [PubMed]
23. Öztürk, A.E.; Erçelebi, E. Real UAV-bird image classification using CNN with a synthetic dataset. *Appl. Sci.* **2021**, *11*, 3863. [CrossRef]
24. Wang, H.; Xu, Y.; Yu, Y.; Lin, Y.; Ran, J. An efficient model for a vast number of bird species identification based on acoustic features. *Animals* **2022**, *12*, 2434. [CrossRef]
25. Ou, Y.-Q.; Lin, C.-H.; Huang, T.-C.; Tsai, M.-F. Machine learning-based object recognition technology for bird identification system. In Proceedings of the 2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan), Taoyuan, Taiwan, 28–30 November 2020; pp. 1–2. [CrossRef]
26. Kumar, A.; Das, S.D. Bird species classification using transfer learning with multistage training. In *Workshop on Computer Vision Applications*; Springer: Singapore, 2018; pp. 28–38. [CrossRef]
27. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 3rd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2008.
28. Ketcham, D.J. Real-time image enhancement techniques. *Image Process.* **1976**, *74*, 120–125.
29. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
30. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
31. Zhang, X.; Gao, Y.; Wang, H.; Wang, Q. Improve yolov3 using dilated spatial pyramid module for multi-scale object detection. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420936062. [CrossRef]
32. RHaralick, M.; Shanmugam, K.; Dinstein, I. Textural features for image classification. *IEEE Trans. Syst. Man Cybern.* **1973**, *SMC-3*, 610–621. [CrossRef]
33. Brynolfsson, P.; Nilsson, D.; Torheim, T.; Asklund, T.; Karlsson, C.T.; Trygg, J.; Nyholm, T.; Garpebring, A. Haralick texture features from apparent diffusion coefficient (ADC) MRI images depend on imaging and pre-processing parameters. *Sci. Rep.* **2017**, *7*, 4041. [CrossRef] [PubMed]

34. Soh, L.-K.; Tsatsoulis, C. Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. *IEEE Trans. Geosci. Remote Sens.* **1999**, *37*, 780–795. [CrossRef]
35. Clausi, D.A. An analysis of co-occurrence texture statistics as a function of grey level quantization. *Can. J. Remote Sens.* **2002**, *28*, 45–62. [CrossRef]
36. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef]
37. Sanvicente-Sánchez, H.; Frausto-Solís, J. A method to establish the cooling scheme in simulated annealing like algorithms. In *International Conference on Computational Science and Its Applications*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 755–763.
38. Boltzmann, L. The second law of thermodynamics. In *Theoretical Physics and Philosophical Problems*; Springer: Berlin/Heidelberg, Germany, 1974; pp. 13–32.
39. Genuer, R.; Poggi, J.-M. Random forests. In *Random Forest with R*; Springer: Cham, Switzerland, 2020; pp. 33–55. [CrossRef]
40. Sain, S.R.; Vapnik, V.N. The nature of statistical learning theory. *Technometrics* **1996**, *38*, 409. [CrossRef]