

Research Article

Pixel-Wise Classification in Hippocampus Histological Images

Alfonso Vizcaíno ¹, Hermilo Sánchez-Cruz ¹, Humberto Sossa ²,
and J. Luis Quintanar ³

¹Departamento de Ciencias de la Computación, Universidad Autónoma de Aguascalientes, Aguascalientes 20131, Mexico

²Centro de Investigación en Computación, Instituto Politécnico Nacional, Ciudad de México, Mexico

³Departamento de Fisiología y Farmacología, Universidad Autónoma de Aguascalientes, Aguascalientes 20131, Mexico

Correspondence should be addressed to Hermilo Sánchez-Cruz; hermilo.sanchez@edu.uaa.mx

Received 10 November 2020; Accepted 11 May 2021; Published 21 May 2021

Academic Editor: Dong Song

Copyright © 2021 Alfonso Vizcaíno 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.

This paper presents a method for pixel-wise classification applied for the first time on hippocampus histological images. The goal is achieved by representing pixels in a 14-D vector, composed of grey-level information and moment invariants. Then, several popular machine learning models are used to categorize them, and multiple metrics are computed to evaluate the performance of the different models. The multilayer perceptron, random forest, support vector machine, and radial basis function networks were compared, achieving the multilayer perceptron model the highest result on accuracy metric, AUC, and F_1 score with highly satisfactory results for substituting a manual classification task, due to an expert opinion in the hippocampus histological images.

1. Introduction

The study of the hippocampus region has been of particular interest because of its relationship with memory and learning processes [1, 2], its volume as an indicator for Alzheimer's disease [3], personality disorder [4], neurological disorders derived from strokes [5–9], and drug addiction effects [10], to mention a few.

Most common hippocampal quantification techniques are based in MRI images [11, 12] for volumetric calculation and histological images [13, 14] for neural cell counting.

However, determining hippocampal volume in histological images is a challenging labour, on the one hand because image conditions are not always good and hippocampus section is of an irregular shape that is only a few pixels thick, which makes this labour an intensive and time-consuming task that demands the help of an expert to correctly identify the area of interest and so be able to determine hippocampus volume.

On one hand, pixel-wise classification has been used broadly for task such as mitosis detection in histological breast images for cancer detection [15–17], gland segmenta-

tion of prostate histology images [18], and nuclei segmentation [19], among others, where the solution range goes from digital image processing approaches used in combination with ML techniques to convolutional neural networks.

On the other hand, studies on hippocampus region have been performed exhaustively using magnetic resonance images from humans and rats, to perform segmentation tasks by applying several methods such as atlas based [20–23], a combination with support vector machines (SVMs) [24], and patch-based approaches [25], among others [26–29].

Even though each published work reports improvements over previous approaches on their own image source type and task types, as far as we know with our deep search in the literature, to perform a pixel-wise classification on hippocampus structure from rat brain histological images cuts, using the coronal anatomical plane, has not been done before. Mesejo et al. [30] perform a segmentation endeavour by using deformable models and random forest (RF) from Allen Brain Atlas [31] image repository using the sagittal anatomical plane. Senyukova et al. [32] do atlas-based segmentation on several brain sections with RF and Markov Random Fields on Nissl-stained histological sections of

mouse brains. And Riklin-Raviv et al. [33] propose a slice-by-slice segmentation with three-dimensional Gaussian mixtures and level sets where the successful segmentation of one section provides a prior for the subsequent one, assuming that the segmentation of few sparsely sampled slices is done manually.

This work reports the implementation of a computer vision method to extract pixel features and use them along with several machine learning (ML) techniques such as multilayer perceptron network (MLP), SVM, radial basis function network (RBFN), and RF, to perform pixel-wise classification on rat brain histological images using the coronal anatomical plane to correctly identify the hippocampus structure and facilitate its measurement.

2. Materials and Methods

The images used in this work are supplied by the Pharmacology Department of the Autonomous University of Aguascalientes. These images are serial coronal sections of approximately $6\ \mu$ thick, from male rat brains of the Sprague Dawley strain around 8-12 weeks old with a weight of 250-330 g. The cuts are stained with a specific chemical dye, and in this case hematoxylin-eosin is employed, to create contrast on the sought anatomical structure.

A total of 25 images were digitized from an optical microscope with a magnifying glass of 0.67x, using an LGE LM-X520 camera model that was configured with an ISO speed rating of 100, a focal length of 3.5 mm, and a variable exposure time ranging from 1/60 s to 1/30 s. Each image was captured at 4161×3120 pixels and was saved originally in JPEG format.

When performing a visual inspection on these images, several conditions can be identified:

- (i) No relevance on a specific colour for detecting anatomical structures
- (ii) Different lighting effects: some images are brighter than others
- (iii) Fuzziness of the hippocampus boundaries
- (iv) Variability of hippocampus' shape and orientation
- (v) Cutoff of regions and presence of markings such as scraps, tears, and streaks in tissue
- (vi) Rotated and uncentred brain position

Examples of some of these characteristics are shown in Figure 1.

Our methodology for hippocampus pixel classification consists of four consecutive steps, as depicted in Figure 2. The first step uses image processing to enhance hippocampus region. The second step characterizes each pixel as a 14-dimensional vector. Those vectors constitute the features used by the classification algorithms. After all images have been characterized, the third step begins. At this step, the training, validation, and test set are created. The last step involves feature classification to differentiate between a non-

hippocampus pixel and hippocampus pixel. The details of each of these steps are now described.

2.1. Image Conditioning and Preprocessing. The provided images contain ample dark areas, since brain image is swiveled and given the large image dimensions, it is necessary to condition them in order to speed up its preprocessing step. Therefore, images are straightened and only brain image is kept along with its aspect ratio. It is determined that an image size average of 1024×832 pixels is big enough to preserve hippocampus pixels and small enough to perform a fast preprocessing step. An example of the conditioning phase is displayed in Figure 3. Finally, the coordinates that make up the main hippocampus bounding box are annotated.

Because of the ample differences between hippocampus images and the image conditions explained in previous section, a preprocessing phase is performed in order to enhance hippocampus region and extract meaningful information to construct the features that will be used later in the classification step. This phase is based in the procedure employed in Vega et al. [34] and Marin et al. [35]. However, given the difference between the images and their respective domain field, the procedure has to be tailored to generate suitable images. Next, the details of the preprocessing step are described.

2.1.1. Colour Independence. Histological cuts contain different colours because of the type, amount of dye, and the exposure time given to the tissue. Consequently, pixels differ in colour and intensity despite belonging to the same hippocampus region. For this reason, the input image is converted from an original RGB colour space to a Hue Saturation Value (HSV) colour space, extracting the Value Channel (VC) to better capture the full range of the different colours that belong to the hippocampus and to make it independent from the illumination of the sample images. Then, the image is cropped to the annotated bounding box to be furthered processed. I_V denotes the resultant image for future references. Because of the nature of VC, the hippocampus region is represented by dark colour pixels which correspond to values close to zero. In order to emboss them, image is negated having I_{NV} as result.

$$I_{NV} = 255 - I_V. \quad (1)$$

Figures 4(a) and 4(d) show an example of this phase.

2.1.2. Background Homogenization. Since the background is not homogeneous, a mean filter followed by a convolution with a Gaussian kernel followed by a histogram correction operation is applied. This phase is performed in the same way that [34] does background homogenization but working with an I_{NV} image. The resulting image of this phase is denoted as I_H , and an example of the outcome is presented in Figures 4(b) and 4(e).

2.1.3. Hippocampus Enhancement. Hippocampus enhancement is performed by applying a top-hat transformation.

$$I_{HE} = \gamma(I_H), \quad (2)$$

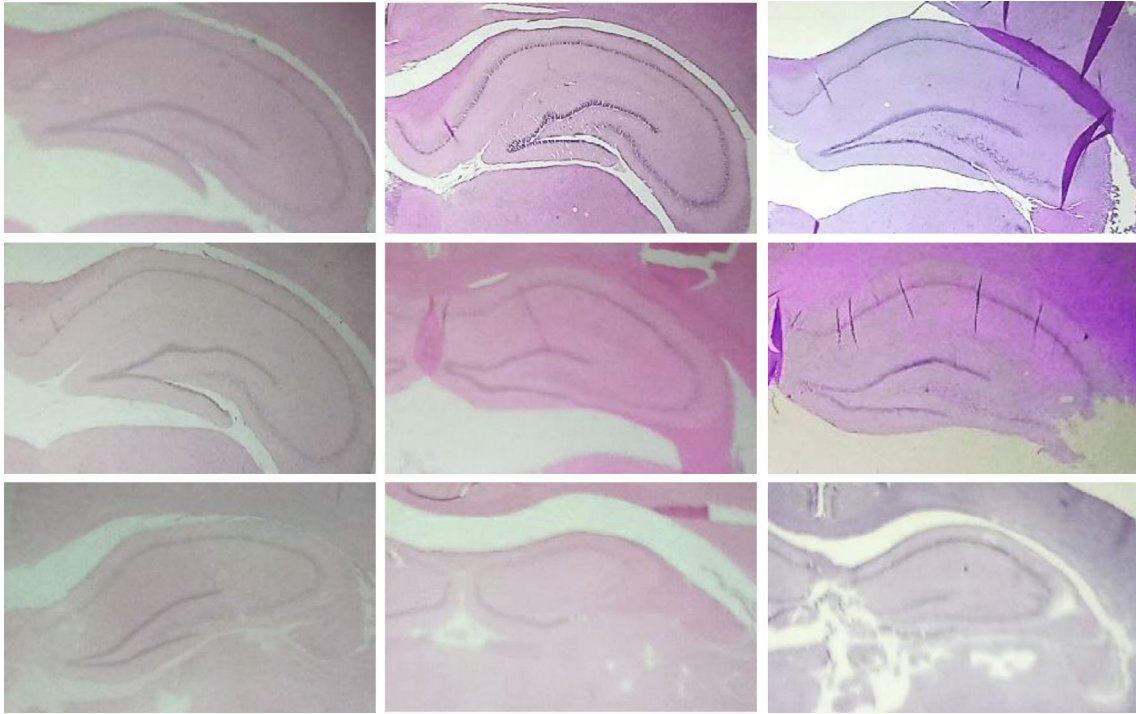


FIGURE 1: Coronal sections of the rat brain where the layers of the hippocampus stained with hematoxylin and eosin are observed. 3x magnification. Columns show difference in colour as well as in lighting. Rows show variability in hippocampus shape and markings presence.



FIGURE 2: Diagram of the implemented methodology for hippocampus pixel-wise classification.

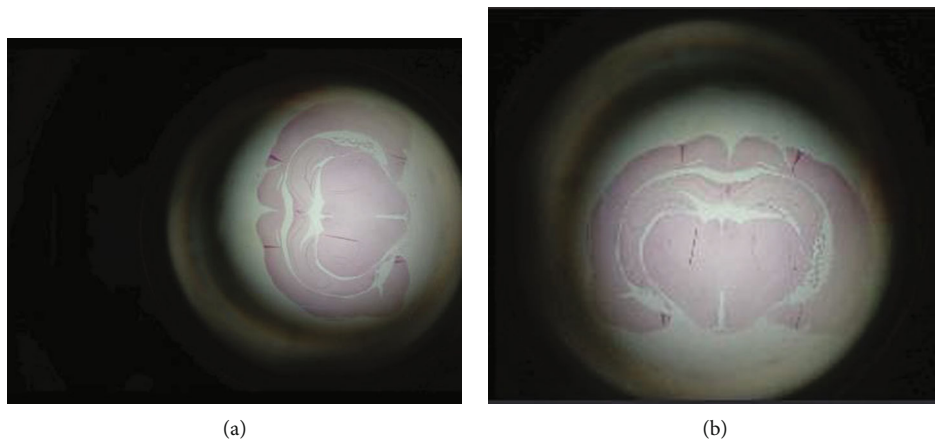


FIGURE 3: (a) Original image. (b) Final image: rotated, centred, and resized.

where γ is a morphological opening using a disc of eight pixels in radius, thus removing most anatomical structures not belonging to hippocampal region and yielding better results than performing a boundary detection with algorithms like Canny, Prewitt, and Sobel. Figures 4(c) and 4(f) exhibit an example of the procedure applied in this phase.

2.2. Feature Extraction. The purpose of this step is to perform a pixel characterization in terms of some quantifiable measurements that can be used latter in the classification step. To accomplish this task, unlike Marin et al. and Vega et al. whom used seven and five functions, respectively, in this work, a total of fourteen functions are used. Our method uses

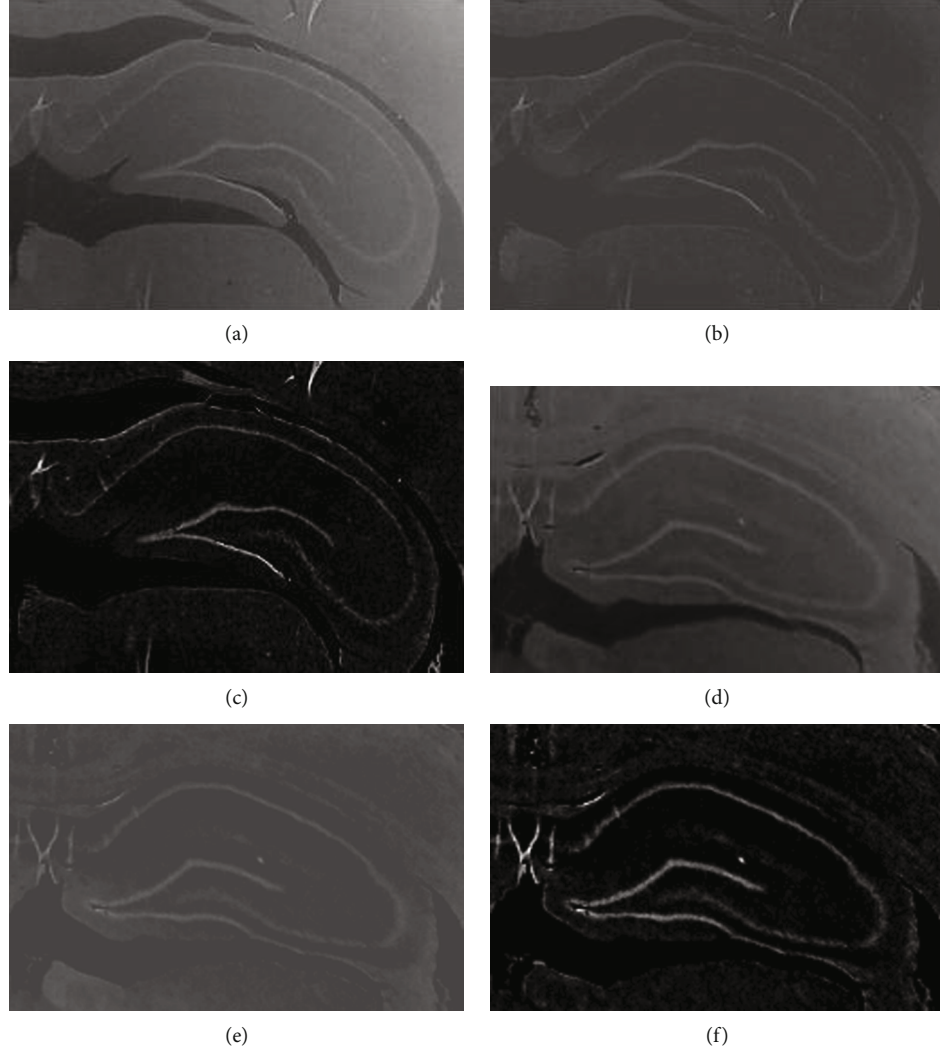


FIGURE 4: (a, d) Colour independence. (b, e) Background homogenization. (c, f) Hippocampus enhancement.

some variants that are described. The first five features are as follows: f_1, f_2, \dots, f_5 , which are based on the pixel's grey-level information available. In this work, we find that the features calculated from I_{HE} produce more meaningful representation of hippocampus pixels. The features are outlined as follows: considering a squared pixel region of size $w \times w$ taken from I_{HE} and centred at pixel (x, y) , we have the following:

f_1 is the value of the pixel being characterized at position (x, y) subtracted from the smallest value of the squared region

$$f_1(x, y) = I_{\text{HE}}(x, y) - \min_{(s, t) \in S_{x,y}^9} \{I_{\text{HE}}(s, t)\}. \quad (3)$$

f_2 is the largest value of the squared region subtracted from the value of the pixel being characterized at position (x, y)

$$f_2(x, y) = \max_{(s, t) \in S_{x,y}^9} \{I_{\text{HE}}(s, t)\} - I_{\text{HE}}(x, y). \quad (4)$$

f_3 is the value of the pixel being characterized at position (x, y) subtracted from the average value of the squared region

$$f_3(x, y) = I_{\text{HE}}(x, y) - \frac{\text{mean}}{(s, t) \in S_{x,y}^9} \{I_{\text{HE}}(s, t)\}. \quad (5)$$

f_4 is the value of standard deviation of the squared pixel region characterized at position (x, y)

$$f_4(x, y) = \frac{\text{stdDev}}{(s, t) \in S_{x,y}^9} \{I_{\text{HE}}(s, t)\}. \quad (6)$$

f_5 is the value of the pixel being characterized at position (x, y)

$$f_5(x, y) = I_{\text{HE}}(x, y). \quad (7)$$

Then, for the next two features, f_6 and f_7 , we use the

first two Hu moment invariants [36] denoted by ϕ_1 and ϕ_2 . These are computed from I_{Hu} image, which is obtained by multiplying a squared pixel region of 17×17 size from I_{HE} and an equal dimension matrix of Gaussian values, whose mean is 0 and variance is $(1.7)^2$; then, I_{Hu} is given by

$$I_{\text{Hu}}(i, j) = I_{\text{HE}}^{S_{xy}^{17}}(i, j) \times G_{0,1.7^2}^{17}(i, j). \quad (8)$$

With these choices of parameters, the 9×9 central values in Gaussian matrix contain 97% of the area of the represented Gaussian distribution, making the remainder values being close to 0. The effect of this multiplication is that the values become sensitive for describing hippocampal and nonhippocampal central pixels. Given that ϕ_1 and ϕ_2 computation can take nonpositive and zero values, f_6 is defined as

$$f_6 = \begin{cases} \log(\phi_1), & \text{if } \phi_1 > 0, \\ -\log(|\phi_1|), & \text{if } \phi_1 < 0, \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

and f_7 is defined as

$$f_7 = \begin{cases} \log(\phi_2), & \text{if } \phi_2 > 0, \\ -\log(|\phi_2|), & \text{if } \phi_2 < 0, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

Given that images contain other brain regions of similar shape and that image conditions are extremely variant, a set of extra seven features are used to acquire even more descriptive pixel information that can help better distinguishing between the sought hippocampus section and the alike structure. Figure 5 illustrates the alike structure that is also obtained as result of the preprocessing step. Table 1 shows the comparison of the performance obtained when using seven features and the increased achievement by adding the extra seven.

For the remaining seven features, f_8, f_9, \dots, f_{14} , the information is extracted from the I_{NV} image by following the same process described above, making a total of a 14-D feature vector.

Therefore, one pixel is represented by the 14-D feature space and is denoted by F

$$F = (f_1, f_2, \dots, f_{14}). \quad (11)$$

2.3. Data Set Creation. The data set, denoted by F_T , is constituted from hippocampal features F_H and nonhippocampal features F_O . F_T is distributed in the following way. First, all features F from hippocampal pixels are collected from all images, acquiring 24,520 hippocampal features. Then, to be sure to obtain a well-balanced data set, the same amount of

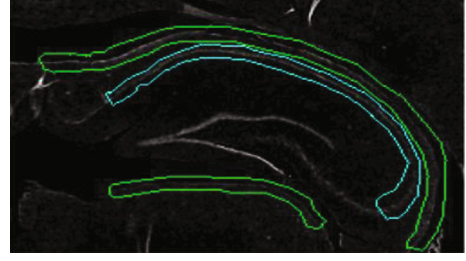


FIGURE 5: A hippocampus section is framed with cyan colour. Other brain regions that have a similar shape and similar pixel characteristics are framed with green colour.

TABLE 1: Model comparison using 7 vs. 14 features.

Model	Accuracy using 7 features	Accuracy using 14 features	Accuracy difference
MLP	0.917652	0.940481	+0.022829
RBFN	0.915206	0.931512	+0.016306
Random forest	0.915614	0.937627	+0.022013
SVM lineal	0.899715	0.918875	+0.019160
SVM poly feat	0.907868	0.931512	+0.023644
SVM poly kern	0.912556	0.935997	+0.023441
SVM RBF kern	0.917856	0.939462	+0.021606
Total average accuracy	0.912352	0.933638	+0.021285

randomly picked features is collected from nonhippocampal pixels from all images. Thus, the entire data set F_T makes a total of 49,040 features F .

$$F_T = F_H + F_O. \quad (12)$$

Next, data set is complemented with ground truth values, C_1 for hippocampal pixel and C_0 for nonhippocampal pixels. For debugging purposes, data set is augmented with C_1 and C_0 pixel coordinates along with the source image name.

Finally, the data set is randomly split into training (D_{TN}), validation (D_{VL}), and test (D_{TS}) set, distributed in 70%, 20%, and 10%, respectively, making sure the same amount of hippocampal features as well as nonhippocampal ones are contained in these data sets. A sample visual examination conducted on one of the images is shown in Figure 6.

2.4. Classification. To be able to determine if a pixel belongs to C_1 or C_0 , in this work, different ML models are employed: MLP, RBFN, SVMs, and RF.

A MLP [37] is an artificial neural network that overcomes the limitations of least mean square algorithm in solving prediction problems. MLP consists of a set of three types of nodes: input nodes, known as input layer; one or more layers of computation nodes, known as hidden layer; and an output layer. The first layer receives an input signal and propagates it through network; then, each node of the hidden layer

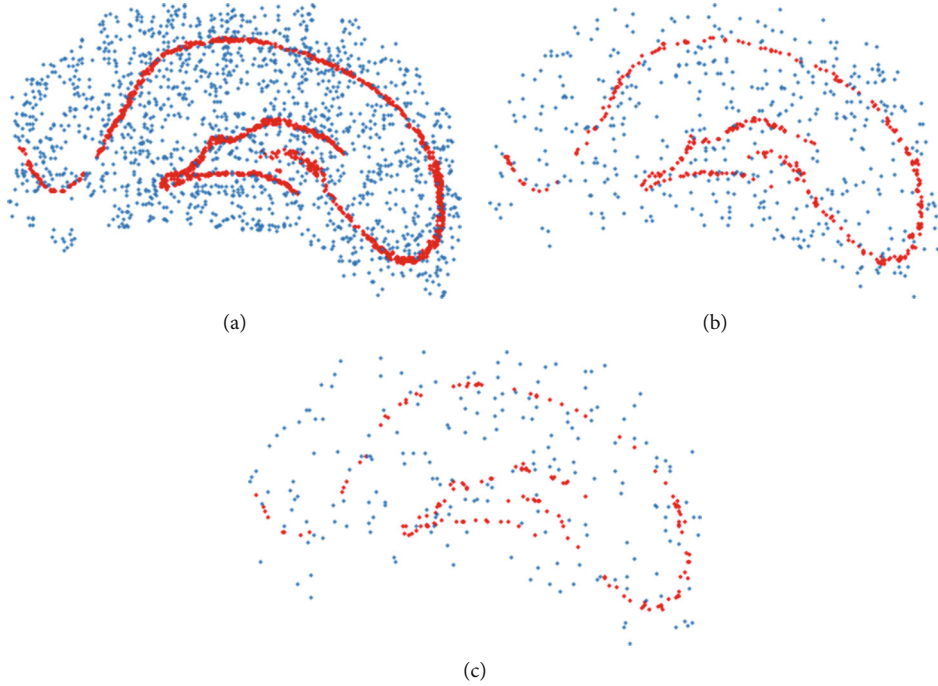


FIGURE 6: Red dots represent hippocampus pixels and blue dots represent nonhippocampus pixels. (a) Sample image from training set, (b) the same sample image taken from validation set, and (c) the same sample image taken from test set.

executes a nonlinear activation function, and the result is transmitted to the output layer where nodes located here perform a final activation function whose result is interpreted as the probability that the input signal corresponds to a known class.

On the other hand, RBFN was first proposed by Broomhead and Lowe [38] where the hidden layer is trained with an unsupervised algorithm, and the output layer is constructed with a supervised one. The key idea is to transform data points into high-dimensional space with the use of a Gaussian function, so that the transformed points become linearly separable.

SVMs [39] are a category of feed forward networks that can be used for pattern classification and nonlinear regression. SVMs construct a hyperplane as the decision surface in such a way that the margin of separation between positive and negative samples is maximized. There are three main types of SVMs: lineal, polynomials, and radial basis functions.

Random forests [40] are variants of clustering algorithms known as decision trees that perform particularly well on small data sets and like SVMs can perform both classification and regression tasks, but unlike decision trees, RF can limit the sensitivity to small variations in the training data by averaging predictions over many trees.

So that models can be more efficient, techniques such as feature engineering [41] can be used, but looking for maximizing their performance, all D_{TN} features, f_i of F , are used and standardized making them zero mean and unit variance in the following way.

$$f_i = \frac{f_i - \mu_i}{\sigma_i}, \quad (13)$$

where μ_i is the average and σ_i is the standard deviation of the i -th feature. For all these models, only f_i of F features are fed to the models, isolating data that is used for debugging purposes.

In MLP, a supervised learning algorithm known as back-propagation is used for training the layers and the synaptic weight between nodes [37]. With the right choice of weights and with the right number of the hidden nodes, MLP can be used to address classification problems [42]. Hence, the function approximation for classification is defined by a nested set of weighted summations.

RBFN solves the classification problem by proceeding in a hybrid manner. First, an input layer is composed with the same number nodes of the features being evaluated. Then, a hidden layer transforms the given set of nonlinearly separable patterns by applying an unsupervised learning algorithm. Finally, RBFN uses least squares estimation to train the output layer in a supervised manner to solve the classification problem. In RBFN, the function approximation for classification is defined by a single weighted sum [37].

RF is settled on decision trees. A decision tree is a machine learning technique, based on the divide and conquer paradigm where the basic idea is to partition the feature space into patches and to fit a model to a patch. RF creates different trees over the same training data set but provides random subset of features to each of the trees in its building process [43] and uses some aggregation technique, such as majority voting to perform the final classification.

SVMs are a type of binary classifiers that construct a hyperplane as the decision boundary and seek to maximize the distance of positive and negative examples given in a

training set [37]. SVMs use a two-step process on nonlinearly separable data to find the decision boundary. In the first step, a nonlinear transformation is applied to the data; in the second step, the points that constitute the decision boundary are then determined in the transformed space [44].

There are algorithms used in ML that have been proven to maximize predictive output such as ensemble learning [45]; however, this work is constrained to the mentioned ML models with the purpose of evaluating the pixel characterization method itself.

3. Experiments and Results

To measure the algorithmic performance of the proposed method, the before mentioned ML models are trained with D_{TN} set, and in order to find the best performing hyperparameters for each model, Random Search and Grid Search techniques are used. The former is used to reduce the search space and the latter to pinpoint the ideal values. Finally, D_{VL} set is used to assess that the found hyperparameters produce good results and that models are not overfitted.

The final architecture of the MLP model is implemented with TensorFlow and consists of an input layer with 14 nodes and then four fully connected layers made up with 31, 68, 13, and 7 nodes, respectively, with a ReLU activation type for each one of them. Finally, an output layer consisting of 1 node with a sigmoid activation type constitutes its architecture. The model was compiled using ADAM optimizer and binary cross entropy as loss function (LF) and trained over 37 epochs.

The RBFN is built in TensorFlow with the implementation provided by Vidnerová [46]. It was trained with K-Nearest Neighbourhood (K-NN) for the unsupervised algorithm and backpropagation for the supervised. The model has 3 layers as well. The first layer contains 14 input nodes, the second layer has 71 hidden nodes, and the third layer has one output node with a sigmoid activation type. The same σ value is used across Gaussian functions, and it is calculated as follows: $\sigma = d_{\max}/\sqrt{2k}$, where d_{\max} represents the maximum distance between clusters and k is the number clusters, which in turn, match the number of nodes of the second layer. The model used a mean squared error as its LF and RMSprop algorithm for the optimizer and trained over 200 epochs.

The rest of the ML models are built with scikit-learn [47]. The best RF model is set with the following hyperparameters: a gini criterion for measuring the quality of splits; a value of $\sqrt{F_n}$ is set for the maximum features, where F_n represents the number of features; a value of 1 for the minimum samples per leaf and minimum samples for node split; and unset values for max depth, max leaf nodes, and max samples.

Finally, four different SVM models are used. The first one is configured as lineal support vector classification with these hyperparameters: a value of 182 for the regularization parameter (C), a squared hinge LF, and l2 penalty function (PF). The second model is a lineal support vector classification fed with polynomial characteristics, and its hyperparameters are set in the following manner: a value of 172 for C , a 2nd degree polynomial characteristic, and a squared hinge LF and a l2 PF. The third model is set with a

3rd degree polynomial kernel; a value of 2.1 for C ; a squared l2 PF; and a γ value of $1/(C_n\sigma^2)$, where C_n is the number of characteristic and σ^2 is the variance; finally, a value of 40 is used for the independent variable (b). The last model uses a radial basis function as kernel type, a squared l2 PF, a value of $1/(C_n\sigma^2)$ for γ parameter, and a value of 182 for C .

After finding the best hyperparameter values for each of the models, the D_{TS} set is employed to objectively compare the performance of the ML models against each other using the following metrics.

On one hand, the receiver operating characteristic (ROC) curve is used to compare the performance of the classification models by plotting two parameters true positive rate (TPR) and false positive rate (FPR). These metrics are defined by

$$\begin{aligned} \text{TPR} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ \text{FPR} &= \frac{\text{FP}}{\text{FP} + \text{TN}}, \end{aligned} \quad (14)$$

where TP means true positives, FN: false negatives, FP: false positives, and TN: true negatives. For this graph, the closer the line is to the upper left corner, the better the classifier is. The ROC curve is displayed in Figure 7.

On the other hand, besides the ROC curve, metrics such as accuracy, precision, recall, and F_1 score are computed with the same purpose. These are defined by

$$\begin{aligned} \text{Accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\ \text{Precision} &= \text{TPR}, \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ F_1 \text{ score} &= 2 \times \left(\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \right). \end{aligned} \quad (15)$$

To evaluate if the added extra seven features resulted in a performance gain, the same data sets D_{TN} , D_{VL} , and D_{TS} are used for all the ML models but are trained, validated, and tested with only the first seven features, respectively. Table 1 shows that when models use 14 features, all models increase their performance.

Table 2 shows all the metrics described before and the values obtained by each model when using 14 features.

In this context, accuracy is a ratio of correctly predicted observation to the total observations; it works better when there is a symmetric data set and if FP and FN have similar cost. When the cost of FP and FN negatives is different, precision and recall metrics need to be considered. The former is the ratio of correctly predicted positive observations to the total predicted positive observations; ergo, it is a good measure to use when the costs of FP are high. The latter, also known as sensitivity, is the ratio of correctly predicted positive observations to all observations in actual class; hence, recall calculates how many of the actual positives the model captures through labelling it as TP. Finally, F_1 score is the weighted average of precision and recall, and it can be selected

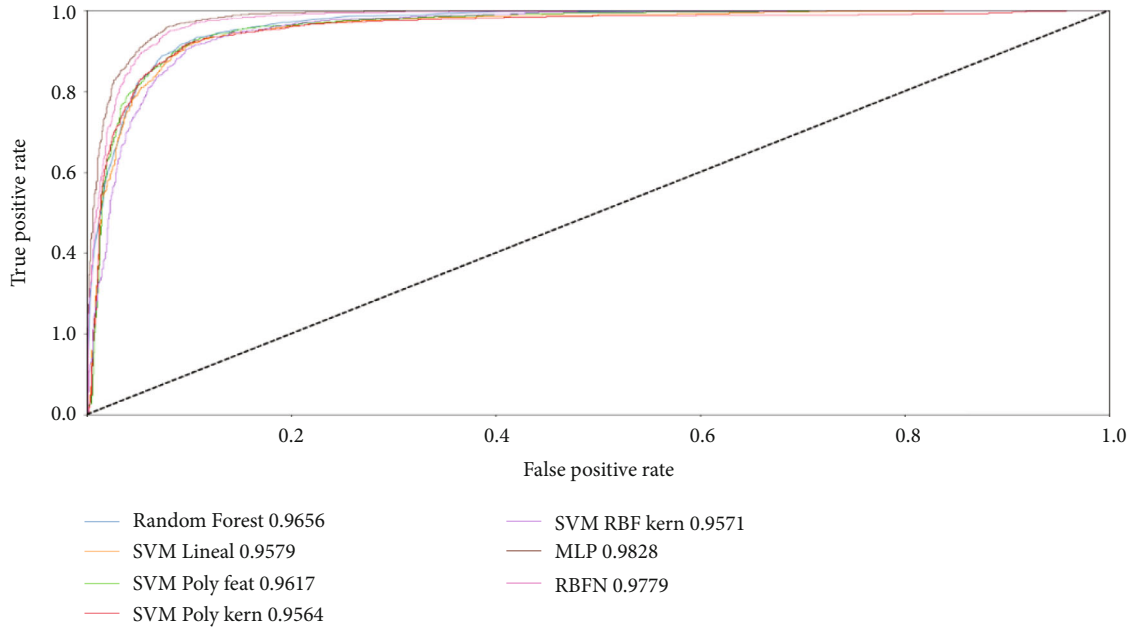


FIGURE 7: ROC curve. Dotted line from bottom left to upper right represents a strictly random classifier.

TABLE 2: Model performance.

Model	Accuracy	AUC	F_1 score	Precision	Recall
MLP	0.940481	0.982761	0.941506	0.925926	0.957620
RBFN	0.931512	0.977904	0.933413	0.908565	0.959658
Random forest	0.937627	0.965623	0.938579	0.924842	0.952730
SVM lineal	0.918875	0.957855	0.920368	0.904088	0.937245
SVM poly feat	0.931512	0.961692	0.932934	0.914319	0.952323
SVM poly kern	0.935997	0.956352	0.937325	0.918623	0.956805
SVM RBF kern	0.939462	0.957067	0.940683	0.922444	0.959658

as the main metric to use when a balance between precision and recall is required and there is an uneven class distribution.

Given that our data set is composed of symmetric quantities between C_1 and C_0 samples and assuming that FP and FN have the same cost, accuracy metric could have picked to evaluate models at one sight. Consequently, the model that presents a better performance is MLP; furthermore, this model also scored the highest on AUC, precision, and F_1 score metrics. However, knowing that all these metrics evaluate models from different perspectives and that the context, in a given problem, plays an important part on deciding which model fits a better solution, the interpretation of the other metrics is important when determining the better model. For this reason, if there was a high cost associated with a FN, i.e., a model predicting that a pixel is C_0 when in fact it is C_1 , then recall metric should be the main evaluation metric and for such scenario, RBFN would be the best performing model.

4. Discussion

Although good results were obtained with a 7-D and a 5-D feature vectors in the experiments by Marine et al. and Vega

et al., respectively, in our experiments, using a 7-D vector did not provide good enough accuracy. This could be, unlike the retinal images that were used in their work, due to the highly variability of the characteristics of the images and the presence of other brain structures that are similar to the seeked hippocampal shape. Nonetheless, when F was increased to 14-D, the experiments yielded an average accuracy increase of 2.1285%. This performance increase is associated to the more complete feature set that was generated with our proposed method and the better pixel characterization.

The best ML models are described next. For our data set, MLP model achieves the highest value in the accuracy metric and correctly predicts C_1 and C_0 pixels 94.0481% of the time. In contrast, when true positive rate needs to be considered cautiously, only 92.5926% C_1 pixels are correctly predicted and when FN has a greater importance in predicting C_1 pixels, MLP achieves a score of 95.7620%. In this regard, RBFN model achieves the best rate with a 95.9658% value. Finally, for properly predicting C_1 and when an equilibrium between FP and FN is sought, the F_1 score metric is picked. In this regard, the best balance is achieved again by MLP model, with a value of 94.1506%.

Despite that techniques such as deep learning can produce models that achieve higher performance on pixel-wise classification tasks [17, 48], and the architecture for these models tends to be of a much larger size, requiring a vast amount of information and special hardware, as GPUs, to account for the complexity and the computations needed in the training and inference phase execution, in a reasonable amount of time. In this regard, the model studied here already achieves and exceeds the accuracy needed by the subject matter expert to locate the hippocampus and can be probably increased by using ensemble methods [45] and feature engineering techniques [41]. The simplicity of the model means it can do, in a timely manner, fast inferences without the need of special hardware, and for its small size, it can even be implemented in portable devices, such as cell phones or tablets.

Even though this paper is for pixel-wise classification, one could argue that having labelled and located the position of hippocampal pixels, it can be considered as segmentation task. Also, knowing that the work from Mesejo et al. [30] is about hippocampus segmentation and that of Senyukova et al. [32] is for brain structure segmentation, with the hippocampus among them, and that both of them use histological images, we could present a second table to easily compare the results. However, Mesejo et al. use a different definition of accuracy of metric that only takes TP % values; Senyukova et al. present just the result of precision metric without giving further information. For these reasons, a fair comparison table cannot be elaborated but their results are written here for the reader's convenience. Mesejo et al. achieve an accuracy of 92.11% and Senyukova et al. accomplish a precision of 60.7194% for hippocampus structure.

5. Conclusions

We showed the robustness of the proposed method by evaluating it with different ML models. Furthermore, by adding samples of every image in the data set, we are increasing the exposure to varying styles of histological images. Likewise, by splitting our data set in 70% for training, 20% for validating, and 10% for testing, we can objectively verify that ML models are not overfitting. Hence, we can conclude that the method will be able to generalize when new images are to be presented.

Not only this, the proposed method described in this work showed that it is possible to do pixel-wise classification for histological images and achieve a remarkable good performance too. Furthermore, the described method can significantly reduce the lengthy effort employed by the subject matter expert on identifying and delimiting the hippocampal region and be considered an adequate operation for substituting a manual classification task.

However, there are some aspects that could be considered for future work. One likely way that performance could be improved is by using an ensemble method algorithm. The shown advantage obtained through the use of a 14-D feature vector could be made more efficient by using PCA or a feature engineering technique that reduces the dimensionality of the feature vector and still obtains a good performance.

The current method could also be leveraged by fully automating the pixel-wise classification task by integrating an algorithm that locates the hippocampus bounding box. Another direction for future research could be on the generation of a hippocampus segmentation method to further facilitate the measurement of this area and help automating the quantification of hippocampal volume.

Data Availability

Source images, image pixel labels, data set, and code can be found at the following repository: https://github.com/alfonso-vizcaino/hipp_pixel_classification.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is funded in part by the Universidad Autónoma de Aguascalientes under grant PIINF20-6. A. Vizcaino thanks CONACYT for the scholarship 2020-000013-01NACF-15018 granted to pursue his doctoral studies. Humberto Sossa thanks the Instituto Politécnico Nacional for the support to undertake this research. We also recognize Ma del Rosario Montoya García for her scientific discussions.

References

- [1] K. A. Norman, "How hippocampus and cortex contribute to recognition memory: revisiting the complementary learning systems model," *Hippocampus*, vol. 20, no. 11, pp. 1217–1227, 2010.
- [2] S. A. Golsorkhdan, M. E. Boroujeni, A. Aliaghaei et al., "Methamphetamine administration impairs behavior, memory and underlying signaling pathways in the hippocampus," *Behavioural Brain Research*, vol. 379, article 112300, 2020.
- [3] J. Barnes, J. W. Bartlett, L. A. van de Pol et al., "A meta-analysis of hippocampal atrophy rates in Alzheimer's disease," *Neurobiology of Aging*, vol. 30, no. 11, pp. 1711–1723, 2009.
- [4] S. Kaya, H. Yildirim, and M. Atmaca, "Reduced hippocampus and amygdala volumes in antisocial personality disorder," *Journal of Clinical Neuroscience*, vol. 75, pp. 199–203, 2020.
- [5] V. Bouët, T. Freret, J. Toutain, D. Divoux, M. Boulouard, and P. Schumann-Bard, "Sensorimotor and cognitive deficits after transient middle cerebral artery occlusion in the mouse," *Experimental Neurology*, vol. 203, no. 2, pp. 555–567, 2007.
- [6] R. Gerlai, H. Thibodeaux, J. T. Palmer, M. van Lookeren Campagne, and N. van Bruggen, "Transient focal cerebral ischemia induces sensorimotor deficits in mice," *Behavior Brain Research*, vol. 108, no. 1, pp. 63–71, 2000.
- [7] S. Knecht, S. Hesse, and P. Oster, "Rehabilitation after stroke," *Deutsches Aertzblatt Online*, vol. 108, no. 6, pp. 600–606, 2011.
- [8] R. P. Allred and T. A. Jones, "Maladaptive effects of learning with the less affected forelimb after focal cortical infarcts in rats," *Experimental Neurology*, vol. 210, no. 1, pp. 172–181, 2008.

- [9] J. Yuan, "Neuroprotective strategies targeting apoptotic and necrotic cell death for stroke," *Apoptosis*, vol. 14, no. 4, pp. 469–477, 2009.
- [10] Y. Li, X. Wang, N. Li et al., "The NAc lesions disrupted the hippocampus-mPFC theta coherence during intravenous cocaine administration in rats," *Neuroscience Letters*, vol. 729, p. 134986, 2020.
- [11] W. Hu, L. Liu, B. Zhao et al., "Use of an automated quantitative analysis of hippocampal volume, signal, and glucose metabolism to detect hippocampal sclerosis," *Frontiers in Neurology*, vol. 9, p. 820, 2018.
- [12] A. C. Coan, B. Kubota, F. P. G. Bergo, B. M. Campos, and F. Cendes, "3T MRI quantification of hippocampal volume and signal in mesial temporal lobe epilepsy improves detection of hippocampal sclerosis," *American Journal of Neuroradiology*, vol. 35, no. 1, pp. 77–83, 2014.
- [13] J. Wang, A. Jahn-Eimermacher, M. Brückner, C. Werner, K. Engelhard, and S. C. Thal, "Comparison of different quantification methods to determine hippocampal damage after cerebral ischemia," *Journal of Neuroscience Methods*, vol. 240, pp. 67–76, 2015.
- [14] T. A. Steve, J. D. Jirsch, and D. W. Gross, "Quantification of subfield pathology in hippocampal sclerosis: a systematic review and meta-analysis," *Epilepsy Research*, vol. 108, no. 8, pp. 1279–1285, 2014.
- [15] A. Tashk, M. S. Helfroush, H. Danyali, and M. Akbarzadeh, "An automatic mitosis detection method for breast cancer histopathology slide images based on objective and pixel-wise textural features classification," in *The 5th Conference on Information and Knowledge Technology*, pp. 406–410, Shiraz, 2013.
- [16] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2013*, Springer, 2013.
- [17] D. Cai, X. Sun, N. Zhou, X. Han, and J. Yao, "Efficient mitosis detection in breast cancer histology images by RCNN," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 919–922, Venice, Italy, 2019.
- [18] S. Naik, S. Doyle, M. Feldman, J. Tomaszewski, and A. Madabhushi, *Gland segmentation and computerized Gleason grading of prostate histology by integrating low-high-level and domain specific information*, MIAAB, 2007.
- [19] Z. Zeng, W. Xie, Y. Zhang, and Y. Lu, "RIC-Unet: an improved neural network based on Unet for nuclei segmentation in histology images," *IEEE Access*, vol. 7, pp. 21420–21428, 2019.
- [20] O. T. Carmichael, H. A. Aizenstein, S. W. Davis et al., "Atlas-based hippocampus segmentation in Alzheimer's disease and mild cognitive impairment," *NeuroImage*, vol. 27, no. 4, pp. 979–990, 2005.
- [21] F. Van der Lijn, T. den Heijer, M. M. B. Breteler, and W. J. Niessen, "Hippocampus segmentation in MR images using atlas registration, voxel classification, and graph cuts," *NeuroImage*, vol. 43, no. 4, pp. 708–720, 2008.
- [22] A. R. Khan, N. Cherbuin, W. Wen, K. J. Anstey, P. Sachdev, and M. F. Beg, "Optimal weights for local multi-atlas fusion using supervised learning and dynamic information (Super Dyn): validation on hippocampus segmentation," *NeuroImage*, vol. 56, no. 1, pp. 126–139, 2011.
- [23] T. A. Steve, C. L. Yasuda, R. Coras et al., "Development of a histologically validated segmentation protocol for the hippocampal body," *NeuroImage*, vol. 157, pp. 219–232, 2017.
- [24] Y. Hao, T. Wang, X. Zhang et al., "Local label learning (LLL) for subcortical structure segmentation: application to hippocampus segmentation," *Human Brain Mapping*, vol. 35, no. 6, pp. 2674–2697, 2014.
- [25] P. Coupé, J. V. Manjón, V. Fonov, J. Pruessner, M. Robles, and D. L. Collins, "Nonlocal patch-based label fusion for hippocampus segmentation," *Lecture Notes in Computer Science*, vol. 6363, pp. 129–136, 2010.
- [26] K. Richards, C. Watson, R. F. Buckley et al., "Segmentation of the mouse hippocampal formation in magnetic resonance images," *NeuroImage*, vol. 58, no. 3, pp. 732–740, 2011.
- [27] M. Chupin, E. Gérardin, R. Cuingnet et al., "Fully automatic hippocampus segmentation and classification in Alzheimer's disease and mild cognitive impairment applied on data from ADNI," *Hippocampus*, vol. 19, no. 6, pp. 579–587, 2009.
- [28] V. Dill, A. R. Franco, and M. S. Pinho, "Automated methods for hippocampus segmentation: the evolution and a review of the state of the art," *Neuroinformatics*, vol. 13, no. 2, pp. 133–150, 2015.
- [29] N. Khalili, N. Lessmann, E. Turk et al., "Automatic brain tissue segmentation in fetal MRI using convolutional neural networks," *Magnetic Resonance Imaging*, vol. 64, pp. 77–89, 2019.
- [30] P. Mesejo, R. Ugolotti, S. Cagnoni, F. Di Cunto, and M. Giacobini, "Automatic segmentation of hippocampus in histological images of mouse brains using deformable models and random forest," in *2012 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS)*, pp. 1–4, Rome, 2012.
- [31] Allen Institute for Brain Science, "Allen reference atlases," <http://mouse.brain-map.org>, 2004–2006.
- [32] O. V. Senyukova, A. S. Lukin, and D. P. Vetrov, "Automated atlas-based segmentation of Nissl-stained mouse brain sections using supervised learning," *Programming and Computing Software*, vol. 37, no. 5, pp. 245–251, 2011.
- [33] T. Riklin-Raviv, N. Sochen, N. Kiryati et al., "Propagating distributions for segmentation of brain atlas," in *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1304–1307, Arlington, VA, USA, 2007.
- [34] R. Vega, G. Sanchez-Ante, L. E. Falcon-Morales, H. Sossa, and E. Guevara, "Retinal vessel extraction using lattice neural networks," *Computers in Biology and Medicine*, vol. 58, pp. 20–30, 2015.
- [35] D. Marín, A. Aquino, M. Gegundez-Arias, and J. Bravo, "A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features," *IEEE Transactions on Medical Imaging*, vol. 30, no. 1, pp. 146–158, 2011.
- [36] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, pp. 179–187, 1962.
- [37] S. Haykin, *Neural Networks and Learning Machines, Third Edition*, Prentice Hall, 2008.
- [38] D. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.
- [39] C. C. Aggarwal, *Neural Networks and Deep Learning. A Textbook*, Springer, 2018.
- [40] L. Breiman, "Random forests," *Journal Machine Learning archive*, vol. 45, no. 1, pp. 5–32, 2001.
- [41] A. Onan, S. Korukoğlu, and H. Bulut, "A hybrid ensemble pruning approach based on consensus clustering and multi-

- objective evolutionary algorithm for sentiment classification,” *Information Processing & Management*, vol. 53, no. 4, pp. 814–833, 2017.
- [42] M. Kubat, *An Introduction to Machine Learning, Second Edition*, Springer, 2017.
- [43] L. Igual and S. Seguí, *Introduction to Data Science a Python Approach to Concepts Techniques and Applications, Second Edition*, Springer, 2017.
- [44] W. Ertel, *Introduction to Artificial Intelligence, Second Edition*, Springer, 2017.
- [45] A. Onan, “On the performance of ensemble learning for automated diagnosis of breast cancer,” *Artificial Intelligence Perspectives and Applications*, vol. 347, pp. 119–129, 2015.
- [46] P. Vidnerová, *RBF-Keras: An RBF Layer for Keras Library*, 2019, https://github.com/PetraVidnerova/rbf_keras.
- [47] F. Pedregosa, G. Varoquaux, A. Gramfort et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [48] W. Ouyang, A. Aristov, M. Lelek, X. Hao, and C. Zimmer, “Deep learning massively accelerates super-resolution localization microscopy,” *Nature Biotechnology*, vol. 36, no. 5, pp. 460–468, 2018.