


Article

Hyperparameter Optimization for Image Recognition over an AR-Sandbox Based on Convolutional Neural Networks Applying a Previous Phase of Segmentation by Color–Space

Andrés Ovidio Restrepo Rodríguez ¹, Daniel Esteban Casas Mateus ¹,
Paulo Alonso Gaona García ¹, Carlos Enrique Montenegro Marín ¹ and
Rubén González Crespo ^{2,*} 

¹ Faculty of Engineering, Universidad Distrital Francisco José de Caldas, Cra. 7 #40b-53, Bogotá, Colombia; aorestrep@correo.udistrital.edu.co (A.O.R.R.); decasasm@correo.udistrital.edu.co (D.E.C.M.); pagaonag@udistrital.edu.co (P.A.G.G.); cemontenegrom@udistrital.edu.co (C.E.M.M.)

² Department of Computer Science, Universidad Internacional de La Rioja, Logroño, 26006 La Rioja, Spain

* Correspondence: ruben.gonzalez@unir.net

Received: 5 November 2018; Accepted: 6 December 2018; Published: 11 December 2018



Abstract: Immersive techniques such as augmented reality through devices such as the AR-Sandbox and deep learning through convolutional neural networks (CNN) provide an environment that is potentially applicable for motor rehabilitation and early education. However, given the orientation towards the creation of topographic models and the form of representation of the AR-Sandbox, the classification of images is complicated by the amount of noise that is generated in each capture. For this reason, this research has the purpose of establishing a model of a CNN for the classification of geometric figures by optimizing hyperparameters using Random Search, evaluating the impact of the implementation of a previous phase of color–space segmentation to a set of tests captured from the AR-Sandbox, and evaluating this type of segmentation using similarity indexes such as Jaccard and Sørensen–Dice. The aim of the proposed scheme is to improve the identification and extraction of characteristics of the geometric figures. Using the proposed method, an average decrease of 39.45% to a function of loss and an increase of 14.83% on average in the percentage of correct answers is presented, concluding that the selected CNN model increased its performance by applying color–space segmentation in a phase that was prior to the prediction, given the nature of multiple pigmentation of the AR-Sandbox.

Keywords: image acquisition; image processing; image recognition; convolutional neural network; dataset; loss function; accuracy; ROC curve; AR-Sandbox; random search

1. Introduction

Artificial intelligence seeks to resemble the capabilities of human beings represented in machines, and it is involved in human fields such as learning, reasoning, adaptation, and self-correction [1]. Within this field, neural networks and image processing are working together in order to generate an accurate classification model, improving learning from the extraction of characteristics and patterns. In addition, the combination of these two fields can be seen applied in regenerative medicine, microbiology, hematology, precision agriculture, and tumor identification, among others.

In Reference [2], the authors obtained an improvement in the classification and learning speed of a model that adapts to the characteristics of the input data. The implementation of a convolutional neural network with image processing was performed, in which the results were produced similarly

for different types of files and sizes, making use of semantic segmentation by the means of an FCN (fully convolutional network), in which the delimitation and separation of objects was carried out. In Reference [2], the authors had difficulty in improving the resolution of high-quality images and videos from making use of convolutional neural networks; by analyzing each of the contained pixels, it was demonstrated that the implementation of a convolutional layer at the pixel level improved the quality of the images and videos without the need for a great expense of computational resources. This is why the combination of a convolutional neural network (CNN) with image segmentation methods offers the opportunity to optimize the results.

The use of an AR-Sandbox [3,4] is a technique that allows us to close the gap between two-dimensional (2D) and three-dimensional (3D) visualization when projecting a digital topographic map on a landscape of isolated space, improving spatial thinking and modeling skills of people, with the purpose of placing it within the use of early childhood education and rehabilitation through motor therapy, which will be expanded upon in the Image Acquisition Section. For these reasons, this study's motivation is to create contributions towards image recognition based in other fields, such as immersive techniques, through deep learning by the means of convolutional neural networks, making use of hyperparameter optimization and image processing.

This article presents the preliminary results of a prediction model implementation that is based on convolutional neural networks for the classification of geometrical figures, contemplating previous steps such as the acquisition of images using the AR-Sandbox augmented reality device, and the processing of these images through segmentation by color-space. The purpose of applying this type of segmentation to the images acquired with the AR-Sandbox is to improve the model performance of the selected convolutional neural network, improving the extraction and identification of characteristics of the geometric figures in the prediction phase, and preserving a high percentage of similarity between the original image and the segmented image. To ensure this aspect, the efficiency of the segmentation method is evaluated by using the similarity coefficients of Jaccard and Sørensen-Dice [5], which will be discussed later, in order to integrate the model with the AR-Sandbox for future applications in rehabilitation and education. The prediction model is made using the Keras neural network library under the TensorFlow framework and the Talos library in order to carry out the configuration, performance, and evaluation of hyperparameter optimization by implementing a Random Search algorithm, through an implementation made in Python. Additionally, the Talos library works with any Keras model [5].

The rest of the article is organized as follows. Section 2 consists of the background, where works related to the topics to be developed are discussed. Next, Section 3 presents a macro-scenario that is subdivided into three components: approach, development, and testing, presenting the characteristics and the implementation of each of these components. In the scenario test component, two datasets with different characteristics are used in order to perform a comparative analysis and to establish the results. Finally, the conclusion and future works are presented.

2. Background

Image recognition through convolutional neural networks has had a large number of applications, together with the previous image processing, in order to optimize pattern recognition [6]. Convolutional neural networks have had continuous improvements since their creation, through the innovation of new layers and making use of different computer vision techniques [7]. In the regenerative medicine field, a study carried out by Reference [8] has developed automatic cell culture systems, where a CNN was implemented as a deep learning method to automate the recognition of cellular differences by means of the contrast in the different images. On the other hand, in China, a study was carried out where white blood cell segmentation was implemented, which proposed a method for segmentation based on color-space, making a color adjustment before segmentation, where an accuracy of 95.7% and an overall accuracy of 91.3% were achieved for the segmentation of the nucleus and segmentation of the cytoplasm [9].

On the other hand, based on studies carried out by Reference [10], randomized trials are more efficient for the optimization of multiple parameters than tests in a grid; for this reason, the Random Search method was selected for the optimization of CNN hyperparameters. With this method, four tasks are performed: (1) the use of the same model with different initial parameters; (2) the best models discovered through cross-validation are taken; (3) different control points for each of the models are identified, and finally, (4) execution of the average of the parameters in the training stage.

For the implementation of Random Search in Python, the Talos library was used, which makes use of optimization algorithms such as Random Search, Grid Search, and correlation-based optimization. Talos has a POD strategy: Prepare, Optimize, and Deploy; this task is automated, and it produces results for prediction problems [5].

In addition, in a study carried out by Reference [11], they proposed a convolutional neuronal network for the improvement of thermal images, incorporating the domain of brightness with a residual learning technique, increasing the performance and speed of convergence. The fast development of precision agriculture has generated the need for agriculture production management and estimation through the classification of crops through satellite images, but due to the complexity and fragmentation of the characteristics, traditional methods have not been able to fulfill the standards of agricultural problems. For this reason, in Reference [12], a classification method of agricultural remote sensing images was proposed based on convolutional neural networks, where the correct classification rate obtained was 99.55%. According to the above, neural networks, together with image processing, are a commonly used alternative for image classification [13].

Additionally, statistical methods have been identified to verify the efficiency of the segmentation method. For this case, the Jaccard coefficient and the Sørensen-Dice coefficient were used, which make comparisons of images (original and with image segmentation) through the bitmaps of each one:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad (1)$$

with A and B being the bitmaps of the selected images. The coefficient takes the intersection between A and B , which will be the points in common, and divides the union between A and B , this being the totality of the two images without repetition of data. This throws a value between 0 and 1 that is known as the Jaccard coefficient [14], as shown in Equation (1).

$$Dice(A, B) = \frac{2 |A \cap B|}{|A| + |B|}, \quad (2)$$

with A and B being the bitmaps of the selected images, we take the rule of the intersection between A and B , which will be the points in common, multiplied by two; this is divided between the sum of the standard of A and the norm of B , which yields a value between 0 and 1, which is known as the Sørensen-Dice coefficient [15], as shown in Equation (2). These coefficients are based on the Kappa coefficient, which is another statistical method that provides a probability of success [16]. A good result of these coefficients is a number that is greater than 0.70, and this means that the segmentation method used supports image processing for prediction making.

In other methods, Reference [17] uses image analysis and fractal dimensions to detect tumors in computed tomography (CT) scan images with high contrast, where the image preprocessing contrast of the cut images was improved by converting the values in the image intensity, using the histogram equalization to increase the accuracy of the tumor diagnosis. The image noise was reduced by the use of median filtering; finally, border detection was carried out by using the border function developed in MATLAB. This method has better performance and provides more acceptable responses than statistical algorithms. In Reference [18], the authors describe the development and implementation of feature selection for content-based image retrieval (CBIR) through a system that automatically extracts features from images using color, texture, and shape in order to use feature selection by means of a genetic algorithm that searches for the best feature-use feature selection. The results of this study conclude that

the CBIR system is more efficient, and that it performs better when using feature selection based on a genetic algorithm, because it reduces the time for retrieval and also increases the retrieval precision.

On the other hand, the authors in Reference [19] presented a study that proposed a 14-layer convolutional neural network, combined with three advanced techniques: batch normalization, dropout, and stochastic pooling in order to carry out multiple sclerosis identification. The results of this study concluded that with this model they obtained an accuracy of $98.77 \pm 0.39\%$. Results were compared with CNN when using maximum pooling and average pooling; the comparison showed that stochastic pooling gave a better performance than the other two pooling methods.

Furthermore, in Reference [20], the researchers proposed to reconstruct objects based on incomplete images and some information on the 3D object using active and passive methods to reconstruct high-resolution items. Additionally, mathematics is essential for image processing; in Reference [21], the researchers used an arithmetic method to find the right and more trustworthy solution for quantization tables, which are tables for image compression. In a similar field, the use of statistical analysis for algorithm optimization is a good way to optimize models; in Reference [22], this type of analysis was used to enhance a cognitive model for a particle swarm based on vorticity, or the tendency of something to rotate.

On basis of this panorama, the present study intends to carry out the recognition of images by the means of deep learning techniques, such as convolutional neural networks and image processing by color-space segmentation, with the purpose of determining the performance variation of a convolutional neural network, applying color-space segmentation to one of the test datasets. In the next section, the selected study scenario is presented.

3. Materials and Methods

3.1. Method and Approach of the Scenario

Figure 1 shows the general approach of the scenario to be worked on, which is composed of three main components, such as (1) image acquisition, (2) image processing, and (3) image recognition. Each one makes use of different methods and tools in order to develop each general component. In image acquisition, augmented reality is implemented by the means of the AR-Sandbox device, which performs real-time projections of a color elevation map, capturing depths through the depth camera, which is connected to a first-generation 3D Kinect, in addition to making use of a standard projector [23]. On the other hand, image processing is based on color-space segmentation, through first applying saturation to the image, using Python and OpenCV. Finally, there is the image recognition component, where a CNN is implemented; due to the versatility that the CNN has with images, it is able to analyze them with distortions, such as different light conditions, different positions, and vertical and horizontal changes, among others. Additionally, with respect to other algorithms, the number of parameters was reduced, and therefore the training time was also reduced [20], which was the central theme of this research. It is important to highlight that the model will be obtained using the Random Search optimization algorithm. The development of the CNN was done through Keras model, since it was a high-level neural network API, written in Python and capable of running on TensorFlow, Cognitive Toolkit (CNTK), or Theano. In addition, it was developed with an approach to allow for rapid experimentation, working under the following principles: easy to use, modularity, and extensibility; these are fundamental factors for an investigation. The purpose of the scenario implementation was to predict geometric figures in different contexts.

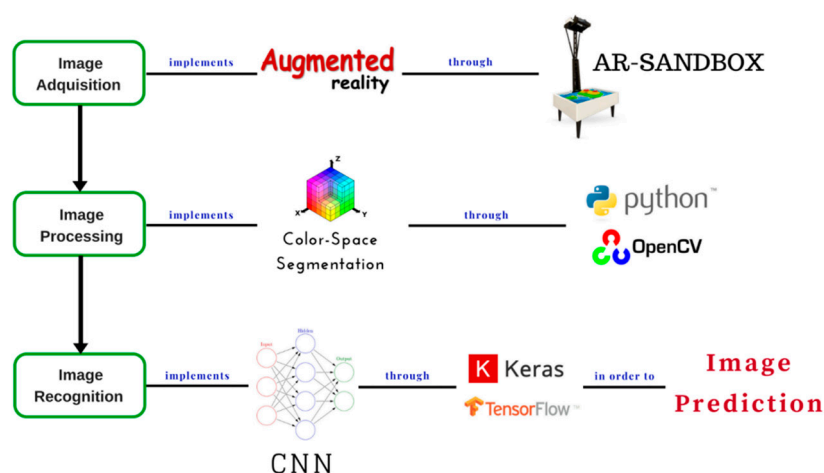


Figure 1. Scenario approach.

3.2. Development of the Scenario

3.2.1. Image Acquisition

As a first instance, the acquisition of the images was made through the AR-Sandbox augmented reality tool, which, through a projector, a Kinect, and a sandbox provided the experience to make colorful representations in the third dimension over sand. As mentioned above, this device aims to support early education and rehabilitation in specific motor therapy, which seeks to change the paradigm of traditional exercises. In order to be innovative and to encourage the patient, this is executed through the realization of exercises on the sand. After the process is carried out on the sand, the performance optimization in the classification of geometric figures is sought through the automation of this task.

For the acquisition of the image, the projection generated by the AR-Sandbox was used, which consisted of a projector and a Kinect that was located at a distance of 40 inches (102 cm) from the sand. This last device used infrared sensors and cameras in order to detect the depth of each of the points on the sand, so that the higher points were assigned a green color, and the lower points were assigned a blue color; in other words, where there were mountains inside the sand, the green color would be projected, and where there were none, the blue color would be projected.

In this way, geometric figures were made on the sand and a screenshot of the projected image was taken. In Figure 2, the result of this process is shown when making figures in the sand.

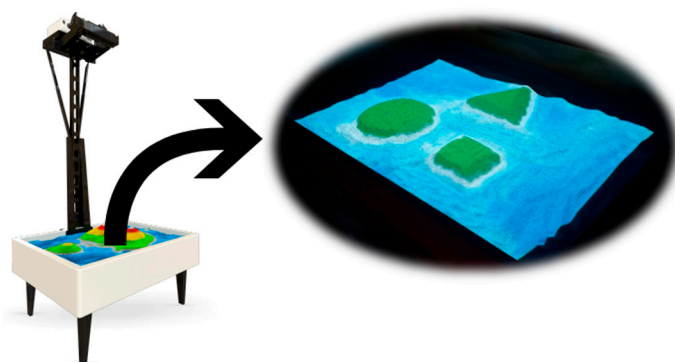


Figure 2. Projection of figures in the AR-Sandbox.

For the acquisition, a set of six people was convened, each of them made 15 images in the AR-Sandbox, distributed as circles, squares, and triangles, thus completing a total of 90 images, or 30 images of each class. Within these 90 images, two sets of test data were constructed.

3.2.2. Image Processing

At the end of the acquisition of the images, we proceeded to select the type of segmentation to be implemented, according to the characteristics of the AR-Sandbox, in which multi-pigmentation was counted according to the height of the sand. This meant that there were several colors in the sand, and it was necessary to use a method that could have a focus and objective where a specific color was selected with a color model; for this reason, segmentation by color-space was selected.

According to Reference [24], there are six options for creating image segmentation by using color-space, these are: grayscale, RGB (red, blue, green), HSV (hue, saturation, value), Opp. (opponent color), and LUV and LAB. After the results analysis, the differences between the models were small; among those differences was a high TPR (true positive rate) in the BGR model, which was better than in the others. This rate was the basis for the confusion matrix and the ROC (receiver operating characteristic) curves; for this reason, the BGR model was selected in order to obtain better performance in the prediction results.

Once the acquisition of images was completed, the image processing stage was commenced; for this purpose, the Python library OpenCV was used, which provides tools to perform segmentation by color-space. The color type and chosen space was the color model BGR (blue, green, red), since the colors of the projections were known in the mentioned scale, which is a requirement for processing. In this case, the reference color was the green equivalent in BGR to (47, 122, 16).

As a first instance, an image resizing was done, obtaining as a result a dimension of 128×128 pixels. From the target color and a range of close colors, segmentation by color-space was carried out in order to separate the colors in the image; therefore, the values within these parameters were painted white and the rest of them were black. This is called a mask, which provides a contrast in the image. Making use of the mask, the white part took the green color, while the rest of the image remained black. In order to return to the reference color range to demarcate the contour of the figure, facilitating the identification of shapes in the model of prediction that will be further explained, in Figure 3, the process that was carried out with the circle, square, and triangle is presented, where the captured image, the obtained mask, and the resulting image after having applied the filter are seen.

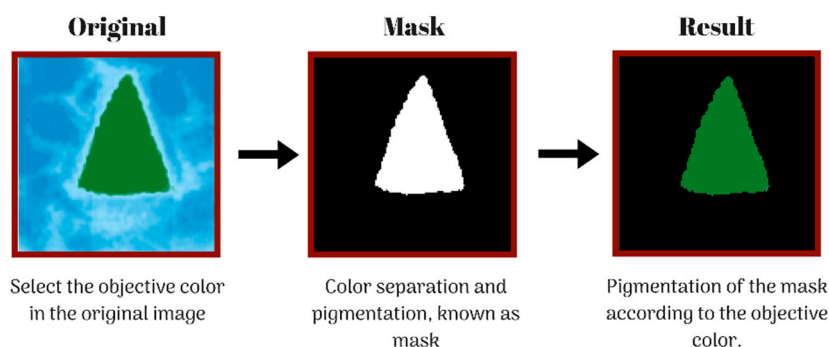


Figure 3. Segmentation process by color-space.

From the above, two sets of tests were formed. The first was `dataSetOriginal`, composed of 90 images in their original state, without applying any type of image processing algorithm. The second was `dataSetFilter`, composed of 90 images, to which the segmentation method by color-space was applied, in order to be used in the execution of the models generated by the Random Search optimization algorithm. Figure 4 presents the two test sets.

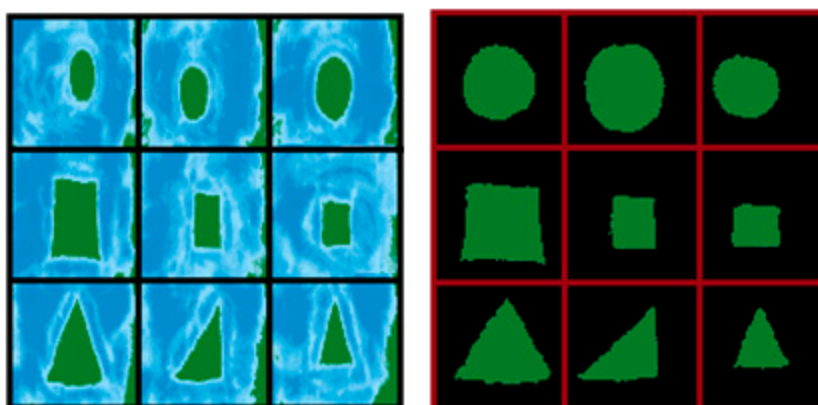


Figure 4. Utilized datasets.

Subsequently, two coefficients, Jaccard and Sørensen-Dice were used, which in different ways, provide a probability of similarity between the original image and the image after performing the process of segmentation by color-space. This task was carried out in Matlab, where the Jaccard coefficient and the Sørensen-Dice coefficient were extracted individually for each of the images. All of the results of each coefficient were then grouped, and an average probability of the set of images was given.

To carry out this process, the original image that was in grayscale was imported to perform the contour analysis with the help of the activeContour function, and then the mask obtained in the segmentation process by color-space was imported to make use of the functions Jaccard and Sørensen-Dice; therefore, for the Jaccard coefficient, $a = 0.8221$ was obtained, and for the Sørensen-Dice coefficient, $a = 0.8767$.

3.2.3. Image Recognition through CNN

In order to carry out the recognition and classification of geometric figures, the convolutional neural network model that will be used for this purpose must be selected. To carry out this task, the Random Search optimization algorithm was used, which consists of proposing a base structure of the CNN model, posing a set of hyperparameters within a list of elements, which will take place in the respective hyperparameters of CNN, and generating the random training of N models, to finally select a single model. To carry out this algorithm, the Talos library was used.

Base Structure of the CNN Model

Figure 5 presents the base structure of the convolutional neural network model, in order to test it with different hyperparameters by using the Random search algorithm, based on this, the model has a total of four general layers, where the first layer is composed of two layers of convolution, defining a kernel of (3,3), a padding of type “same”, and only waiting for the assignment of the number filters and the activation function for this first general layer. The first convolution layer has an input dimension of 128×128 pixels with its three-color channels, and a pooling layer with a pool size of (2,2), to reduce the number of parameters to keep the most common characteristics and a layer of dropout, waiting for the definition of its elimination rate, in order to avoid over-training on CNN. This composition of sublayers is repeated three times; therefore, it has three equal layers, but with different parameters. In each of these layers, the same activation function assigned randomly will be used in order to maintain homogeneity in this aspect. The objective of these three general layers is the extraction of the characteristics. Finally, there is the last general layer, consisting of a Flatten layer to convert the elements of the image matrix to a flat arrangement, followed by a Dense layer, which predicts the number of hidden layers. Afterwards, there is a layer of Dropout without the allocation of elimination rate, and ending the output layer, there is a Dense layer, where three hidden layers are used according

to the number of classes in the algorithm; the layer also has a Softmax activation function, since it is necessary to make a representation of the categorical distribution in order to generate the classification.

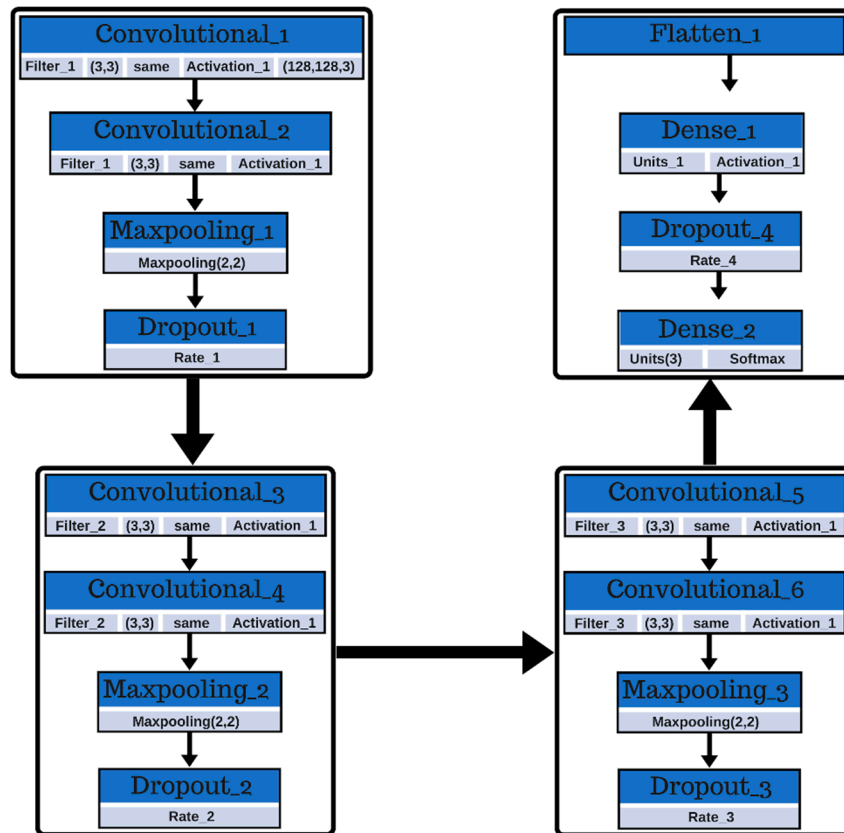


Figure 5. Base structure of the convolutional neural network (CNN) model.

The compilation hyperparameters that were established for the convoluted neural network are presented in Table 1, where for the loss function, the categorical_crossentropy function was chosen, since it is recommended for when there are more than two classes, and in this case, we had three classes. In addition to this, we had the target in a categorical format, and we had the metrics parameter, where a classification metric was used for accuracy, given that from this function, the total percentage of hits of the CNN was obtained according to a test set and a regression metric, such as the mean squared error. On the other hand, we had the training optimizer, where this parameter was left for random assignment, so that it was left with that name in the value box, since it was the key of the dictionary to which it was linked.

Table 1. Compilation hyperparameters.

Parameter	Value
Optimizer	optimizer
Loss	categorical_crossentropy
Metrics	Accuracy, mean squared error

Dictionary of Hyperparameters

In order to carry out Random Search, we defined a dictionary of hyperparameters that will take place within the CNN model proposed above, where Filter_1 refers to the number of filters in convolution layers 1 and 2, Filter_2 refers to the filters applied in convolution layers 3 and 4, and Filter_3 refers to convolution filters 5 and 6. For each Dropout layer, there is a range of elimination

rates named Rate_n, where n goes from 1 to 4. In addition, Units_1 refers to the number of hidden layers that will be added in general to layer number 4, and finally, Activation_1 is the activation function that will be applied in each layer. This dictionary can be seen in Figure 6.

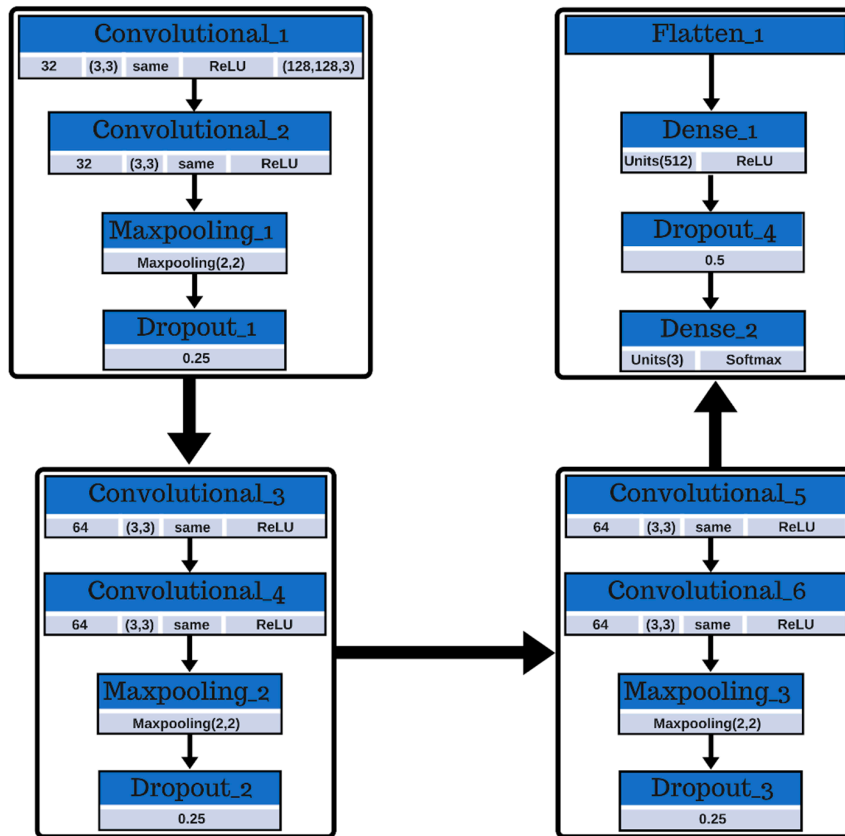


Figure 6. Selected model architecture.

The filter assignment is designed to cause an increase or an equality with respect to the previous convolution layer, increasing at a proportion of 2^n ; for this reason, the values between 16, 32, 64, and 128 were selected. On the other hand, it was selected as possible activation functions: ReLU and LeakyReLU; on the one hand, the first one is the activation function that is most commonly used in deep learning, and the second one is an attempt to solve the dying ReLU problem. The number of units or hidden networks was defined at a proportion of 2^n , proposing 64, 128, 256, and 512 as possibilities. The elimination rate must be between 0 and 1, but having a rate that is equal to 0 or 1 would be illogical; for this reason, a range between 0.25 and 0.75 was determined with a step of 0.25. Finally, we have the hyperparameter of batch_size and the optimizer; for the first one, a sequence of 2, 4, 8, 16, and 32 was generated, given that the data set was not very large for making larger batches. The options of the established optimizer were: Adam, RMSprop, Nadam, and Adadelta.

Model Preselection

Starting from the base structure of the CNN and the dictionary of hyperparameters, a total of 50 models were trained, randomly assigning the hyperparameters stored in the data dictionary. The training was carried out with a data set consisting of 480 images distributed among the three classes of figures that were handled; in this case, reference was made to Circle, Square, and Triangle, where each class consisted of 160 training images, distributed among images that were obtained from the AR-Sandbox, the Internet, and through some basic drawing tools. For the training of each of the models, the training parameters were defined by 10 epochs and a batch_size generated by the sequence of 2^n , up to $n = 5$, given the size of the training dataset.

The model preselection consisted of selecting the 10 best models, taking as a point of reference a high-accuracy coefficient, a low-loss coefficient, and a low-regression coefficient with the training dataset. Table 2 presents the preselected models with the data mentioned above.

Table 2. Preselected models.

Key	List of Values
Filter_1	16, 32
Filter_2	32, 64
Filter_3	64, 128
Activation_1	Relu, LeakyReLU
Rate_1	0.25, 0.5, 0.75
Rate_2	0.25, 0.5, 0.75
Rate_3	0.25, 0.5, 0.75
Rate_4	0.25, 0.5, 0.75
Units_1	64, 128, 256, 512
Batch_size	2, 4, 8, 16, 32
Optimizer	Adam, RMSProp, Nadam, Adadelta

Model's ID	Accuracy	Mean Squared Error	Loss
1	0.9871	0.0634	0.0088
2	0.9615	0.1021	0.0194
3	0.9615	0.0729	0.0138
4	0.9923	0.1537	0.0185
5	0.941	0.1448	0.0279
6	0.9333	0.1771	0.032
7	0.9641	0.2625	0.0251
8	0.9743	0.0958	0.0158
9	0.9820	0.1619	0.0129
10	0.8820	0.2886	0.0551

Model Selection

In order to select a single model for integration with AR-Sandbox, the validation was performed using the original dataSetOriginal and dataSetFilter, evaluating the loss validation coefficient, the accuracy validation coefficient, the validation coefficient of the mean squared error, and its respective variation between datasets. Table 3 shows each model with the values obtained with both datasets.

Table 3. Preselected model validation.

Measure	Dataset	1	2	3	4	5	6	7	8	9	10
Validation_accuracy	Original	0.7333	0.7111	0.7111	0.6444	0.5333	0.6111	0.7000	0.6555	0.7111	0.5888
	Filter	0.7777	0.8999	0.7666	0.8333	0.7555	0.7444	0.8700	0.7666	0.8666	0.7222
	Difference	0.0444	0.1888	0.0555	0.1889	0.2222	0.1333	0.1700	0.1111	0.1555	0.1334
Validation_Loss	Original	1.0977	1.0986	1.0978	1.1144	1.1027	1.0975	0.7200	1.0978	1.0985	0.9946
	Filter	0.6981	0.6234	0.7078	0.7630	0.7082	0.7320	0.3600	0.6678	0.4541	0.8605
	Difference	0.3996	0.4752	0.3900	0.3514	0.3945	0.3655	0.3600	0.4300	0.6444	0.1341
Validation_mean_squared_error	Original	0.2222	0.2222	0.2222	0.2257	0.2231	0.2219	0.2326	0.2222	0.2222	0.1766
	Filter	0.1061	0.0595	0.1340	0.0922	0.1052	0.1308	0.1108	0.1074	0.0651	0.1665
	Difference	0.1161	0.1627	0.0882	0.1335	0.1179	0.0911	0.1218	0.1148	0.1571	0.0101

According to the data presented in Table 3, model number 7 was chosen, given that it has the highest accuracy coefficient demarcating the fourth largest variation of this metric; in addition, it has the lowest coefficient of categorical_crossentropy, chosen as a function of loss, despite not having the most significant variation in this hyperparameter; finally, the value of the mean squared error coefficient, taken as a second metric, is an intermediate value within this set of models. All of this was calculated through validation with the dataset segmented by color-space. Figure 6 presents a representation of the architecture of the convolutional neural network model, in which the layers with their corresponding types are shown, denoting the characteristics and hyperparameters used.

4. Measurement of the Model and Performance Evaluation

In this section, the measurement of the selected model will be presented, making use of the aforementioned datasets: `dataSetOriginal` and `dataSetFilter`; with each of these test sets, the elaborated CNN was evaluated in order to determine the accuracy, the loss function, and the confusion matrix, and to observe the behavior of the receiver operating characteristic (ROC) curve.

4.1. Function Evaluate

DataSetOriginal. Table 4 shows the loss value, which is 0.72, the percentage of successes, which is 70%, and the percentage of mean squared error, which is 23.26%, according to the established metric.

Table 4. Function evaluation: `dataSetOriginal`.

Parameter	Function	Value
Loss	categorical_crossentropy	0.7200
Metrics	Accuracy	0.7000
	Mean squared error	0.2326

DataSetFilter. Table 5 shows the loss value, which is 0.36, the percentage of successes, which is 87%, and the percentage of mean squared error, which is 11.08%, according to that metric.

Table 5. Function evaluation: `dataSetFilter`.

Parameter	Function	Value
Loss	categorical_crossentropy	0.3600
Metrics	Accuracy	0.8700
	Mean squared error	0.1108

4.2. Confusion Matrix

In order to carry out the confusion matrix deployment with `dataSetOriginal` and `dataSetFilter`, this was done through the `sk-learn` library that was implemented in Python, as this library has a module to carry out the confusion matrix.

DataSetOriginal. Table 6 shows the number of hits that CNN had when testing with 30 images of each class in different orders; therefore, during analysis, the table of the neural network had a success rate of 61%.

Table 6. Confusion Matrix: `DataSetOriginal`.

	Circle	Square	Triangle
Circle	24	0	6
Square	13	13	4
Triangle	7	5	18

DataSetFilter. Table 7 shows the number of hits that CNN had when testing with 30 images of each class in different orders; therefore, when analyzing the table, the neural network had an 87.8% rate of success.

Table 7. Confusion Matrix: `DataSetFilter`.

	Circle	Square	Triangle
Circle	28	0	2
Square	5	22	3
Triangle	0	1	29

4.3. ROC Curve

According to the authors of References [25,26], the possibility of implementing the ROC curves as a factor to determine the level of precision in multi-classifier models was presented. For this reason, the ROC curves, which were obtained through the two validation datasets, are shown.

DataSetOriginal. Figure 7 shows the ROC curve for the dataset with the original images, where there are five curves, two of them at a general level, and the other three at a specific level. The general curves show the averages of areas under the curve at the micro- and macro-level; on the other hand, the curves at a specific level show the area under the curve (AUC) of each of the classes, these being 0, 1, and 2 which correspond to circle, square, and triangle, respectively.

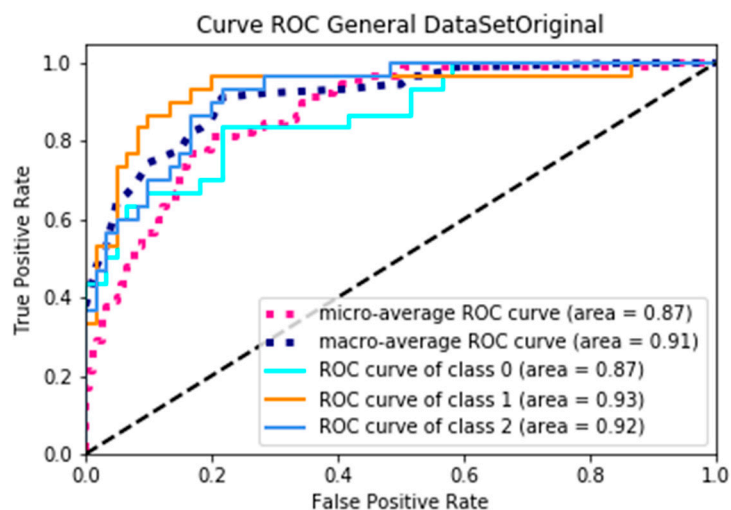


Figure 7. Receiver operating characteristic (ROC) Curve: DataSetOriginal.

DataSetFilter. Figure 8 shows the ROC curve for the dataset with the processed images, where the area under the curve of the averages at the micro-level, and the macro-level of the ROC curve is shown, as well as the AUCs of each of the classes of the geometric figures that composed the test dataset.

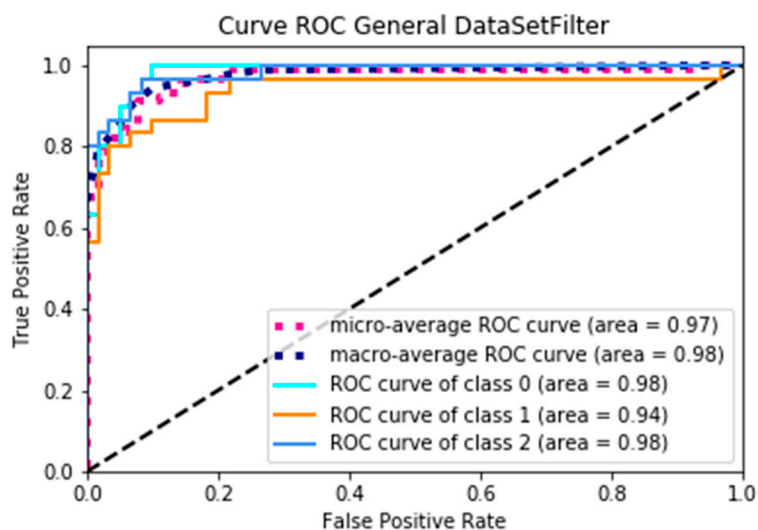


Figure 8. ROC Curve: DataSetFilter.

5. Results, Analysis, and Discussions

According to the Table 3, taking as a guide Reference [27], and the results and performances based on individual, average, and overall classification model accuracies, the 10 best models were selected

by Random Search, and variations in the coefficients of loss, accuracy, and mean squared error that were input to their validation through `dataSetOriginal` and `dataSetFilter`, are presented. An average variation of 0.3945 in the function of loss, and an average variation of 0.1483 in the function accuracy, and a variation of 0.1118 in the function mean squared error, were obtained.

From the evaluation made for the CNN model, with `dataSetOriginal` and `dataSetFilter`, the following aspects were analyzed: loss function, hit metric, confusion matrix, and ROC curve, and the following results were obtained:

When evaluating the CNN with `dataSetOriginal`, 72% was obtained as a function of loss, while with `dataSetFilter`, 36% was obtained; therefore, a 36% decrease was obtained. On the other hand, through `dataSetOriginal`, a metric of hits was obtained with a percentage of 70%; while using `dataSetFilter`, 87% correct answers were obtained, presenting an increase of 17% between the two test datasets.

Regarding the confusion matrix, the percentage of correct answers when using `dataSetOriginal` was 61%, while with `dataSetFilter`, a percentage of 87.7% was obtained, presenting an increase of 26.7%. In addition, in Table 7, the specific analysis for each of the geometric figures is presented.

An ROC curve is a graph that shows the performance of a classification model in all of the classification thresholds [28]. When analyzing an ROC curve, the determining parameter is the area under the curve (AUC); therefore, it is the factor that will be analyzed next.

Figure 8 represents the ROC curve when using `dataSetOriginal`, where, according to the AUC, the minimum average is 0.87 and the maximum average is 0.91, obtaining an average yield of 0.89; on the other hand, Figure 9 presents the curve of the ROC when using `dataSetFilter`, where the minimum average of the AUC is 0.97 and the maximum average is 0.98, obtaining an average yield of 0.975, and presenting an increase of 0.085 yield. Through a specific analysis, Table 8 presents the data and variations of the AUC of each of the classes by using the two test datasets.

Table 8. Confusion matrix analysis.

Datasets	Figure	Hits (%)	Failures (%)
dataSetOriginal	Circle	80	20
	Square	43.3	56.3
	Triangle	60	40
dataSetFilter	Circle	93.3	6.7
	Square	73.3	26.3
	Triangle	96.7	3.3

From Table 9, the performance in each of the classes presented variations with respect to the other test datasets, but the AUC value for image recognition, which previously had segmentation by color-space, was higher.

Table 9. ROC curve analysis.

Datasets	Class	Figure	AUC
dataSetTOrginal	0	Circle	0.87
	1	Square	0.93
	2	Triangle	0.92
dataSetFilter	0	Circle	0.98
	1	Square	0.94
	2	Triangle	0.98

In addition, we agree with results shown in Reference [29], since it states that for a prediction model to be considered optimal, the curve described must be convex, and in this case in Figure 9, taking the Macro curve as a reference, we can show that convexity occurs, since when making a region

between this curve and the diagonal, putting any two points in the region, and passing a line through them, the curve is entirely within this region.

6. Conclusions and Future Works

From the analysis presented, it can be determined that: when evaluating the set of preselected CNN models with two data sets, one of them is previously processed by applying segmentation by color-space; an average decrease of 39.45% is obtained in the function of loss categorical_crossentropy, which is an increase of 14.83% on average in the accuracy coefficient, and a decrease of 11.18% on average in the regression coefficient.

On basis of the analysis presented, when evaluating the CNN with a previously processed dataset by applying color-space segmentation, a decrease of 36% in the loss value is obtained, increasing the value of hits generated by the accuracy metric by 17%, given that the distance between the value of the prediction and the expected value decreases. By implementing color-space segmentation to a set of test data, a positive contribution was made to the identification and extraction of patterns or characteristics that are necessary for the classification of images. Since the data has a coefficient of 0.8221 for the Jaccard method and a coefficient of 0.8767 for the Sørensen-Dice method, and since its probabilities are greater than 0.70, we can conclude that the segmentation of images by color-space contributes to image processing and subsequent prediction, given that through these values, it can be inferred that when applying segmentation by color space, the original and segmented images retain a large percentage of similarity, emphasizing the definition of the characteristics of the images.

When the area under the ROC curve is taken as a reference, an average of 0.975, corresponding to the CNN performance, is presented under the processed dataset, increasing by a value of 0.085 with respect to the average AUC generated from the raw dataset.

From the implementation of these tools, it is evident that the combination of areas, such as multimedia and artificial intelligence, provides a great field of action to continue researching and making proposals. The development of the technologies worked provides opportunities for making proposals in sectors such as health and education. In these sectors, proposals can be made for motor skill development, the development of basic knowledge in early childhood, reactions to certain situations, and psychology, among others.

In future work, it is proposed that an immersive environment is implemented by using augmented reality tools and devices to support motor therapy in children, continuously monitoring the child's emotional behavior through brain-computer interfaces. On the other hand, an expansion of the scope of the developed CNN is planned, in order to achieve the identification of other geometric figures, forms, and symbols, in order to apply it to the abovementioned fields of action.

Author Contributions: A.O.R.R. and D.E.C.M. worked on the implementation, P.A.G.G and C.E.M.M. worked on the methodological part, R.G.C. worked on the global and methodological review of the paper.

Funding: This research received no external funding.

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

References

1. Shi, W.; Caballero, J.; Husz, F.; Totz, J.; Aitken, A. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
2. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 640–651.
3. Giorgis, S.; Mahlen, N.; Anne, K. Instructor-Led Approach to Integrating an Augmented Reality Sandbox into a Large-Enrollment Introductory Geoscience Course for Nonmajors Produces No Gains. *J. Geosci. Educ.* **2017**, *65*, 283–291. [[CrossRef](#)]

4. Woods, T.; Reed, S.; His, S.; John, W.; Woods, M. Pilot Study Using the Augmented Reality Sandbox to Teach Topographic Maps and Surficial Processes in Introductory Geology Labs. *J. Geosci. Educ.* **2016**, *64*, 199–214. [[CrossRef](#)]
5. Github. Talos: Manual de Usuario. [En línea]. Available online: https://autonomio.github.io/docs_talos (accessed on 15 November 2018).
6. Hijazi, S.; Kumar, R.; Rowen, C. Using Convolutional Neural Networks for Image Recognition. *Cadence* 2015. Available online: https://ip.cadence.com/uploads/901/cnn_wp-pdf (accessed on 4 November 2018).
7. Neha, S.; Vibhor, J.; Anju, M. An Analysis of Convolutional Neural Networks for Image Classification. *Procedia Comput. Sci.* **2018**, *132*, 337–384.
8. Niioka, E.H.; Satoshi, A.; Yoshimura, A.; Ohigashi, H.; Tagawa, S.; Miyake, J. Classification of C₂C₁₂ cells at differentiation by convolutional neural network of deep learning using phase contrast images. *Hum. Cell* **2018**, *31*, 87–93. [[CrossRef](#)] [[PubMed](#)]
9. Congcong, Z.; Xiaoyan, X.; Xiaomei, L.; Ying-Jie, C.; Wu, Z.; Jun, C.; Chengyun, Z.Z.L. White Blood Cell Segmentation by Color-Space-Based K-Means Clustering. *Sensors* **2014**, *14*, 16128–16147.
10. Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
11. Lee, K.; Lee, J.; Lee, J.; Hwang, S.; Lee, S. Brightness-Based Convolutional Neural Network for Thermal Image Enhancement. *IEEE Access* **2017**, *5*, 26867–26879. [[CrossRef](#)]
12. Yao, C.; Zhang, Y.; Zhang, Y.; Liu, H. Application of Convolutional Neural Network in Classification of High Resolution Agricultural Remote Sensing Images. *Int. Arch. Photogram. Remote Sens. Spat. Inf. Sci.* **2017**, *42*, 989–992. [[CrossRef](#)]
13. Pang, S.; Du, A.; Orgun, M.; Yu, Z. A novel fused convolutional neural networks for biomedical image classification. *Med. Biol. Eng. Comput.* **2018**. [[CrossRef](#)] [[PubMed](#)]
14. Real, R.; Vargas, J. The Probabilistic Basis of Jaccard’s Index of Similarity. *Syst. Biol.* **1996**, *45*, 380–385. [[CrossRef](#)]
15. Shamir, R.; Duchin, Y.; Kim, J.; Sapiro, G.; Harel, N. Continuous Dice Coefficient: A Method for Evaluating Probabilistic Segmentations. *Surgical Inf. Sci.* **2018**. [[CrossRef](#)]
16. Zou, K.; Warfield, S.; Bharatha, A.; Tempny, C.; Kaus, M.; Haker, S.; Wells, W.; Jolesz, F.; Kikinis, R. Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index. *Radiol. Alliance Health Serv. Res.* **2004**, *11*, 178–189.
17. Hernández, F.; de Ory, E.; Aguilar, S.; Crespo, R. Residue properties for the arithmetical estimation of the image quantization table. *Appl. Soft Comput.* **2018**, *67*, 309–321. [[CrossRef](#)]
18. Meza, J.; Espitia, H.; Montenegro, C.; Crespo, R. Statistical analysis of a multi-objective optimization algorithm based on a model of particles with vorticity behavior. *Soft Comput.* **2016**, *20*, 3521–3536. [[CrossRef](#)]
19. Wang, S.-H.; Tang, C.; Sun, J.; Yang, J.; Huang, C.; Phillips, P.; Zhang, Y.-D. Multiple Sclerosis Identification by 14-Layer Convolutional Neural Network with Batch Normalization, Dropout, and Stochastic Pooling. *Front. Neurosci.* **2018**, *12*, 818. [[CrossRef](#)]
20. Kushwaha, P.; Welekar, R. Feature Selection for Image Retrieval based on Genetic Algorithm. *Int. J. Interact. Multimed. Artif. Intell.* **2016**, *4*, 16–21. [[CrossRef](#)]
21. Pacheco, A.; Barón, H.; Crespo, R.; Espada, J. Reconstruction of High Resolution 3D Objects from Incomplete Images and 3D Information. *Int. J. Interact. Multimed. Artif. Intell.* **2014**, *2*, 7–16. [[CrossRef](#)]
22. Rezaie, A.A.; Habiboghli, A. Detection of Lung Nodules on Medical Images by the Use of Fractal Segmentation. *Int. J. Interact. Multimed. Artif. Intell.* **2017**, *4*, 15–19.
23. Rosyadi, H.; Gökhan, C. Augmented reality sandbox (AR sandbox) experimental lanscape for fluvial, deltaic and volcano morphology and topography models. *Turqua* **2016**. [[CrossRef](#)]
24. Wang, X.; Hansch, R.; Ma, L.; Hellwich, O. Comparison of Different Color Spaces for Image Segmentation using Graph-cut. In Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal, 5–8 January 2014; pp. 301–308.
25. Blessie, E.C.; Karthikeyan, E.; Selvaraj, B. Empirical study on the performance of the classifiers based on various criteria using ROC curve in medical health care. In Proceedings of the International Conference on Communication and Computational Intelligence (INCOCCI), Erode, India, 27–29 December 2010.

26. Wang, X.; Shu, P.; Cao, L.; Wang, Y. A ROC Curve Method for Performance Evaluation of Support Vector Machine with Optimization Strategy. In Proceedings of the International Forum on Computer Science-Technology and Applications, Chongqing, China, 25–27 December 2009.
27. Senthilnath, J.; Kulkarni, S.; Benediktsson, J.A.; Yang, X.S. A Novel Approach for Multispectral Satellite Image Classification Based on the Bat Algorithm. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 599–603. [[CrossRef](#)]
28. Wald, N.; Bestwick, J. Is the area under an ROC curve a valid measure of the performance of a screening or diagnostic test? *J. Med. Screen.* **2014**, *21*, 5160. [[CrossRef](#)] [[PubMed](#)]
29. Prati, R.C.; Batista, G.E.A.P.A.; Monard, M.C. Curvas ROC para avaliação de classificadores. *IEEE Latin Am. Trans.* **2008**, *6*, 215–222. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).