Check for updates

Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 9729343, 12 pages
https://doi.org/10.1155/2022/9729343

*Research Article*

# Postprocessing of Edge Detection Algorithms with Machine Learning Techniques

**Pablo Flores-Vidal**,[1] **Javier Castro**,[2] **and Daniel Gómez**[2]

[1]*Producción Animal, Facultad de Veterinaria, Universidad Complutense de Madrid, Madrid 28040, Spain*
[2]*Estadística y Ciencia de los Datos, Facultad de Estudios Estadísticos, Universidad Complutense de Madrid, Madrid 28040, Spain*

Correspondence should be addressed to Pablo Flores-Vidal; pflores@ucm.es

Received 5 October 2021; Revised 3 January 2022; Accepted 21 February 2022; Published 23 March 2022

Academic Editor: Jude Hemanth

Copyright © 2022 Pablo Flores-Vidal et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, machine learning (ML) techniques are applied at an early stage of Image Processing (IP). The learning procedures are usually applied from at least the image segmentation level, whereas, in this paper, this is done from a lower processing level: the edge detection level (ED). The main objective is to solve the edge detection problem through ML techniques. The proposed methodology is based on a classification of edges made pixel by pixel, but the predictors employed for the ML task include information about the pixel neighborhood and structures of connected pixels called edge segments. The Sobel operator is employed as input. Making use of 50 images that belong to the Berkeley Computer Vision data set, the average performance of the validation sets when employing our Neural Networks method reached an F-measure significatively higher than with the Sobel operator. The experiment results show that our post-processing technique is a promising new approach for ED.

## 1. Introduction

Digital Image Processing or Image Processing (IP) evolved during the sixty's thanks mainly to space programs like Range 7, conducted by the Jet Propulsion Laboratory [1]. IP encompasses many different tasks ranging from simple ones, for instance, enhancement or smoothing, to the most complex ones related to the understanding of a complete visual scene. This graduality can be roughly represented as a continuum (see Figure 1).

Edge detection (ED) is one of the main IP techniques, and it has found applications in a wide range of tasks, such as pathological diagnostics in medicine [2, 3], with a special focus in tumoral discovering, as well as remote sensing [4], which is useful for agriculture and biology, and more recently for research related to climate change. Other relevant fields in which ED has been applied successfully are the military industry, surveillance [5], and others [6–10].

In the last decades, machine learning techniques have started to be employed for solving IP problems [11–16]. Nevertheless, it is not easy to find learning approaches employed for low-level processing tasks, for instance, IP tasks lower than contour detection [17, 18]. An example of a close technique to ED is the use of Convolutional Neural Networks (CNN) for low-medium level tasks such as corner detection [19]. However, corner detection can be considered a more complex task than ED. This task is useful in video processing for tracking objects. An example of research making use of supervised techniques on the ED problem can be seen in [20], where the aim was classifying edge segments–linked structures of edges–. In that research, the goal was limited in the sense that the edges could not be classified locally—pixel by pixel— an approach that is specifically explored in this work. Due mainly to that limitation, in this paper, an algorithm that performs a classification at the pixel level is proposed. This approach allows classifying a pixel as "edge" or "nonedge" considering the pixel information combined with other local information which includes its neighborhood and the segment to which that pixel belongs. Moreover, this paper's approach does not consist of taking the feature vector from the whole image (or a subimage) as it has been done in so many researches so far (and that can be

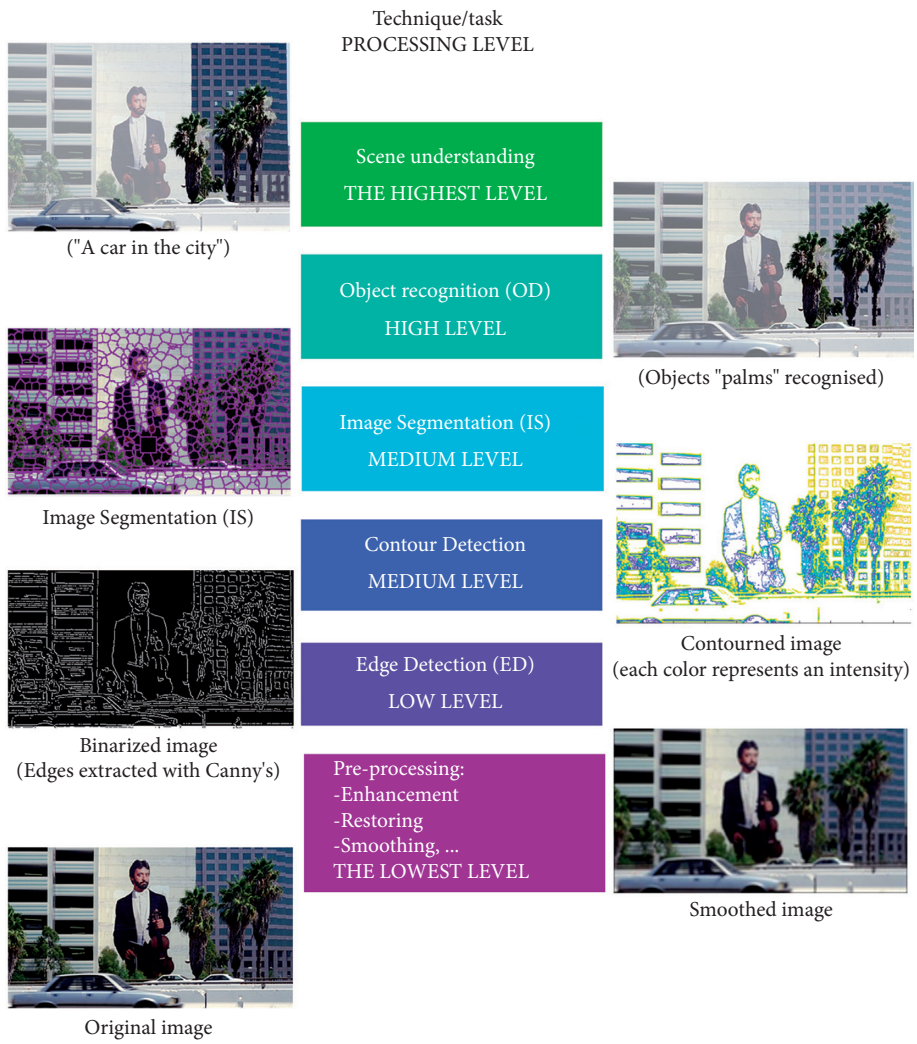



Figure 1: Different processing levels of image processing tasks.

considered the state of the art). Instead of that, the feature vector is created for each pixel.

ML has been traditionally used mainly for applied research, which is logically associated with high-level processing tasks (see Figure 1). As it is shown in this paper, there are several advantages for using ML for ED purposes.

This paper is structured as follows: the first three subsections of the next section are devoted to some needed preliminaries, including the basis of a digital image, ED problem, and ML techniques. "The proposed methodology" is focused on our proposal, which can be considered as a postprocessing technique for ED. In the first place, the problem of building a new ground-truth version is explored (see "Building a suitable ground-truth"), secondly, the construction of the variable predictors (see "Collecting the predictor variables"), and finally in the experiment phase ("Using Machine Learning for Edge detection Problems"). The last sections are devoted to the results ("Results and Discussion") and conclusions, respectively.

## 2. Edge Detection with Machine Learning Techniques

*2.1. Digital Image.* Let us denote a digital image by $I$, and its pixel coordinates in the *spatial domain* by $(i, j)$. For clarity's sake, the coordinates are integers, where each point $(i, j)$ represents a pixel with $i = 1, \ldots, n$ and $j = 1, \ldots, m$. Therefore, the size of an image, $n \times m$, is the number of its horizontal pixels multiplied by its number of verticals. The value of the spectral information depends on the digital image type that is considered. Some of these image types are:

(i) *Binary map* ($I^{\text{bin}}$): $I_{\{i,j\}} \in \{0, 255\}$ (as well it is usually expressed as $I_{\{i,j\}} \in \{0, 1\}$).

(ii) *Grayscale image* ($I^{\text{gray}}$): $I_{i,j} \in \{0, 1, \ldots, 255\}$.

(iii) *Soft image* ($I^{\text{soft}}$): $I_{i,j} \in [0, 1]$. This is as well referred as a normalized grayscale image.

In this paper, we deal with binary maps, grayscale images, and soft images.

26292, 2022, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2022/9729343 by Spanish Cochrane National Provision (Ministerio de Sanidad), Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

*2.2. Edge Detection Problem.* The main goal of ED algorithms is to detect those pixels in which the intensity change is significant. An ED algorithm transforms an image into a binary image. In this binary image, the white pixels (or "one values," as they are usually presented) represent those pixels that have been identified by the ED algorithm as edge pixels. On the contrary, the black pixels represent those identified as non-edge pixels.

In a digital image $I$, a pixel $(i, j)$ is considered as an edge if in this pixel there is an important change of the intensity function $I_{i,j}$ [3, 21]. An Edge Detector is an algorithm that takes the image $I$ as an input and then converts it into a binary image–edge map– $(I_{i,j}^{\text{solution}})$ [22].

A well-known example of ED algorithm is the Sobel operator [23, 24]. It performs only through two ED steps (see [11] for more information about the ED steps): the feature extraction and the blending or aggregation of the features–channels–. Mathematically, it works as follows: given a pixel $(i, j)$ and $i \geq 2, \mathrm{j} \geq 2$, we define

$$S_x\left(I_{i,j}\right) = \sum_{a=i-1}^{i+1} \sum_{b=j-1}^{j+1} S^x I_{a,b}, \tag{1}$$

as the horizontal component, and

$$S_y\left(I_{i,j}\right) = \sum_{a=i-1}^{i+1} \sum_{s=j-1}^{j+1} S^y I_{a,b}, \tag{2}$$

as the vertical one. The horizontal and vertical masks are, respectively, these two:

$$S^x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, S^y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}.$$

Throughout the blending phase, both components are aggregated into a single value named the gradient. The most common aggregation function for computing the gradient is the Euclidean distance.

$$I_{i,j}^{\text{soft-Sobel}} = \sqrt{\left(S_x\left(I_{i,j}\right)\right)^2 + \left(S_y\left(I_{i,j}\right)\right)^2}. \tag{3}$$

Finally, this soft image (We consider that the blended image is as well normalized, and then it is a soft image.) is binarized through the last ED step: the scaling. The most common procedure for this last step is the use of a threshold value $\alpha \in (0, 1)$.

$$I_{i,j}^{\text{bin-Sobel}} = \begin{cases} 1, & \text{if } I_{i,j}^{\text{soft-Sobel}} \geq \alpha, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

*2.3. Supervised Classification Problem.* The goal of supervised learning is to build a concise model to classify items into known classes in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances, where the values of the predictor features are known but the value of the class label is unknown [25]. It is possible to find a huge number of classification algorithms that have the common aim of maximizing the considered accuracy measures depending on a specific problem or dataset.

The classification problem can be expressed mathematically as follows [25]. Let be $x^i$ the predictor variables with $i \in (1, \ldots, k)$ in such a way that they configure the measure space $X$ with. The predictor $x^i$ can be either a quantitative or a categoric variable. The aim is to classify a certain case $l$ from the sample, then $l \in (1, \ldots, n)$. Every case takes $(x_l^1, x_l^2, \ldots, x_l^k)$ as the vector's value. This vector provides the information that is going to be used for the classifier to assign the case to any of the $J$ previously defined classes $j = \{1, \ldots, J\}$. The key point of this classifying process is to build the classification rule that is going to assign to any case a specific class.

A classifier is an algorithm that from the set $X$ with $J$ classes $A_1, \ldots, A_J$ assigns any nonlabelled case $x$ to a single class, and this classification is made by means of its characteristics.

The classifiers can be supervised or unsupervised. The supervised ones make use of example cases already tagged belonging to the training set, from which the algorithm builds—"learns"—the classification rules [26]. Contrarily, in the case of unsupervised algorithms, all cases used for the learning process are unlabeled. Supervised and unsupervised learning are both global techniques inside what is called machine learning.

In this paper, we have focused on three well-known supervised classifiers such as Logistic regression, Neural Networks (NN), and Random Forest (RF) [27]. These classifiers are then used for binary classification as there are two possible outcomes for a given pixel: "edge" or "non-edge." A novel methodology that combines ML with ED is presented throughout the next two sections.

*2.4. The Proposed Methodology.* Firstly, a data set of images with their respective ground-truth is needed. The usual approach is that this ground-truth is made of "true" edges previously drawn by at least one human. The Berkeley Image Segmentation Data Set [28] is a well-known example that precisely works this way. The pixels labels are built with the information provided by the ground-truth as it is explained in the Subsection "Building a suitable ground-truth."

For the learning process, in the literature, the usual approach for edge detection problem has been employing a few characteristics related to the intensity of each pixel if the image consists of a single channel, or a few intensities in the case of color edge detection. One of the advantages of using supervised ML techniques against nonsupervised methods is that they can deal with more information. Thus, more variables can be included in the process of classifying the edges, so a better classification is expected.

Traditionally, this classification of edges—called "extraction" of edges as well—is made by finding a suitable value for the parameters involved in the ED algorithm, which could be done by means of nonsupervised or supervised methods. In this research, multiple predictor variables are used together to provide the needed information

26:29, 2022, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2022/9729343 by Spanish Cochrane National Provision (Ministerio de Sanidad), Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

for taking the decision pixel by pixel. These predictors are collected from the soft value provided by an edge detection algorithm as an input. In this sense, the proposed method is not competing with other existing algorithms, as it is working as a post-processing technique with the capability of improving them. This soft value is obtained right after the blending phase and before the scaling phase. The predictor variables that are being used for the ML models are specified in the "Collecting the predictor variables" Subsection.

As it is usual in ML, once the models are trained, the true label is not needed anymore, as the labels are being predicted employing the soft value as input. Due to this, once the model has been trained, the complexity barely increases over its original level that was reached to generate the input.

*2.5. Building a Suitable Ground-Truth.* To work with supervised classification, the use of labeled cases is needed. In IP, the use of ML techniques requires employing a labeled ground-truth as a reference that may provide the true classes.

Let $I_{i,j}^{\text{Gt–human}_k}$ be one of the $K$ different humans made references, i.e., "sketches," that are available for each image as a ground-truth. Then, for a specific human or "draftsman" a pixel is labeled as an "edge" if it was drawn by him/her, and as a "nonedge" if it was not drawn. As every image has K human references, they must be aggregated into one for ML purposes. In this research, the next aggregated label is used for the learning process: the pixel is considered as an edge if it has been drawn by at least $h$ humans (Obviously, for applying this kind of aggregation a data set of images with more than one reference is needed. This is the case of Berkeley segmentation data set [19].). This image, which has been created as an aggregation of multiple humans, tends to avoid the subjective tendency of a single human, minimizing the likeliness of building wrong labels. However, this "raw" aggregated human reference is not going to be the definitive image from which the labels are built. The reason for this lies in the great difficulty for any human to draw with absolute precision the location of a certain "edge." During the traditional matching process employed in Image Processing (see for example the method presented in [29]) to match a pixel that is being evaluated with its equivalent pixel in the human reference, a window of $n \times n$ pixels is allowed. For example, a $3 \times 3$ pixels window (i.e., $\pm$ 1 pixel deviation from the central or possible "true edge" pixel) it seems a natural range to allow some tolerance for classifying the candidate pixel as an edge in a permissive or a "soft" way. Because of this, the above created $I_{i,j}^{\text{label}}$ must be expanded or thickened to cover this window of $n \times n$ pixels and converted into the definitive label image $I_{i,j}^{\text{labelExt}}$. An easy method to build this thickened image is through mathematical morphology [30]. More specifically, a dilation ($\oplus$) is applied over $I_{i,j}^{\text{label}}$ with a square$_n$, which is a square of $n \times n$ white pixels acting as a structural element.

The combination of both steps is made in this way: first, a dilation is applied over the different human references and then the aggregation of these dilated sketches is built. This process for building the definitive experiment's ground-truth is mathematically expressed by equations (5) and (6), and it is shown in Figures 2 and 3:

$$I_{i,j}^{\text{dilatedhuman}_k} = I_{i,j}^{\text{Gt–human}_k} \oplus \text{square}_n, \tag{5}$$

$$I_{i,j}^{\text{label}} = \begin{cases} 1, & if \sum_{k=1}^{K} I_{i,j}^{\text{dilatedhuman}_k} \geq h, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

*2.6. Collecting the Predictor Variables.* In this research, the predictor variables are extracted from the soft image $I_{i,j}^{\text{soft–Sobel}}$, which results after Sobel operator is applied over a gray image, and before the binarization—scaling—is done. The type of variables being used is divided into four different categories:

*2.7. Pixel Predictor Variables.* These are the variables that use information related to the pixel. The next ones are the pixel predictor variables that have been used.

(i) $v_1$: the intensity of gray channel $I_{i,j}^{\text{soft–Sobel}}$ which results after the blending phase (aggregation of directions) but before the scaling phase (see the ED phases in [31]). This value or intensity can be considered as the pixel quality traditionally called "edginess."

(ii) $v_2$: the potential edge angle $\theta$, which is computed as in the case of Sobel (see equations (1) and (2)). It is a circular variable, so a practical way to use the angle information when building the ML classifiers is to decompose this variable in two components: $\sin(\theta)$ and $\cos(\theta)$.

(iii) $v_3$ and $v_4$: the pixel horizontal $i$ and vertical positions $j$, which gives spatial information that is relevant for the classification.

(iv) $v_5$ and $v_6$: two variables for the minimum distances from the pixel position to the horizontal borders and the vertical borders, respectively.

*2.8. Neighborhood Predictor Variables.* These are the predictor variables related to the pixel neighborhood. The most natural neighborhood variable is the soft edge detection value of the neighbor pixels or any aggregation of these values. Clearly, for creating these variables, a specific window size for the neighborhood must be decided. In the case of this research a neighborhood grade of 1—$3 \times 3$ pixels—is used, therefore, a pixel has 3, 5, or 8 different neighbors (depending on the pixel position). The neighborhood variables employed were:

(i) $v_7$ to $v_{14}$: the $3 \times 3$ neighbor's value of $I_{i,j}^{\text{soft–Sobel}}$

(ii) $v_{15}$ to $v_{17}$: maximum, minimum, and average intensity of the 8 neighbor pixels

(iii) $v_{18}$ to $v_{20}$: maximum, minimum, and average intensity of the 3 neighbor pixels of the pixel's row

26292, 2022, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2022/9729343 by Spanish Cochrane National Provision (Ministerio de Sanidad), Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

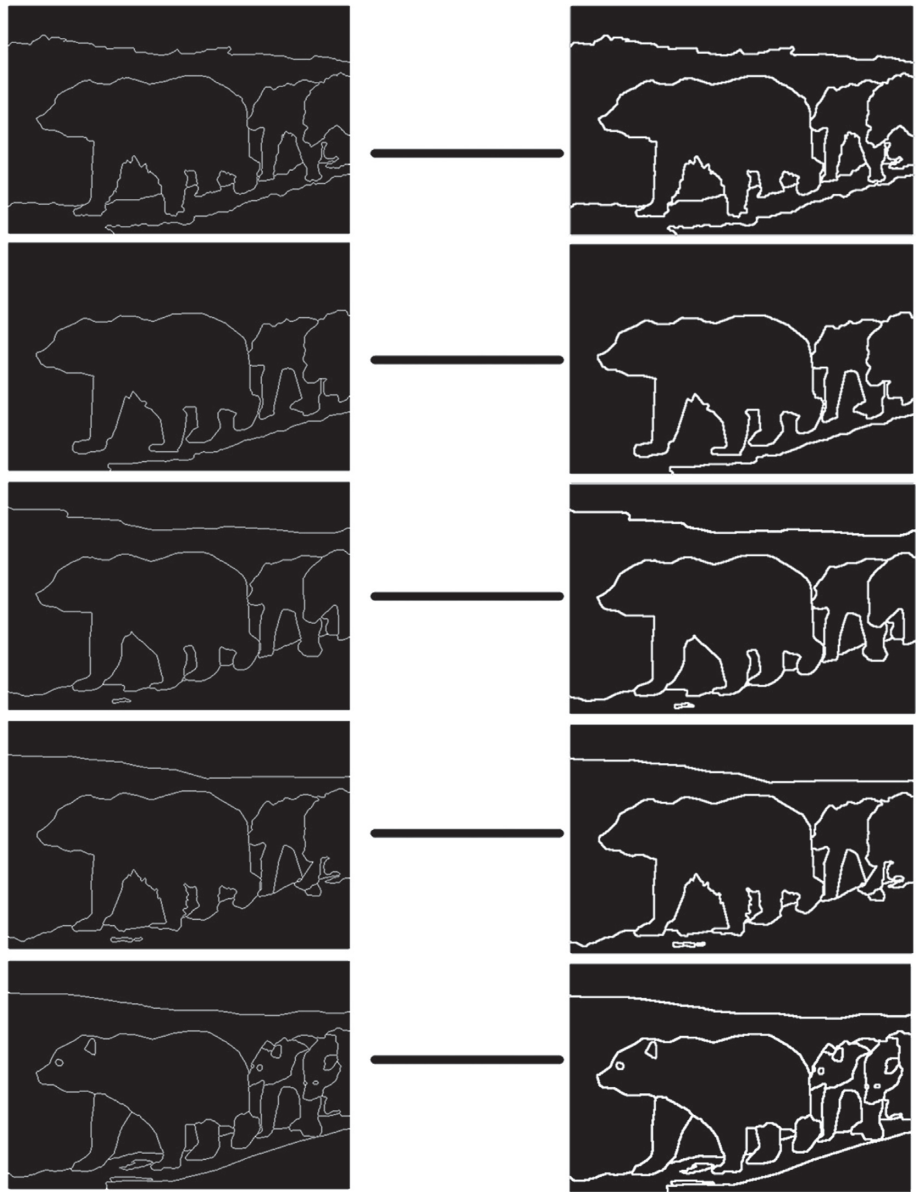


Figure 2: The first step: the human sketches are dilated.

(iv) $v_{21}$ to $v_{23}$: maximum, minimum, and average intensity of the 3 neighbor pixels of the pixel's column

(v) $v_{24}$ to $v_{29}$: maximum, minimum, and average intensity of the 3 neighbor pixels of both diagonals

*2.9. Segment Predictor Variables.* This set of variables is obtained from the edge segments that result from connecting the pixel candidates of $I_{i,j}^{\text{soft-Sobel}}$. In [20, 31] was presented the concept of edge segments. When a certain pixel did not belong to a segment, these variables values were set to 0. Some of the segment's valuable information is related to its length, average position, rectangle containing it, and so on. For the experiment results of this paper the next segment variables have been employed:

(i) $v_{30}$: length. For each segment $S_l$, $x_1^l = \text{Length}_l = |S_l|$. Therefore, it can be seen as the number of pixels in the segment.

(ii) $v_{31}$: intensity mean. For each segment $S_l$, $x_2^l = IM_l = \sum_{(i,j)\in S_l} I_{i,j}^{\text{thin}}/x_l^1$, where $I_{i,j}^{\text{thin}}$ represents the intensity of pixel $(i, j)$ inside the already thinned image (see ED phases in [31]).

(iii) $v_{32}$: maximum edginess. For each segment $S_l$, we obtain $x_3^l = \text{Max}\{I_{i,j}^{\text{thin}}: (i, j) \in S_l\}$.

(iv) $v_{33}$: standard deviation of the intensity. For each segment $S_l$: $x_5^l = \sigma_l = \sum_{i,j\in S_l} (I_{i,j}^{\text{thin}} - x_2^l)^2/x_1^l$.

(v) $v_{34}$: "rule of thirds" position. For each segment $S_l$, we obtain the coordinates of the pixel that occupies the central position in the segment:

26:29, 2022, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2022/9729343 by Spanish Cochrane National Provision (Ministerio de Sanidad), Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

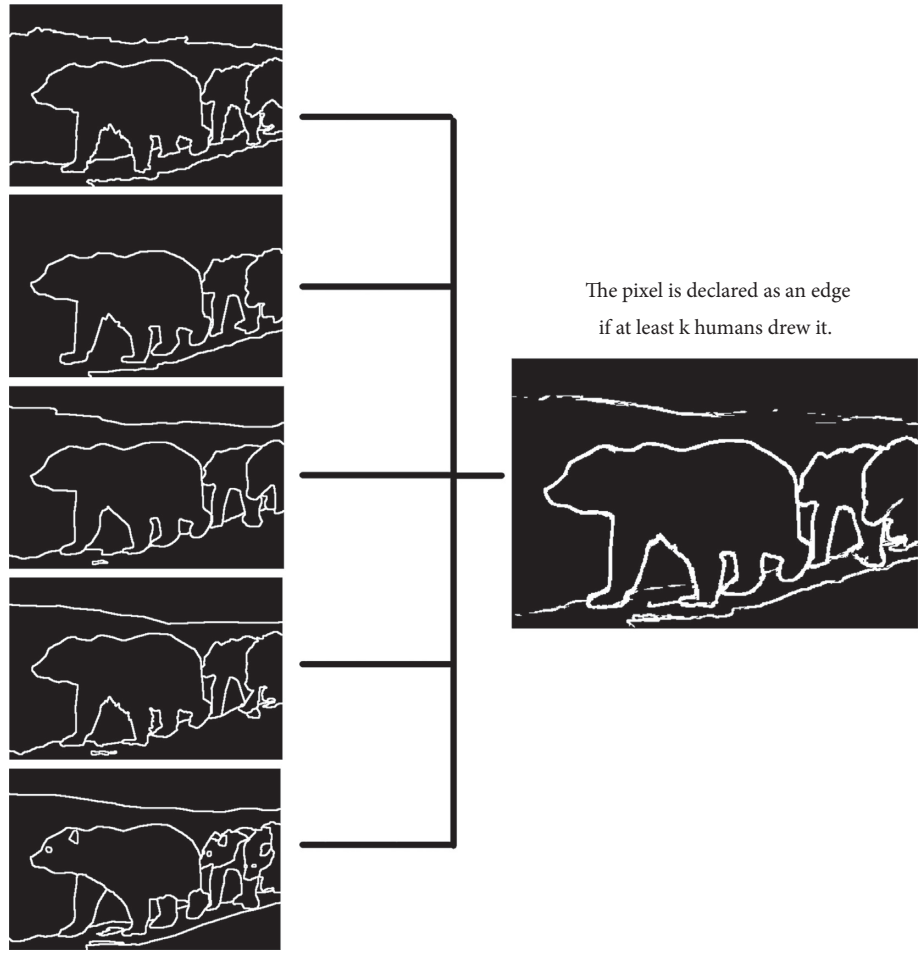



Figure 3: The second step: the dilated sketches are aggregated into a single image.

$$\left(x_6^l, x_7^l\right) = \text{Central}_l, \tag{7}$$

where $x_6^l$ is the average vertical position and $x_7^l$ is the average horizontal position of the pixels in $S_l$:

$$\begin{aligned} x_6^l &= \frac{\sum_{i,j=\left((i,j)_1,(i,j)_2\right)\in S_l}(i,j)_2}{x_1^l}, \\ x_7^l &= \frac{\sum_{(i,j)=\left((i,j)_1,(i,j)_2\right)\in S_l}(i,j)_1}{x_1^l}\frac{1}{2}. \end{aligned} \tag{8}$$

Once the gravity center is computed, its Euclidean distance to the intersection points following the rule of thirds is obtained, which is a standard in photography composition [32]. This rule establishes that the most important objects inside an image are usually placed close to the intersection of the lines that divide the image into three equal parts. Following this principle, we computed the minimum of its four distances, as there are four intersection points created by these four lines.

(i) $v_{35}$: the area of the minimum rectangle that contains the segment to which the pixel belongs.

(ii) $v_{36}$: the belonging itself of the pixel $(i, j)$ to a certain segment. $\text{Thin}(i, j) = 1$ if $I_{i,j}^{\text{thin}} \rangle 0 \cdot \text{Thin}(i, j) = 0$ otherwise.

*2.10. General Predictor Variables.* These are the variables whose value depends over the whole image, i.e., they provide general information about the image and not of a certain pixel. These variables must be used carefully as if each image had a different value for a certain general variable. This could be wrongly used by the ML method to identify the specific image, which would not be correct for classification purposes. An example of a general variable is the average soft value inside the image or another aggregation function over the soft values, the percentage of the pixels upper a given soft value, the image dimension, and so on.

In the experimental results of this paper, this type of variables has not been used. The reason for this is to avoid the image identification tendency that could affect badly the

26:29, 2022, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2022/9729343 by Spanish Cochrane National Provision (Ministerio de Sanidad), Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

ML classification, especially when the training data set size is not big enough.

*2.11. Collecting the Predictors and the Labels.* Below is specified a pseudo-code that helps understanding the process of collecting the predictor variables and the labels for each pixel.

(1) $I$ = read(imagefile) #an image I is taken as input.
(2) NumHumans = $h$ #required number of humans for drawing $(i, j)$ as an edge.
(3) For every pixel $(i, j)$ of $I$ do
(4) Compute variable 1 ($v_1$ = "The intensity of gray channel $I_{i,j}^{\text{soft–Sobel}}$")
(5) Compute variable 2 ($v_2$ = "The potential edge angle $\theta$.")
. . .
(42) Compute variable 36 ($v_{36}$ = "The belonging itself of the pixel $(i, j)$ to a certain segment.")
(43) FeatureVector = $(v_1, \ldots, v_{36})$
(44) Label$(i, j) = 1$ if $\sum_{k=1}^{K} I_{i,j}^{\text{dilatedhuman}_k} \geq$ NumHumans (else Label$(i, j) = 0$)
(45) end

As a result of the above code compilation, a vector combining the feature vector and its respective label are created for each pixel. An example of this can be seen in Figure 4.

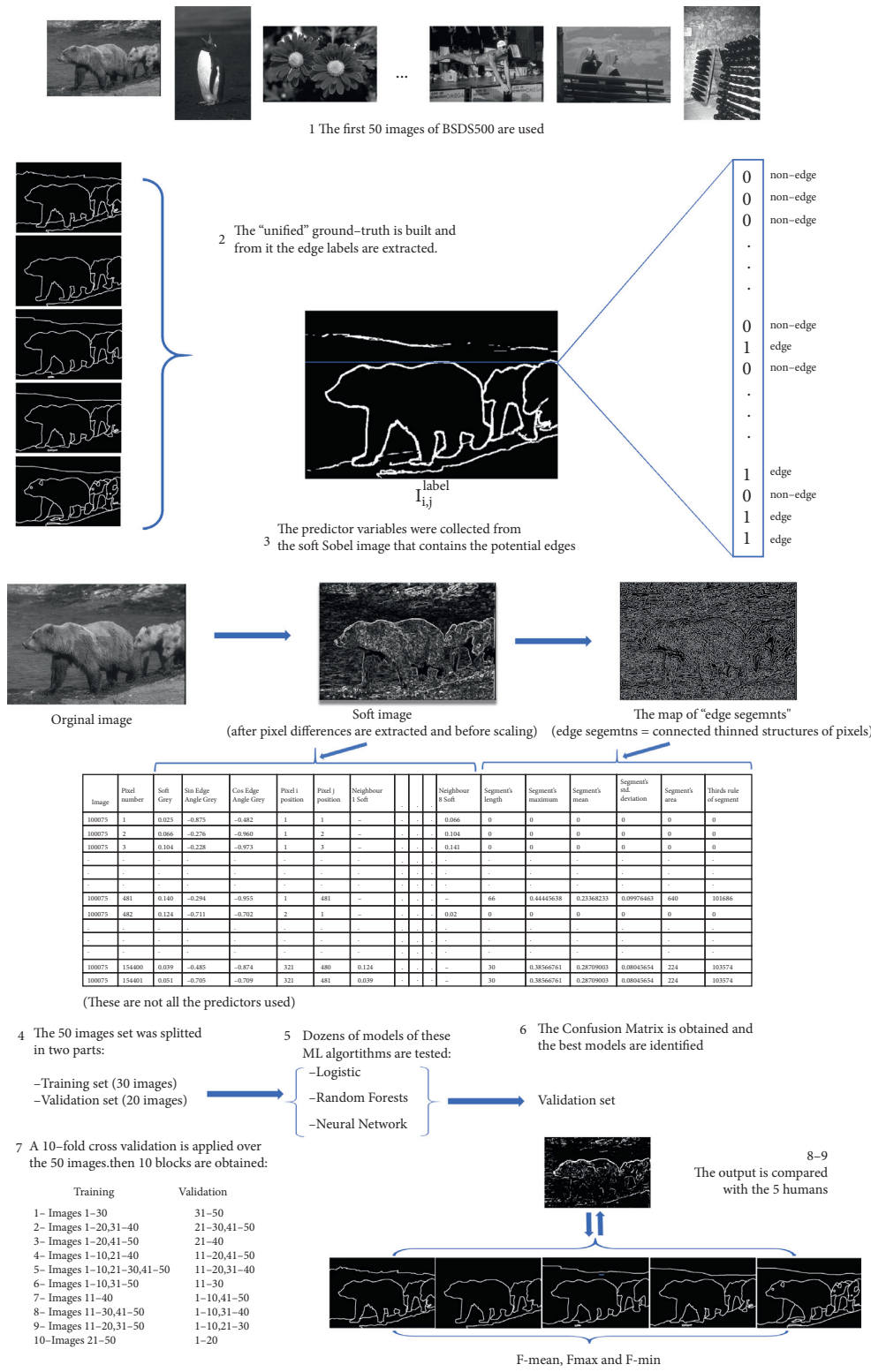
| Image | Pixel number | $V_1$ | $V_2$ | ... | $V_{36}$ | label |
|---|---|---|---|---|---|---|
| bear.jpg | 1 | 0.025 | 0.875 | ... | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... |
| bear.jpg | 154401 | 0.051 | 0.705 | ... | 0 | 0 |

Figure 4: The feature vector and the labels.

*2.12. Using Machine Learning for Edge Detection Problems.* The experiment was conducted following the next steps:

(1) The 50 first images—sorted by number—of Berkeley segmentation data set [28] were employed (from 100075.jpg to 16052.jpg).
(2) A modified version of the ground-truth of these images was created following the method explained in "Building a suitable ground-truth" Subsection (see Figure 2) and it was used for creating the pixel labels. It was employed $h = 2$ (2 humans) and $n = 3$ (the dilation was made with squares of $3 \times 3$ pixels).
(3) The predictor variables explained were extracted taking as input the soft value obtained after applying the Sobel operator. As a result of this step, a matrix with all cases—i.e., the pixels—as rows, and 36 predictor variables as columns was created (The size of this matrix was 7720050 rows $\times$ 36 columns.).
(4) The 50 images set was split into two parts: the first 30 images were used as the training set, while the rest configured the validation set.
(5) Three machine learning (ML) algorithms were employed over the training test: Logistic regression, Neural networks (NN) and Random forests (RF). A fourth algorithm was as well tested: the SVM, but the training set was smaller than with the other algorithms, and due to that, it has not been compared with the others. Dozens of models were tested, which were created by means of changing the values of the parameters of these algorithms. In the case of logistic regression all predictor variables were included, and three well-known methods for selecting the variables were employed: Forward Selection, Backward Elimination and their combination Stepward Selection [33]. Moreover, these methods were applied using different $p$ values. Finally, three different functions were used: Logic, Probit and Cloglog. NN models were all built with one single hidden layer but varying the number of nodes from one to seven. As well, three different activation functions were employed: Arctangent, Exponential and Hyperbolic tangent. The independent variables used for these NN models were those chosen by the best logistic model. The main intention for not employing all predictor variables was removing multicollinearity in NN models. Finally, RF models [27] used all possible predictors as this kind of models deals properly with multicollinearity. For the RF algorithm the number of trees was limited to 100, the number of variables per tree was ranged from 5 to 30, the train fraction was settled to 0.6, the leaf size ranged from 30 to 100000, the maximum depth was 50 and three different $p$ values were employed: 0.01, 0.05 and 0.1.
(6) For the validation set the confusion matrix was computed. This was done 15 times as the classification is made using 15 different scoring thresholds (from 1% to 15% of the highest scorings). For each algorithm, the model that reaches the highest true positive rate is considered the best one.
(7) Following a 10-fold cross-validation method (In the case of SVM models the training size was shortened to 25000 pixels for simplification purposes as computational costs of these models are higher than the others.), the 50 images were divided into 5 blocks of 10 images each, which resulted in 10 different combinations of training sets of 30 images and validation sets of 20 images.
(8) For the 10 cross-validation combinations of the previous step, and for the best three models plus Sobel's, three different F-measures were computed for each image: "$F$ minimum," representing the most different human, "$F$ mean," representing the average human, and "$F$ maximum," representing the closest human. The matching between the outputs and the humans was made allowing a tolerance of $3 \times 3$ pixels window, following the procedure proposed in [29].

26292, 2022, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2022/9729343 by Spanish Cochrane National Provision (Ministerio de Sanidad), Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License




| Image | Pixel number | Soft Grey | Sin Edge Angle Grey | Cos Edge Angle Grey | Pixel i position | Pixel j position | Neighbour 1 Soft | . | . | . | Neighbour 8 Soft | Segment's length | Segment's maximum | Segment's mean | Segment's std. deviation | Segment's area | Thirds rule of segment |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 100075 | 1 | 0.025 | −0.875 | −0.482 | 1 | 1 | – | . | . | . | 0.066 | 0 | 0 | 0 | 0 | 0 | 0 |
| 100075 | 2 | 0.066 | −0.276 | −0.960 | 1 | 2 | – | . | . | . | 0.104 | 0 | 0 | 0 | 0 | 0 | 0 |
| 100075 | 3 | 0.104 | −0.228 | −0.973 | 1 | 3 | – | . | . | . | 0.141 | 0 | 0 | 0 | 0 | 0 | 0 |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 100075 | 481 | 0.140 | −0.294 | −0.955 | 1 | 481 | – | . | . | . | – | 66 | 0.44445638 | 0.23368233 | 0.09976463 | 640 | 101686 |
| 100075 | 482 | 0.124 | −0.711 | −0.702 | 2 | 1 | – | . | . | . | 0.02 | 0 | 0 | 0 | 0 | 0 | 0 |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |
| 100075 | 154400 | 0.039 | −0.485 | −0.874 | 321 | 480 | 0.124 | . | . | . | – | 30 | 0.38566761 | 0.28709003 | 0.08045654 | 224 | 103574 |
| 100075 | 154401 | 0.051 | −0.705 | −0.709 | 321 | 481 | 0.039 | . | . | . | – | 30 | 0.38566761 | 0.28709003 | 0.08045654 | 224 | 103574 |

Figure 5: A general scheme of the experiment.

26(29, 2022, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2022/9729343 by Spanish Cochrane National Provision (Ministerio de Sanidad), Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Table 1: Statistical differences between methods.

| Classifier/ algorithm | Human max $F$ | AUC max (from 1% to 15%) | Human mean $F$ | AUC mean (from 1% to 15%) | Human min $F$ | AUC min (from 1% to 15%) |
|---|---|---|---|---|---|---|
| Logistic | 0.524 | 0.496 | 0.381 | 0.359 | 0.323 | 0.305 |
| Neural network (NN) | **0.552** | **0.524** | **0.403** | **0.381** | **0.346** | **0.326** |
| Random forests (RF) | 0.547 | 0.517 | 0.400 | 0.376 | 0.345 | 0.323 |
| Sobel-grey | 0.487 | 0.457 | 0.356 | 0.335 | 0.304 | 0.285 |

Bold values represent the highest values reached for each column.

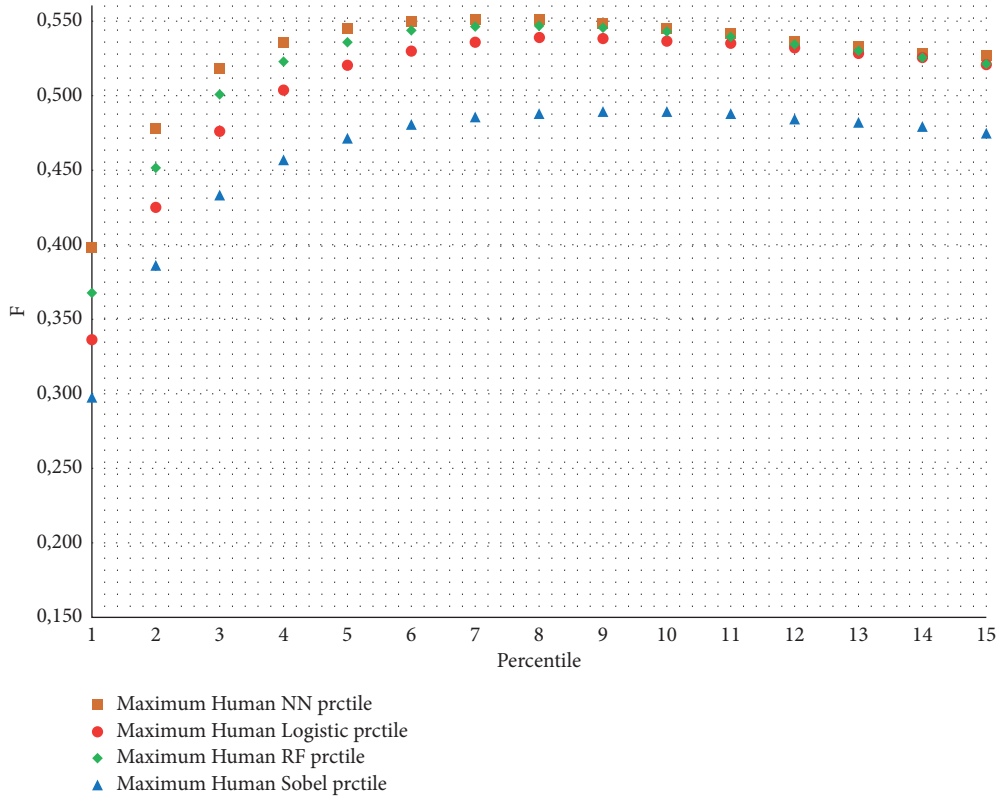



Figure 6: Logistic, NN, and RF performances versus sobel performance (maximum human).

(9) The different F values of each algorithm were compared.

We can see the scheme of the whole experiment in Figure 5.

## 3. Results and Discussion

Table 1 shows that the new proposed methodology based on the use of post-processing of edge detection output with machine learning techniques improves Sobel performance by 13% ($F = 0.403$ against $F = 0.356$). As an example of this, we can see in Figures 6–8 that Neural Network models outperform both, Random Forests and logistic models.

In Table 2 we can see that Logistic, RF, and NN performances were significantly better than Sobel's performance. Moreover, between the ML algorithms, NN reached the best performance.

The best Logistic model found employed 19 variables: $\{v_1, v_2, v_5 - v_7, v_{14} - v_{16}, v_{19}, v_{22}, v_{24}, v_{25}, v_{27}, v_{30}, v_{31}, v_{33} - v_{36}\}$. Furthermore, it used the Logit function, The Stepwise method, a $p$ value of $10^{-14}$ and the best scoring threshold found was 8%. The best RF model found had a leaf size of 100, a maximum depth of 50, a $p$ value of 0.01 and the best scoring threshold found was 8%. Finally, the best NN model employed the same 19 variables listed above, used arc tangent as the activation function, 5 nodes, and the best scoring threshold found was 10%.

We can see in Figure 9 an example of three binarized images after learning the edges through the different classifiers. It can be appreciated that the best quality of edge extraction was reached by the Neural Networks method, followed closely by Random Forest. These edges can be considered "better" edges especially because of the strong continuity that they show.

26292, 2022, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2022/9729343 by Spanish Cochrane National Provision (Ministerio de Sanidad), Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

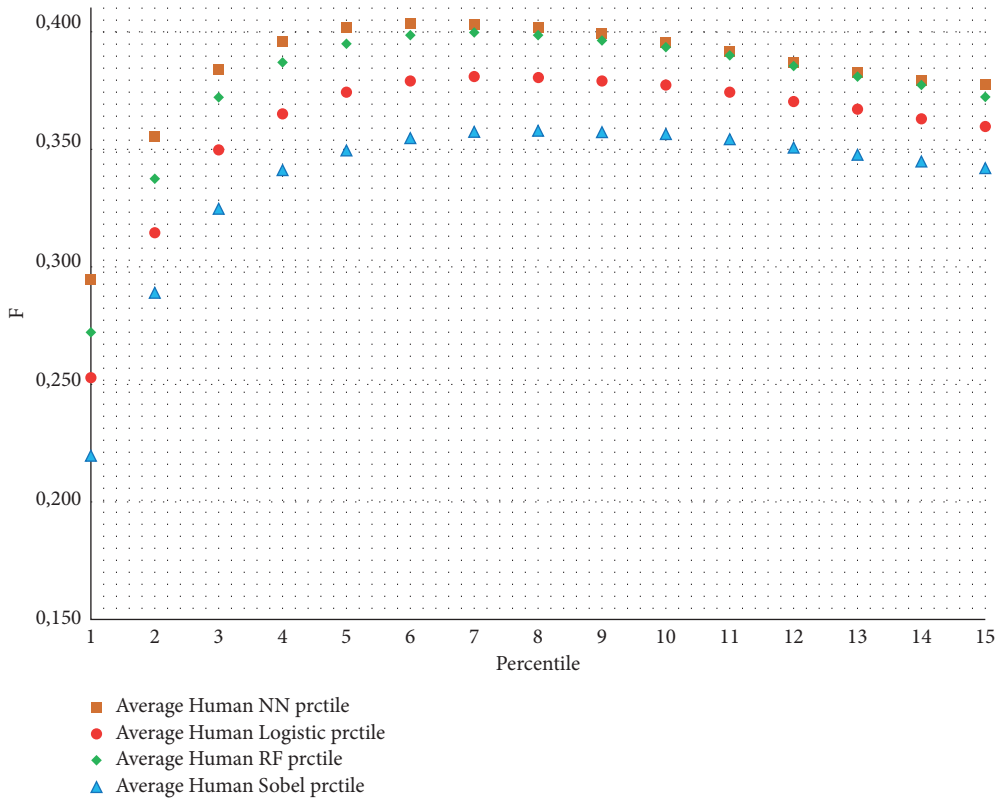



Figure 7: Logistic, NN, and RF performances versus sobel performance (average human).

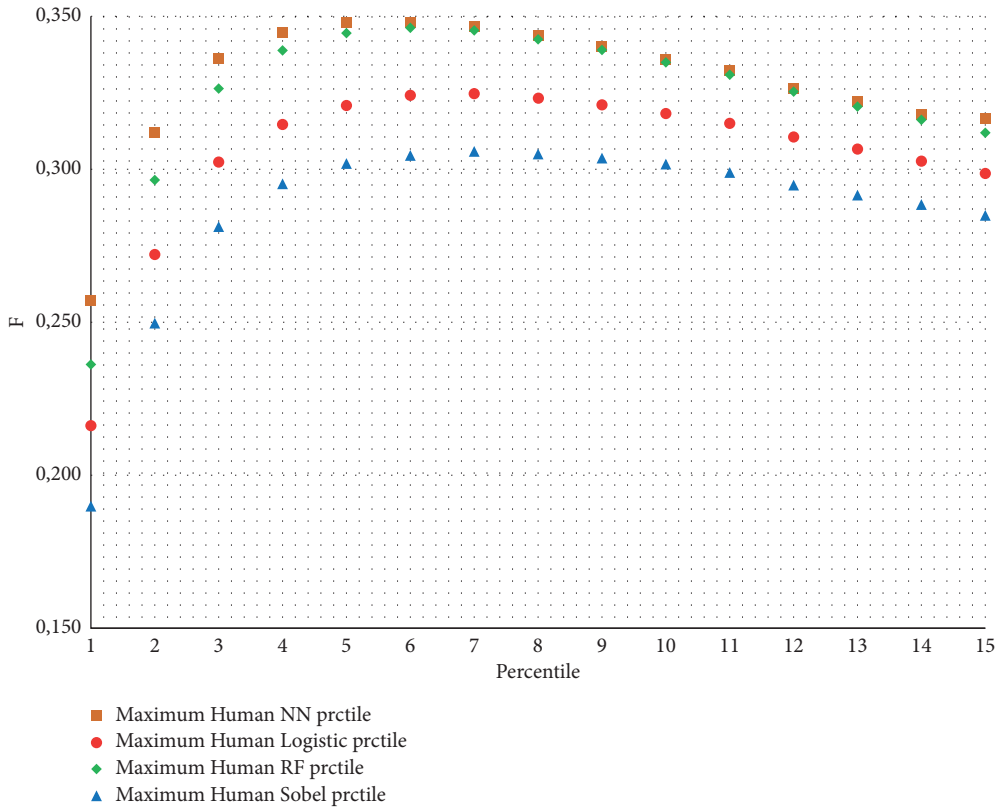



Figure 8: Logistic, NN, and RF performances versus sobel performance (minimum human).

2629, 2022, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2022/9729343 by Spanish Cochrane National Provision (Ministerio de Sanidad), Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Table 2: Statistical differences between methods.

| | Log > sobel | RF > sobel | RN > sobel | RF > Log | RN > Log | RN > RF |
|---|---|---|---|---|---|---|
| Number of images with superiority | 166 | 182 | 193 | 171 | 191 | 146 |
| Percentage of superiority | 83.0% | 91.0% | 96.5% | 85.5% | 95.5% | 73.0% |
| Probability (significance) | **0.000** | **0.000** | **0.000** | **0.000** | **0.000** | **0.000** |

Bold values represent the significant p-values.

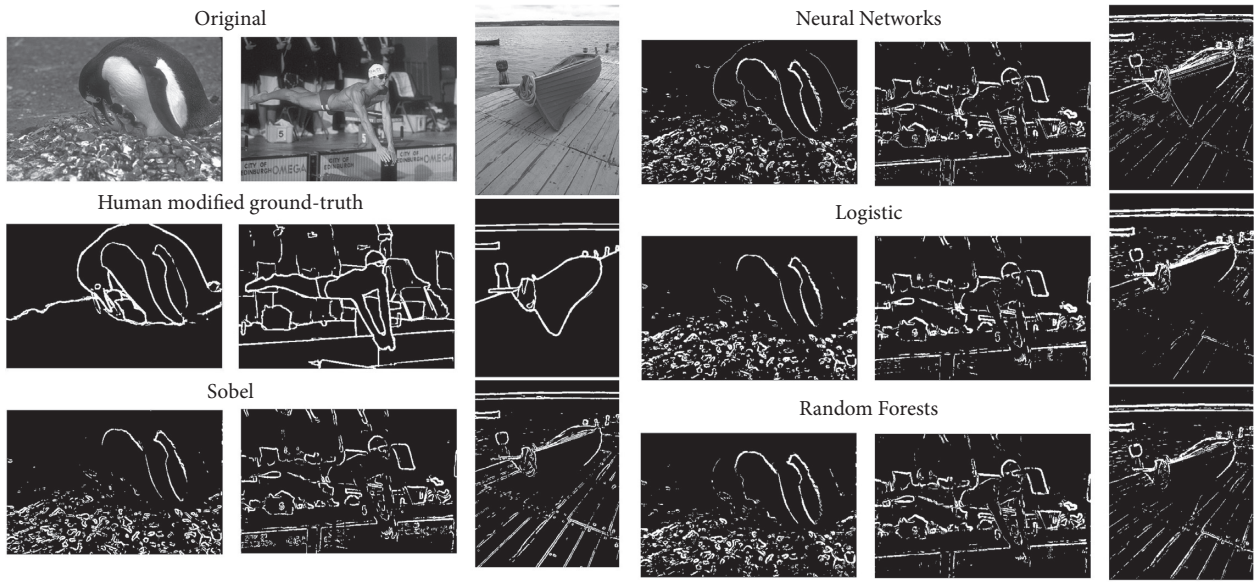



Figure 9: Three examples among the outputs of the algorithms.

## 4. Conclusions

The proposed methodology has interesting and promising results that deserve deeper research. Our postprocessing approach that makes use of machine learning techniques showed that it is possible to improve an edge detection output as Sobel's. Furthermore, the same procedure seems to be easily adapted to many other ED algorithms.

The edges extracted by our method proved to be more relevant than the ones obtained through classic ED, as they managed to keep more valuable information related to the important objects of the image. This improvement was possible thanks to an intelligent modification of the original ground-truth, which can be considered an interesting novelty of the present research. Such an approach was possible after allowing thicker than one-pixel thick edges. The utility of creating these thickened or dilated edges is justified by the fact that human vision is not able of matching a certain edge to its exact pixel location.

An immediate extension of this research may explore the ED problem with any other ED algorithm as Canny's [34]. In the case of the Canny algorithm this may require the inclusion of different predictor variables than with Sobel's.

A natural extension of this research would be based on considering more channels/colors with the extra difficulty of aggregating these channels information in an intelligent and useful way. Another extension could consist of running more tests with SVM so the results can be similar to the other algorithms allowing suitable comparisons.

Finally, it seems promising the idea of developing deep learning methods that made use of single pixels as input.

## Data Availability

The data used to support the findings of this study have been deposited in the data.world repository (https://data.world/pflores/edge-detection-with-machine-lerarning). The instructions to work with it are included in it. The set of images that is used in this experiment was developed for the first time in [19], and it can be downloaded from the resources available in Berkeley Computer Vision Group's web: https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html#bsds500.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

For the conducting of this research, the CODE created by Kermit Research Unit has been strongly helpful [5]. This research has been partially supported by the Spanish Ministry of Science (PGC2018-096509-BI00).

26:29, 2022, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2022/9729343 by Spanish Cochrane National Provision (Ministerio de Sanidad), Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

## References


[1] R. C. González and R. E. Woods, *Digital Image Processing*, Pearson Education, New York, NY, USA, 3rd edition, 2008.

[2] G. Gao, X. Wan, S. Yao, Z. Cui, C. Zhou, and X. Sun, "Reversible data hiding with contrast enhancement and tamper localization for medical images," *Information Sciences*, vol. 385-386, pp. 250–265, 2017.

[3] M. Sonka, "IEEE transactions on medical imaging statement of editorial policy," *IEEE Transactions on Medical Imaging*, vol. 33, no. 4, 2014.

[4] J. B. Campbell and R. H. Wynne, *Introduction to Remote Sensing*, Guilford Press, New York, NY, USA, 2011.

[5] P. Rosin, "Thresholding for change detection," in *Proceedings of the Sixth International Conference on Computer Vision*, pp. 274–279, IEEE, Bombay, India, January 1998.

[6] M. Fathy and M. Y. Siyal, "An image detection technique based on morphological edge detection and background differencing for real-time traffic analysis," *Pattern Recognition Letters*, vol. 16, no. 12, pp. 1321–1330, 1995.

[7] S. K. Pal and R. A. King, "On edge detection of x-ray images using fuzzy sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, pp. 69–77, 1983.

[8] I. Perfilieva, P. Hodáková, and P. Hurtík, "Differentiation by the F-transform and application to edge detection," *Fuzzy Sets and Systems*, vol. 288, pp. 96–114, 2016.

[9] P. Vlasánek and I. Perfilieva, "The f-transform in terms of image processing tools," *Journal of Fuzzy Set Valued Analysis*, vol. 2016, no. 1, pp. 54–62, 2016.

[10] T. Zielke, M. Brauckmann, and W. Vonseelen, "Intensity and edge-based symmetry detection with an application to car-following," *CVGIP: Image Understanding*, vol. 58, no. 2, pp. 177–190, 1993.

[11] V. K. Ha, J. Ren, X. Xu, S. Zhao, G. Xie, and V. M. Vargas, "Deep learning based single image super-resolution: a survey," in *Proceedings of the International Conference on Brain Inspired Cognitive Systems*, pp. 106–119, Springer, Xi'an, China, July 2018.

[12] Z.-W. He, L. Zhang, and F.-Y. Liu, "Discostyle: multi-level logistic ranking for personalized image style preference inference," *International Journal of Automation and Computing*, vol. 17, no. 5, pp. 637–651, 2020.

[13] Y.-H. Wu, Y. Liu, L. Zhang, and M.-M. Cheng, "Salient object detection via extremely-downsampled network," 2020, https://arxiv.org/pdf/2012.13093.

[14] C. Zhao, Y. Hao, S. Sui, and S. Sui, "A new method to detect the license plate in dynamic scene," in *Proceedings of the 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 414–419, IEEE, Enshi, China, May 2018.

[15] L. Gou, H. Li, H. Zheng, H. Li, and X. Pei, "Aeroengine control system sensor fault diagnosis based on CWT and CNN," *Mathematical Problems in Engineering*, vol. 2020, Article ID 5357146, 12 pages, 2020.

[16] H. Jingzhong, X. Kewen, Y. Fan, and Z. Baokai, "Strip steel surface defects recognition based on SOCP optimized multiple Kernel RVM," *Mathematical Problems in Engineering*, vol. 2018, Article ID 9298017, 8 pages, 2018.

[17] P. Dollar and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 8, pp. 1558–1570, 2014.

[18] C. López-Molina, *The Breakdown Structure of Edge Detection: Analysis of Individual Components and Revisit of the Overall Structure*, Ph.D. thesis, Universidad Pública de Navarra, Pamplona, Spain, 2012.

[19] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the European Conference on Computer Vision*, pp. 430–443, Springer, Graz, Austria, May 2006.

[20] P. A. Flores-Vidal, D. Gómez, J. Montero, and G. Villarino, "Classifying segments in edge detection problems," in *Proceedings of the 2017 12th International Conference on Intelligent Systems Y Knowledge Engineering (ISKE)*, pp. 1–6, JiangSu, China, November 2017.

[21] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[22] D. Ziou and S. Tabbone, "Edge detection techniques-an overview," *Pattern Recognition and Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii*, vol. 8, pp. 537–559, 1998.

[23] I. Sobel, "Camera Models and Machine Perception," Technical report, Computer Science Department, Technion, Israel, 1972.

[24] I. Sobel, *History and Definition of the So-Called "Sobel Operator," More Appropriately Named the Sobel-Feldman Operator*, Universitetet i Linkoping, Linkoping, Sweden, 2014.

[25] L. Breiman, *Classification and Regression Trees*, Routledge, England, UK, 2017.

[26] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised classification algorithms in machine learning: a survey and review," in *Advances in Intelligent Systems and Computing*, vol. 937, pp. 99–111, Springer, 2020.

[27] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[28] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images y its application to evaluating segmentation algorithms y measuring ecological statistics," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 416–423, Cambridge, MA, USA, June 2001.

[29] F. J. Estrada and A. D. Jepson, "Benchmarking image segmentation algorithms," *International Journal of Computer Vision*, vol. 85, no. 2, pp. 167–181, 2009.

[30] P. Soille, *Morphological Image Analysis: Principles and Applications*, Springer Science & Business Media, Berlin, Germany, 2013.

[31] P. A. Flores-Vidal, P. Olaso, D. Gómez, and C. Guada, "A new edge detection method based on global evaluation using fuzzy clustering," *Soft Computing*, vol. 23, no. 6, pp. 1–13, 2018.

[32] E. B. Goldstein, "Sensación Y Percepción," Thomson Editors, Madrid, Spain, 2009.

[33] M. Efroymson, "Multiple regression analysis," *Mathematical Methods for Digital Computers*, vol. 3, pp. 191–203, 1960.

[34] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[35] B. De Baets and C. López-Molina, *The Kermit Image Toolkit (Kitt)*, Ghent university, Ghent, Belgium, 2016, https://www.kermitimagetoolkit.net.

[36] J. Kivinen, C. Williams, and N. Heess, "Visual boundary prediction: a deep neural prediction network and quality dissection," in *Proceedings of the Artificial Intelligence and Statistics*, pp. 512–521, PMLR, Reykjavik, Icelan, April 2014.

[37] Berkeley Computer Vision page, "Berkeley Computer Vision page (09/23/2021). Berkeley Segmentation Data Set (BSDS500)," 2021, https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html#bsds500.


26/29, 2022, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/2022/9729343 by Spanish Cochrane National Provision (Ministerio de Sanidad), Wiley Online Library on [08/11/2024]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License