Comput Methods Programs Biomed

-
-
-

. 2017 Jul;146:109-123.
doi: 10.1016/j.cmpb.2017.05.005. Epub 2017 May 20.

I. An Intelligent Support System for Automatic Detection of Cerebral Vascular Accidents From Brain CT Images

Elmira Hajimani[1], M G Ruano[2], A E Ruano[3]
Affiliations expand

- PMID: 28688480

- DOI: 10.1016/j.cmpb.2017.05.005

An Intelligent Support System for Automatic Detection of Cerebral Vascular Accidents from Brain CT Images

Elmira Hajimani[a,*], M.G. Ruano[b], A. E. Ruano[c]

[a] Faculty of Science and Technology, University of Algarve, Faro, Portugal, (e-mail: ehajimani@csi.fct.ualg.pt).

[b] Faculty of Science and Technology, University of Algarve, Faro, Portugal and CISUC, University of Coimbra, Portugal (e-mail: mruano@ualg.pt)

[c] Faculty of Science and Technology, University of Algarve, Faro, Portugal and IDMEC, Instituto Superior Técnico, University of Lisbon, Portugal, (e-mail: aruano@ualg.pt)

[*]Corresponding author. Tel: (+351) 915 918 528. Postal address: Lab. 2.75, FCT, Universidade do Algarve, Campus Gambelas, 8005-139 Faro, Portugal.

Abstract

Objective: This paper presents a Radial Basis Functions Neural Network (RBFNN) based detection system, for automatic identification of Cerebral Vascular Accidents (CVA) through analysis of Computed Tomographic (CT) images.

Methods: For the design of a neural network classifier, a Multi Objective Genetic Algorithm (MOGA) framework is used to determine the architecture of the classifier, its corresponding parameters and input features by maximizing the classification precision, while ensuring generalization.

This approach considers a large number of input features, comprising first and second order pixel intensity statistics, as well as symmetry/asymmetry information with respect to the ideal mid-sagittal line.

Results: Values of specificity of 98% and sensitivity of 98% were obtained, at pixel level, by an ensemble of non-dominated models generated by MOGA, in a set of 150 CT slices (1,867,602 pixels), marked by a NeuroRadiologist. This approach also compares favorably at a lesion level with three other published solutions, in terms of specificity (86% compared with 84%), degree of coincidence of marked lesions (89% compared with 77%) and classification accuracy rate (96% compared with 88%).

Keywords: Neural Networks; Symmetry features; Multi-Objective Genetic Algorithm; Intelligent support systems; Cerebral Vascular Accident.

1. Introduction

Cerebral Vascular Accident (CVA), also called stroke, is caused by the interruption of blood supply to the brain, mainly due to a blood vessel blockage (i.e., ischemic), or by a haemorrhagic event. The cut-off of oxygen and nutrients supplies causes brain tissue irreversible damages, if not detected during the first 2-3 hours. Stroke accounted for approximately 1 of every 19 deaths in the United States in 2009, according to [1]. Computed Tomography (CT) is the most commonly used CVA imaging diagnosis modality, available at almost all emergency units. When CVA is timely diagnosed, morbidity and mortality are minimised [2].

The increasing occurrence of CVAs in developed and developing countries, together with the insufficient number of Neuroradiologists and the lack of full-time expert Radiologists in some institutions, exacerbated by the fact that each exam is constituted by several images requiring an accurate analysis, leads to delays on the production of the clinical final report and subsequent clinical treatment, compromising the CVA's recovery prognosis. In addition, at early stages of CVA, changes in the tomographic images may not be perceived by the human eye due to the inconspicuousness of the lesions.

For these reasoning, the existence of an automated system would be of paramount importance to detect subtle alterations, motivating the authors to create a computational intelligent application capable of assisting the Neuroradiologist in the analysis of CT scan images. The proposed application envisage enabling a preliminary trigger of a pathologic occurrence and a better performance of the CVA green line.

In this paper, a Radial Basis Functions Neural Network (RBFNN) based system is proposed for automatic detection of CVAs from brain CT images. The majority of the existing methods for designing the neural network classifier do not use an approach that determines the model structure that best fits the application at their hand, while simultaneously selecting the best input features. Moreover, their design typically does not consider multiple conflicting objectives such as minimization of the number of False Detections (FD) in the training dataset while achieving a desired level of model generalization, or, maximising the classification precision while reducing the model complexity. For this purpose, an hybrid of a multi-objective evolutionary technique is used as a design framework for features and topology selection, and, state-of-the-art derivative based algorithms are used for neural network parameter estimation, taking into account multiple objectives, as well as their corresponding restrictions and priorities [2, 3].

Moreover, to the best of our knowledge, none of existing classifiers consider the asymmetry caused by lesions in the intracranial area. In the proposed approach, a group of symmetry features that were proposed in [4], is used along with other statistical features, as inputs to the designed classifier.

The rest of the paper is organized as follows: Section 2 provides an overview of existing lesion segmentation strategies. The data acquisition process is explained in section 3. Section 4 describes the features that are used in this study. Section 5 explains how the Multi Objective Genetic Algorithm (MOGA) is used to automatically design the RBFNN classifier. Experimental formulations and results are given in section 6. Section 7 discusses the results obtained in comparison with other published approaches. Conclusions are drawn in section 8.

2. Related works

Based on [5], a review of different lesion segmentation approaches, one can divide lesion segmentation strategies into two subgroups: supervised and unsupervised strategies.

Supervised approaches are those that use some kind of a priori information to perform the lesion segmentation. The group of supervised strategies can be further subdivided into two sub-groups:

a) In the first subgroup, all approaches use atlas information, therefore requiring the application of a registration process to the analyzed image to perform the segmentation.

As an example, the authors in [6] used a combination of two techniques for brain lesion detection from CT perfusion maps: finding asymmetries among the two hemispheres and then comparing the captured images to a brain atlas anatomy. For generating the asymmetry map, first the symmetry axis is approximated as the straight line that minimizes the least square error between all centers of masses' coordinates, and then the intensity values of the corresponding pixels on the left and right side of the image are compared. Those pixels with a significant difference are considered as potential lesions. To perform a detailed description of lesions a second step is required, where position image registration of the brain template is made. The goal of the registration algorithm is to maximize the similarity between the template image and the newly acquired image.

The work done in [7] can also be considered in this subgroup. This study presents an automated template-guided algorithm for the segmentation of ventricular CerebroSpinal Fluid (CSF) from ischemic stroke CT images. In the proposed method, the authors use two ventricular templates, one extracted from a normal brain (VT1) and the other built from several pathological scans (VT2). VT1 is used for registration and VT2 to define the region of interest. In the registration process, they use the Fast Talairach Transformation [8], which takes care of the "tilting" angle. Automatic thresholding is applied on a slice-by-slice basis, to cater for the variability of CSF intensity values across the slices in the same scan. The distributions of the CSF, White Matter (WM) and Gray Matter are analyzed and only voxels in the CSF range and WM range are used in the calculation of the histogram, employed by Otsu's automatic thresholding algorithm [9]. Finally, artifacts are removed with the help of VT2.

b) All approaches which perform an initial training step on features extracted from manually segmented images, annotated by Neuroradiologists, can be considered as another subgroup of supervised strategies [5]. In this subgroup, different classifiers, such as Artificial Neural Networks, k-Nearest Neighbors, AdaBoost, Bayesian classifiers or decision trees, alone or combined, have been used to perform the segmentation. The work presented in this article can be considered in this category.

The method applied in [10] is also an example of this subgroup. Primarily the method uses morphology operations and wavelets based filtering for image denoising. Then asymmetric parts of the brain and their neighbors are extracted being considered as the region of interest for specifying relevant features (such as texture, contrast, homogeneity, etc). Finally, k-means clustering and Support Vector Machines (SVM) are used for classification and provide the contour of the brain tumor.

The work presented in [11] uses a wavelet based statistical method for classifying brain tissues into normal, benign and malignant tumours. The authors first obtain the second level discrete wavelet transform of each CT slice. The Gray Level Co-occurrence Matrix (GLCM) is then calculated over the low frequency part of the transformed image. Finally, features are calculated from the GLCM matrix. Genetic algorithms and principle component analysis are employed for feature selection and SVM for classification.

In [12] a computer tomography (CT) brain image analysis system is proposed, with four phases: enhancement, segmentation, feature extraction and classification. The enhancement phase reduces the noise using an edge-based selective median filter (ESMF); the segmentation phase extracts the suspicious region applying a modified version of a genetic algorithm; the feature extraction phase extracts the textural features from the segmented regions and the last phase classifies the image. To diagnose and classify the image, the authors used a RBFNN classifier.

Regarding unsupervised strategies, where no prior knowledge is used, two different sub-groups can also be identified:

- A sub-group of methods that segment the brain tissue to allow lesion segmentation. These approaches usually detect lesions as outliers on each tissue, rather than adding a new class to the classification problem. The works presented in [13] and [14] follow this strategy.

- A sub-group that uses only lesion properties for segmentation. These methods directly segment the lesions according to their properties, without providing tissue segmentation. The works described in [15, 16] belong to this category.

3. Data acquisition

A database of existing CT images was used in this prospective study. All images were obtained from the same Siemens equipment. Each exam, where no contrast was applied, was composed by two ranges, one to study the posterior fosse and the other covering the remaining of the brain (5/5 mm and 10/10 mm, respectively).

In order to collect the opinion of Neuroradiologists about pathologic areas within brain CT images in an accurate and convenient way, a web-based tool was developed [17]. Using this tool, the existing database of CT images was used for Neuroradiologists to analyze and mark the images either as normal or abnormal. For the abnormal ones, the doctor was asked to designate the lesion type and to manually trace the contours of abnormal region(s) on each CT's slice image. Figure 1 shows the activity diagram of the Neuroradiologist in the developed tool.

The administrator of the developed web-based tool can then download a text file in which the coordinates of each marked pixel (i.e., a lesion) are specified. These pixels are considered as abnormal data samples. The resolution of each CT slice is 512×512 pixels and the intensity value of each pixel is an integer in the range [0 255], 0 being completely black and 255 completely white. Within a CT slice, all the intracranial pixels which are not marked as lesions will be considered as normal data samples.

Our collaborating Neuroradialogist registered his opinion for 7 patients (150 CT slices). 24 out of the 150 CT slices had lesions within their intracranial area, corresponding to 64,786 abnormal pixels. All lesions were marked as ischemic stroke. To obtain the coordinates of all normal pixels, Algorithm 1 (described below) is used. It calls Algorithm 2 (described afterwards) for artifact removal on each image. Applying Algorithm 1, we obtained 1,802,816 normal pixels. As a result, we have a total of 1,867,602 normal and abnormal pixels to work with.

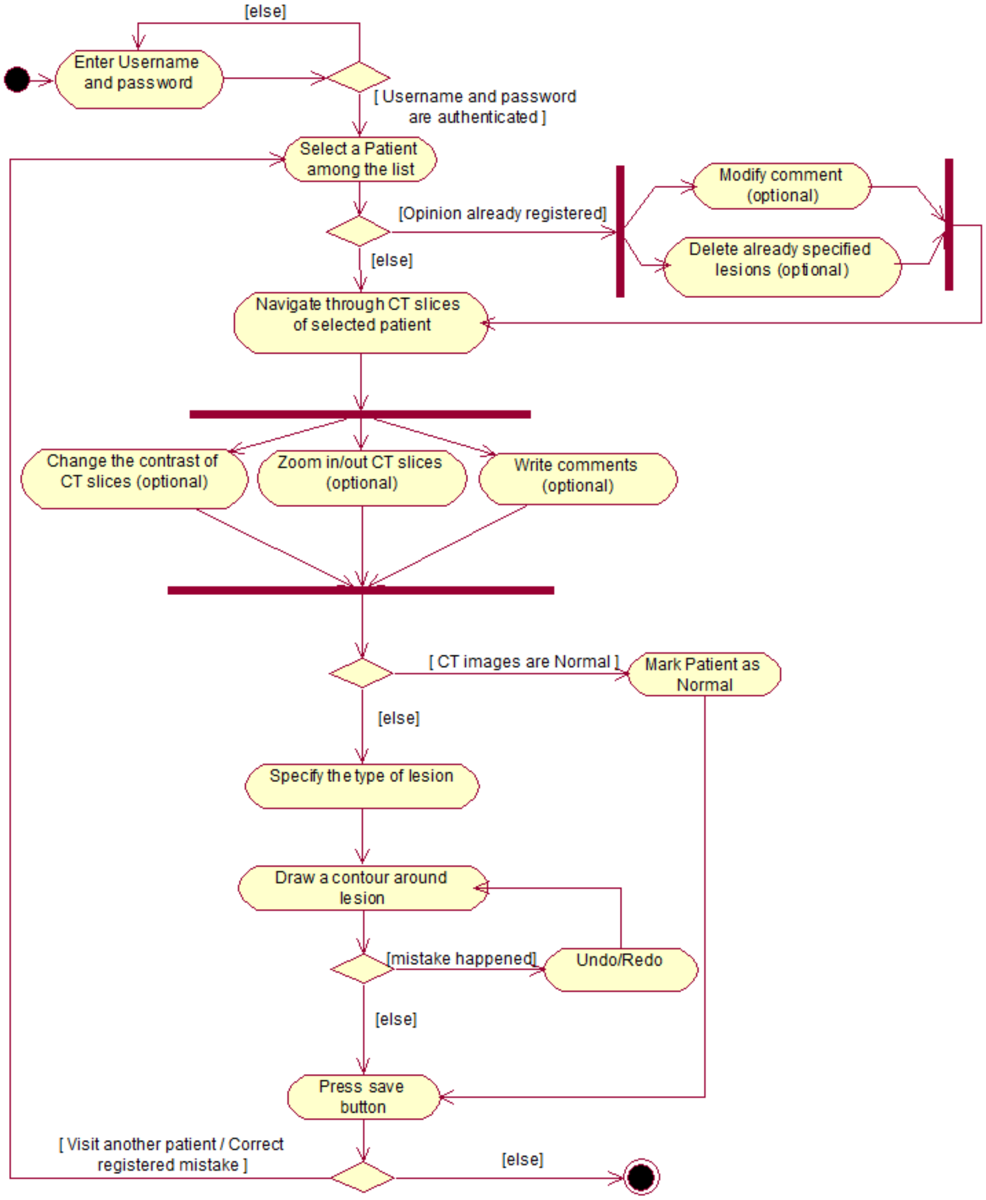

[else]
Enter Username and password
[Username and password are authenticated]
Select a Patient among the list
[Opinion already registered]
Modify comment (optional)
Delete already specified lesions (optional)
[else]
Navigate through CT slices of selected patient
Change the contrast of CT slices (optional)
Zoom in/out CT slices (optional)
Write comments (optional)
[CT images are Normal]
Mark Patient as Normal
[else]
Specify the type of lesion
Draw a contour around lesion
[mistake happened]
Undo/Redo
[else]
Press save button
[Visit another patient / Correct registered mistake]
[else]

Figure1. Activity diagram of the Neuroradiologist in the data acquisition tool

Algorithm 1 Obtaining the coordinate of normal pixels

```
Input: text file, say T, where the coordinate of abnormal pixels and the path from which the image can be retrieved are saved.
1. Let $Exams$ be a structure that is constructed from text file T. $Exams(i)$ contains the information of each CT exam.
2. For $i = 1$ to $length(Exams)$
    2.1. Pass $Exams(i)$ through Algorithm 2, to remove the skull and other artifacts.
    2.2. For each image in $Exams(i)$
        2.2.1. Let $X, Y$ be two vectors containing the location of abnormal pixels
        2.2.2. Let $P(a, b)$ be the intensity of the pixel located in $(a, b)$
        2.2.3.  If $P(a,b) \neq 0$ and $a \notin X$ and $b \notin Y$
            2.2.3.1.    Insert $(a,b)$ and $path(image)$ as a row in a text file O.
                End if
        End for
    End for
Output: text file O
```

Algorithm 2 Artifact removal algorithm in brain CT images [18]

```
Input: Brain CT images of one examination
1.  Skull detection:
    1.1. Remove pixels whose intensities are less than 250.
    1.2. Use the Connected Component algorithm [19] to choose the largest component as the candidate
         skull.
    1.3. Remove the small holes within the candidate skull, by inverting the matrix of candidate skull and
         applying the Connected Component algorithm for the second time. Those connected components
         whose area are less than 200 pixels are considered as holes and will be filled using the bone
         intensity value
2.  Removing CT slices with either unclosed skulls or skull containing too many separate regions:
    Having completed step 1.3, we have already all connected components at hand. As a result, we can
    count the number of big holes (e.g., areas more than 200 pixels wide). If this number is equal to 2, it
    will be considered as closed skull; otherwise the slice will be removed from the desired set.
3.  Intracranial area detection: All CT images that successfully passed step 2 contain only two black
    regions separated by the skull. To detect which black area is related to the intracranial part, the mass
    centre of the skull is calculated, the region containing the mass centre being considered as intracranial
    area.
Output: Intracranial part of a subset of input CT images.
```

4. Feature space

Having a set of pixel coordinates whose labels (normal or abnormal) are already determined by expert, we are now able to produce our dataset by extracting the corresponding features from the images. Each CT image is represented as a matrix I with M rows and N columns where $I(m,n)$ stands for the intensity of pixel in row m and column n. Three groups of features are used to construct the feature space: first order statistics, second order statistics and symmetry features. Table I describes the 51 features considered. First order statistics estimate properties of individual pixel values (e.g. $f1$ to $f16$ and $f37$ to $f41$), ignoring the spatial interaction between the image pixels. Second order features estimate properties of two pixel values occurring at specific locations relative to each other by constructing a Gray Level Co-occurrence Matrix. To extract some of the first and second order statistical features, a window w of size 31*31 [20, 21] centered at the pixel (x,y) is employed.

The variance of pixel intensities within a window w is denoted by var_w. Given w centred at point (x,y) , $\mathrm{L_h}$ is a row vector with the intensities of the 31 pixels taken from the horizontal line centered at (x,y) and $\mathrm{L_v}$ is a column vector with the intensities of the 31 pixels taken from the vertical line centered at (x, y). For calculating features $f15, f16$ and $f38$ to $f41$, $L = 8$ gray levels of the histogram of pixel intensities within window w are calculated. Each bin of histogram is represented by H_l.

A GLCM matrix is a two-dimensional matrix C where both the rows and the columns represent a set of possible image values G (e.g. gray tones). The value of $C(i\ ,j)$ indicates how many times the value i co-occurs with value j in some designated spatial relationship. The spatial relationship is usually defined by a distance d and a direction θ. In this study, to calculate the 8 gray level GLCM of w, the displacement parameters considered were $d = 1$ and $\theta = 0,45,90,135$. As a result, 4 GLCM matrices are derived, each one belonging to one specific θ and then the average is computed in order to obtain a direction invariant GLCM matrix.

In the formulas used in Table I, the mean value of matrix C is represented by μ and the mean and standard deviation for the rows and columns of C are defined in (1) and (2) respectively.

$$\mu_x = \sum_{i,j} i.C(i,j)\ ,\ \mu_y = \sum_{i,j} j.C(i,j) \tag{1}$$

$$\sigma_x = \sum_{i,j}(i-\mu_x)^2.C(i,j)\ ,\sigma_y = \sum_{i,j}\left(j-\mu_y\right)^2.C(i,j) \tag{2}$$

Moreover, $C_x(i)$ is the i^{th} entry in the marginal-probability matrix obtained by summing the rows of $C(i,j)$ and $C_y(i)$ is obtained by summing the columns of $C(i,j)$.

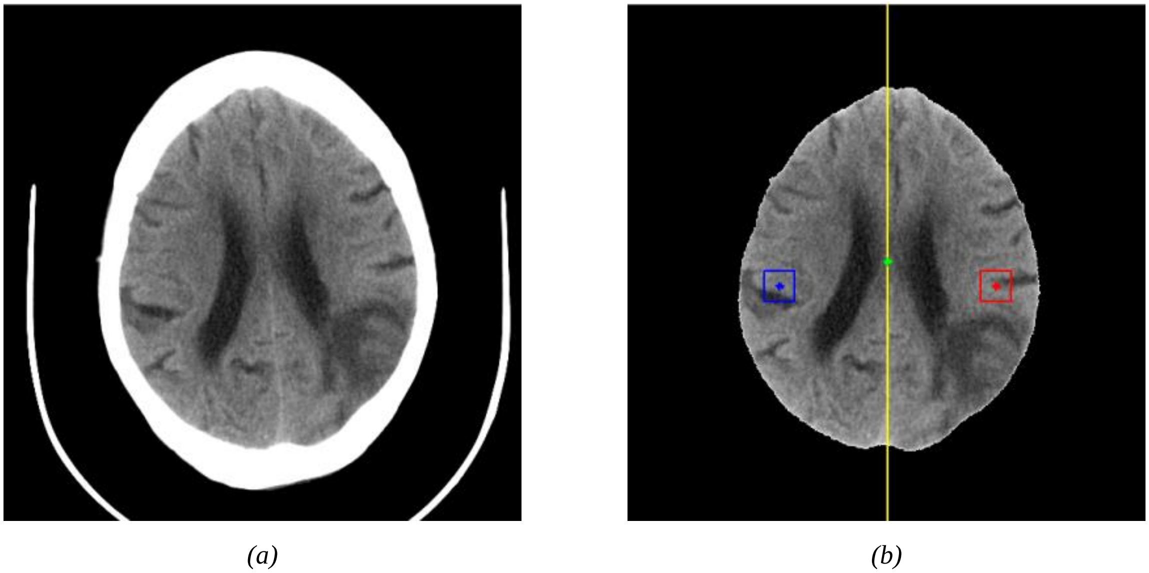
Given the ideal mid-sagittal line, symmetry features aim to compare one side of the brain to the other side and discover if there are any suspicious differences. To detect the ideal midline and to rotate tilted images to make the ideal midsagittal line perpendicular to the x-axis, the method summarized in Algorithm 3 is employed.

Algorithm 3 Ideal midline detection of the brain CT [18, 22]

```
Input: Brain CT images of one exam
1. Use Algorithm 2 to remove artifacts.
2. Since the concave shape of intracranial region will affect the accuracy of search for finding ideal
   midline, CT slices with high amount of concavity are found and excluded:
   2.1. For each CT slice
        2.1.1.Extract the contour of intracranial region.
        2.1.2.$Concavity = 0$
        2.1.3.For $\emptyset = 0$ to 180
              2.1.3.1.    Rotate contour by $\emptyset$ degree.
              2.1.3.2.    $Concavity_{\emptyset} = 0$
              2.1.3.3.    For $i = 1$ to $number\ of\ rows$
                          2.1.3.3.1.    Scan the pixels of the contour in row $i$ and define the Far Left ($FL_i$) and
                                        Far Right ($FR_i$) junctions.
```

```
                2.1.3.3.2. Let $C_i$ be the number of pixels in row $i$ which reside between $FL_i$ and $FR_i$ and that are not located inside the intracranial region.
                2.1.3.3.3. $Concavity_{\emptyset} = Concavity_{\emptyset} + C_i$
            End for
            2.1.3.4. $Concavity += Concavity_{\emptyset}$
        End for
    End for
    2.2. Sort CT slices based on their corresponding $Concavity$ values and select the first $\lambda$ slices with the least amount of concavity.
3. To find the line that maximizes the symmetry of the resulting halves, a rotation angle search around the mass centre of the skull is performed:
    3.1. For each CT slice remaining from step 2:
        3.1.1. Let $\theta$ be the maximum angle that a given CT image can be tilted.
        3.1.2. Let $S_j$ be the symmetry cost at angle $j$
        3.1.3. For $j = -\theta$ to $\theta$
            3.1.3.1. Calculate $S_j = \sum_{i=1}^{n} |l_i - r_i|$ where $n$ is the number of rows in the current CT slice, $l_i$ and $r_i$ are the distances between the current approximate midline and the left and right side of the skull edge in row, respectively.
        End for
        3.1.4. Select rotation angle $j$ whose symmetry cost $S_j$ is minimum.
    End for
    3.2. The final rotation degree for all CT slices is determined as the median value of rotation angles obtained for each CT slice.
4. Rotate all CT slices around their corresponding skull mass centre based on the rotation degree obtained in step 3.2
5. Line $x = x_{mass\ center(i)}$ is considered as the ideal midline of the brain CT image $i$.
Output: CT images are aligned and their corresponding ideal midlines determined.
```

To extract symmetry features, a window $w_1$of size $s \times s$ centred at pixel (x, y), marked by a clinical expert as normal or abnormal, and its contralateral part with respect to the midline, window w_2 centred at the pixel (x', y'), are considered (please see Figure 2-b).



(a) *(b)*

Figure 2. (a) Original brain CT image; (b) After skull removal and realignment, the ideal midline is drawn in yellow. The green point shows the mass centre (centroid) of the skull upon which the alignment rotation is performed. A window of size 31x31 is considered around the pixel located at (365,279) and shown in red; its contralateral part with respect to the midline is shown in blue [4].

Having identified w_1 and w_2, we can then specify how similar these two regions are by calculating the Pearson Correlation Coefficient (PCC), as stated in (3). The L_1 norm and squared L_2 norms are also two dissimilarity measures that can be obtained using (4) and (5), respectively. Comparing the intensity value of the pixel that is marked by the expert, and its corresponding pixel in the contralateral part, another symmetry feature defined in (6) can be obtained.

$$PCC = \frac{1}{n}\sum_{i=1}^{s}\sum_{j=1}^{s}\left(\frac{I_{w_1}^{i,j}-\mu_{w_1}}{\sigma_{w_1}}\right)\left(\frac{I_{w_2}^{i,j}-\mu_{w_2}}{\sigma_{w_2}}\right) \tag{3}$$

$$L_1 = \sum_{i=1}^{s}\sum_{j=1}^{s}\left|I_{w_1}^{i,j} - I_{w_2}^{i,j}\right| \tag{4}$$

$$L_2^2 = \sum_{i=1}^{s}\sum_{j=1}^{s}\left(I_{w_1}^{i,j} - I_{w_2}^{i,j}\right)^2 \tag{5}$$

$$diff = I_{w_1}^{x,y} - I_{w_2}^{x',y'} \tag{6}$$

In the previous equations, $I^{i,j}$ is the intensity value of pixel located at (i,j) within the corresponding window; μ_{w_1}, σ_{w_1} μ_{w_2}, and σ_{w_2} are the mean and standard deviation of the intensity values within window w_1 and its contralateral part, window w_2 , respectively. In Table I, $f42$ to $f51$ are symmetry features for three different window sizes ({11,21,31}).

In Table I, features $f1$ to $f13$ are taken from [21]; $f13$ to $f16$ are used in [23]; $f17$ to $f23$ and $f25$ to $f34$ are from [24]; $f24$ is a built-in MATLAB function; $f35$ and $f36$ are used in [25]; $f38$ to $f41$ are obtained from [20] and $f42$ to $f51$ were proposed in [4].

Table I. Features considered

	Description
$f1$	$I(x,y)$
$f2$	$\min\limits_{m,n\,\in\,w} I(m,n)$
$f3$	$\operatorname*{average}\limits_{m,n\,\in\,w} I(m,n)$
$f4$	$\max\limits_{m,n\,\in\,w} I(m,n)$
$f5$	$\operatorname*{median}\limits_{m,n\,\in\,w} I(m,n)$
$f6$	$\mathrm{std}_w=(\frac{1}{width(w)\times height(w)-1}\times\sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}}\sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}}(I(m,n)-f3)^2)^{1/2}$
$f7$	$\operatorname*{average}\limits_{1\le m\le M,\ 1\le n\le N} I(m,n)$
$f8$	$\operatorname*{average}\limits_{m,n\,\in\,w} I(m,n)-\operatorname*{average}\limits_{1\le m\le M,\ 1\le n\le N} I(m,n)$
$f9$	$I(x,y)-\operatorname*{average}\limits_{1\le m\le M,\ 1\le n\le N} I(m,n)$
$f10$	$\mathrm{Plh}=\sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}}\lvert \mathrm{L_h(x,n+1)-L_h(x,n)}\rvert$
$f11$	$\mathrm{plv}=\sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}}\lvert \mathrm{L_v(m+1,y)-L_v(m,y)}\rvert$
$f12$	cxm $=x/512$
$f13$	Skewness=$\frac{1}{{var_w}^3}\sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}}\sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}}(I(m,n)-f3)^3$
$f14$	Kurtosis=$\frac{1}{{var_w}^4}\sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}}\sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}}(I(m,n)-f3)^4$
$f15$	Energy=$\sum_{\mathrm{l=1}}^{\mathrm{L}}(\frac{H_l}{width(w)\times height(w)})^2$
$f16$	Entropy=$-\sum_{\mathrm{l=1}}^{\mathrm{L}}\frac{H_l}{width(w)\times height(w)}\log_2\{\frac{H_l}{width(w)\times height(w)}\}$
$f17$	Autocorrelation=$\sum_{i,j}(ij)C(i,j)$
$f18$	Correlation=$\frac{\sum_{i,j}(ij)C(i,j)-\mu_x\mu_y}{\sigma_x\sigma_y}$
$f19$	Cluster Prominence=$\sum_{i,j}\left(i+j-\mu_x-\mu_y\right)^4C(i,j)$
$f20$	Cluster shade=$\sum_{i,j}\left(i+j-\mu_x-\mu_y\right)^3C(i,j)$
$f21$	Dissimilarity=$\sum_{i,j}\lvert i-j\rvert.C(i,j)$
$f22$	GLCM Energy =$\sum_{i,j}C(i,j)^2$
$f23$	GLCM Entropy=$-\sum_{i,j}C(i,j)\log\left(C(i,j)\right)$
$f24$	Homogeneity= $\sum_{\mathrm{i,j}}\frac{C(i,j)}{1+\lvert i-j\rvert}$

$f25$	Homogeneity = $\sum_{\mathrm{i,j}} \mathrm{C(i,j)}/(1+(\mathrm{i}-\mathrm{j})^2)$
$f26$	Maximum probability=$\underset{i,j}{\mathrm{MAX}}\, C(i,j)$
$f27$	Sum of squares =$\sum_{i,j}(i-\mu)^2 C(i,j)$
$f28$	Sum average=$\sum_{i=2}^{2G} iC_{x+y}(i)$ where $C_{x+y}(k)=\sum_{i=1}^{G}\sum_{j=1}^{G} C(i,j) \mid i+j=k\ ,k=2,3,\dots,2G$
$f29$	Sum variance= $\sum_{i=2}^{2G}(i-f30)^2 C_{x+y}(i)$ where $C_{x+y}(k)=\sum_{i=1}^{G}\sum_{j=1}^{G} C(i,j) \mid i+j=k\ ,k=2,3,\dots,2G$
$f30$	Sum entropy=$-\sum_{i=2}^{2G} C_{x+y}(i)\log(C_{x+y}(i))$ where $C_{x+y}(k)=\sum_{i=1}^{G}\sum_{j=1}^{G} C(i,j) \mid i+j=k\ ,k=2,3,\dots,2G$
$f31$	Difference variance= $variance\ of\ C_{x-y}$ where $C_{x-y}(k)=\sum_{i=1}^{G}\sum_{j=1}^{G} C(i,j) \mid i-j=k\ ,k=0,1,\dots,G-1$
$f32$	Difference entropy= $-\sum_{i=0}^{G-1} C_{x-y}(i)\log\left(C_{x-y}(i)\right)$ where $C_{x-y}(k)=\sum_{i=1}^{G}\sum_{j=1}^{G} C(i,j) \mid i-j=k\ ,k=0,1,\dots,G-1$
$f33$	Information measure of correlation1= $\frac{f23-HXY1}{\max\{HX\ ,HY\}}$ where HX and HY are Entropies of C_x and C_y and $HXY1=-\sum_{i,j} C(i,j)log\{C_x(i)C_y(j)\}$
$f34$	Information measure of correlation2= $(1-exp[-2.0(HXY2-f23)])^{1/2}$ where $HXY2=-\sum_{i,j} C_x(i)C_y(j)log\{C_x(i)C_y(j)\}$
$f35$	Inverse difference normalized=$\sum_{i,j=1}^{G}\frac{C(i,j)}{1+\vert i-j\vert/G}$
$f36$	Inverse difference moment normalized=$\sum_{i,j=1}^{G}\frac{C(i,j)}{1+(i-j)^2/G^2}$
$f37$	$Var_w=\frac{1}{width(w)\times height(w)}\times\sum_{m=x-\frac{(height(w)-1)}{2}}^{x+\frac{(height(w)-1)}{2}}\sum_{n=y-\frac{(width(w)-1)}{2}}^{y+\frac{(width(w)-1)}{2}}(I(m,n)-f3)^2$
$f38$	$(H_1+H_2)/(width(\ w))^2$
$f39$	$(H_3+H_4)/(width(\ w))^2$
$f40$	$(H_5+H_6)/(width(\ w))^2$
$f41$	$(H_7+H_8)/(width(\ w))^2$
$f42$	$PCC\ ,\quad s=31$
$f43$	$diff$
$f44$	$L_1\ ,\quad s=31$
$f45$	$L_2^2\ ,\quad s=31$
$f46$	$PCC\ ,\quad s=21$
$f47$	$L_1\ ,\quad s=21$
$f48$	$L_2^2\ ,\quad s=21$
$f49$	$PCC\ ,\quad s=11$
$f50$	$L_1\ ,\quad s=11$
$f51$	$L_2^2\ ,\quad s=11$

5. Neural Network design using the Multi Objective Genetic Algorithm framework

The identification of Neural Network inputs, topology and parameters from data is often done iteratively in an ad-hoc fashion, focusing mainly on parameters identification. This is because the number of possibilities for the selection of the model structure (inputs and topology) are usually very large. Moreover, typically the design criterion is a single measure of the error obtained by the model, such as the mean-square error or the root-mean-square error, while typically the aim is to obtain a satisfactory performance (determined by typically more than one criterion) with small networks, i.e., the design problem should be formulated as a multiple-objective problem.

The model used for pixel classification is a Radial Basis Function Neural Network (RBFNN).

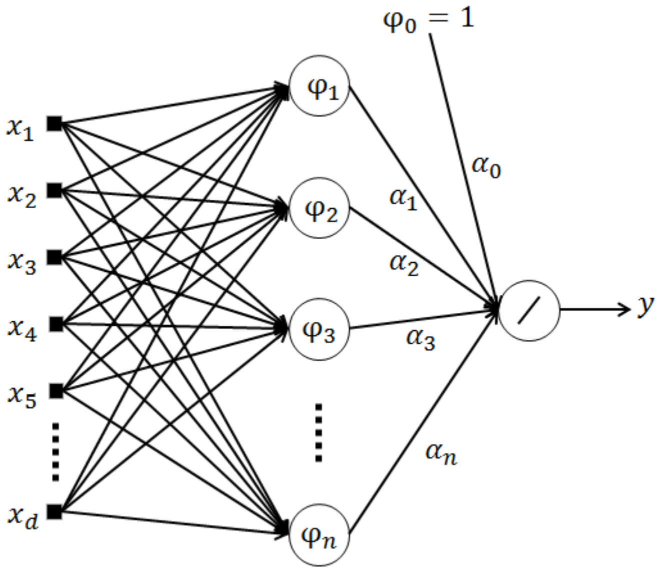


Figure 3. Topology of a RBFNN

The topology of a RBFNN is presented in Figure 3. The second layer units, known as neurons, are nonlinear functions of their vector inputs, given by,

$$\varphi_i(\boldsymbol{x}, \boldsymbol{c}_i, \sigma_i) = e^{\frac{-\|\boldsymbol{x}-\boldsymbol{c}_i\|^2}{2\sigma_i^2}}, \qquad \varphi_0 = 1, \tag{7}$$

where $\|\ \|$ denotes the Euclidean norm, and $\boldsymbol{c}_i$ and σ_i are, respectively, the location of the Gaussian function in the input space (denoted as centers) and its spread. The RBFNN output is given by:

$$y(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{C}, \boldsymbol{\sigma}) = \sum_{i=0}^{n} \alpha_i \varphi_i(\boldsymbol{x}, \boldsymbol{c}_i, \sigma_i) = \boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{C}, \boldsymbol{\sigma})\, \boldsymbol{\alpha}, \tag{8}$$

where n is the number of neurons and the α_i are the weights of the network output linear combiner.

As in this application the RBFNNs are used as classifiers, the output *y* is passed through a threshold function, in such a way that if *y*>0, the pixel is classified as abnormal, and normal otherwise.

In order to identify the best possible Radial Basis Functions neural network structure and parameters, this work uses a Multi-Objective Genetic Algorithm (MOGA) design framework, described in, for instance, [2, 26]. This method also allows us to handle multiple, possibly conflicting objectives. MOGA finds a non-dominated set of individuals through n number of generations and then selects preferable individuals from the non-dominated set. A solution is called non-dominated if none of the objectives can be improved in value without sacrificing some of the other objective values [27].

In order to be able to use the MOGA approach for finding the best possible model structure and its corresponding parameters, each possible structure for the NN needs to be formulated as a chromosome. To do that, the number of neurons in hidden layer is considered as the first component of the chromosome and the remaining components are the indices of features to be selected from a feature space. Figure 4 shows the topology of the chromosome. The algorithm starts its work by producing a pre-defined number of individuals as the first generation. The method then needs a mechanism to compare the individuals and select the best ones with respect to pre-defined objectives. The objectives, for the case at hand, can be selected from the set obj as described in (9).

$$obj = \left\{FN_s\, , FP_s\, , MC \;\middle|\; s = \{TE, TR\}\right\} \tag{9}$$

where FN_s is the number of False Negatives (i.e., those abnormal pixels that are wrongly classified as normal); FP_s is the number of False Positives (i.e., those normal pixels that are wrongly classified as abnormal) and MC stands for the Model Complexity. TE and TR represent Test and Training sets respectively. The formula for calculating Model Complexity is given in (10).

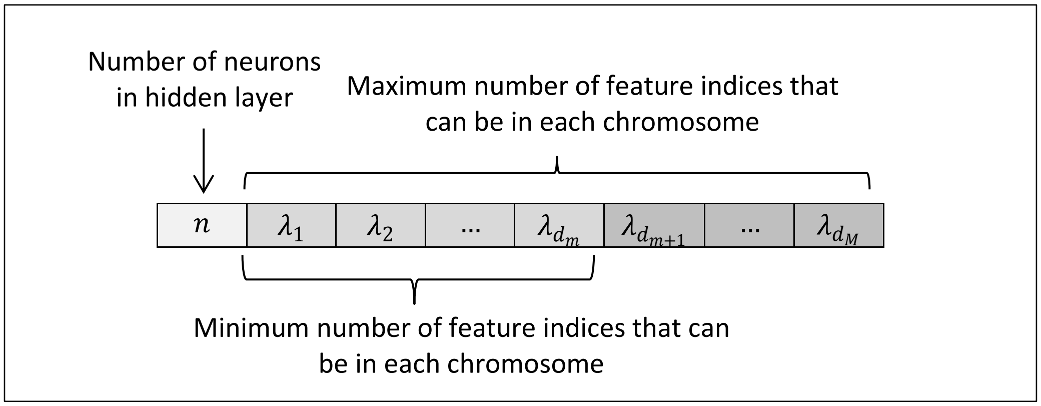

Number of neurons in hidden layer
Maximum number of feature indices that can be in each chromosome
n
λ_1
λ_2
...
λ_{d_m}
$\lambda_{d_{m+1}}$
...
λ_{d_M}
Minimum number of feature indices that can be in each chromosome

Figure 4. The topology of the chromosome

$$MC = (Number\ of\ input\ features + 1) \times Number\ of\ neurons\ in\ hidden\ layer \tag{10}$$

For evaluating the individuals in one generation, each NN model is trained with the provided training dataset (i.e., using the features whose indices are depicted in chromosome). The Levenberg-Marquardt (LM) algorithm [28, 29], with a formulation that exploits the linear-nonlinear separability of the NN parameters [30, 31] is employed for training due to its higher accuracy and convergence rate. Training is terminated if an user-specified number of iterations is reached, or the performance on a test set reaches a minimum (a procedure known as early-stopping [32]).

Since the result of gradient-based methods, such as LM, depends on the model initial parameters' values, for each individual in the current generation, the training procedure is repeated α times. Within these, the best result is picked up for determining the parameters of the individual (i.e., the centres, spreads and weights in RBFNNs). In MOGA, there are $d + 2$ different ways for identifying which training trial is the best one, considering d as the number of the objectives. The first strategy is to select the training trial which has minimized all objectives better than the others. In other words, if we consider a d dimensional space, the one whose Euclidean distance from the origin is the smallest, will be considered as the best. The green arrow in Figure 5 indicates this situation for $d = 2$. In the second strategy, the average of the objective values for all training trials is calculated and then the trial whose value is the closest to this average will be selected as the best one (i.e., the red arrow in Figure 5).

The other d strategies are to select the training trial which minimized the i^{th} objective ($i = 1,2,\cdots,d$) better than the other trials. As an example, the yellow and blue arrows in Figure 5 are the training trials which minimized objective1 and objective 2, respectively.

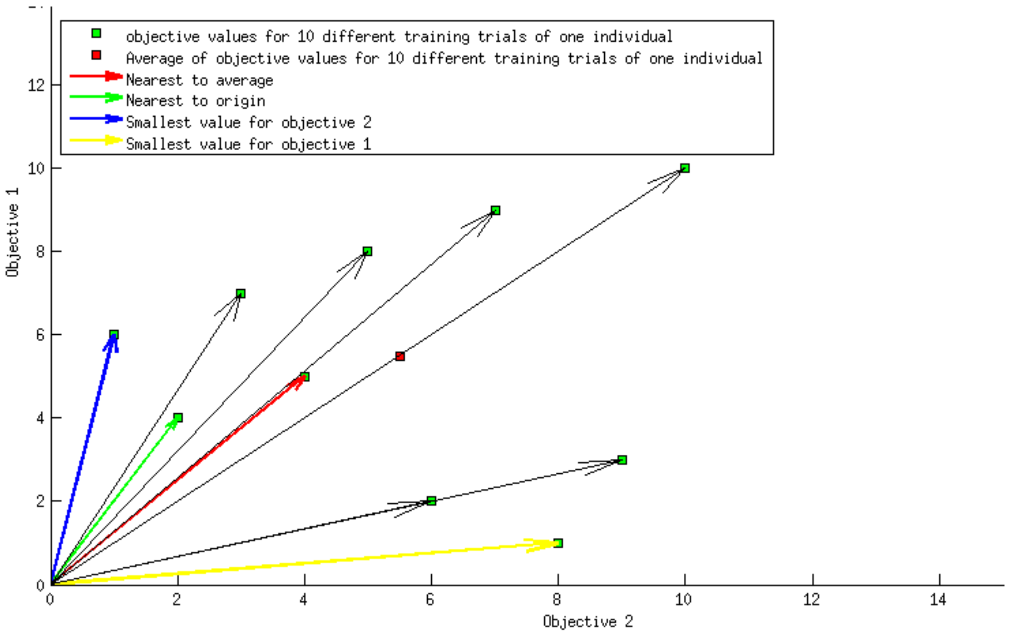


Figure 5. Four different strategies for identifying the best training trial within α=10 training trials

Having trained each individual, we are now able to assign a fitness value which reflects the individual's quality. MOGA uses a Pareto-based fitness assignment approach, which was first proposed by Goldberg and then modified by Fonseca and Fleming [33]. In this method, the individuals are ranked according to the number of individuals by which they are dominated. For example, if an individual is non-dominated, its corresponding rank is 0 and if an individual is dominated by three other individuals, its corresponding rank will be 3. Figure 6 visualizes the Pareto ranking notion.

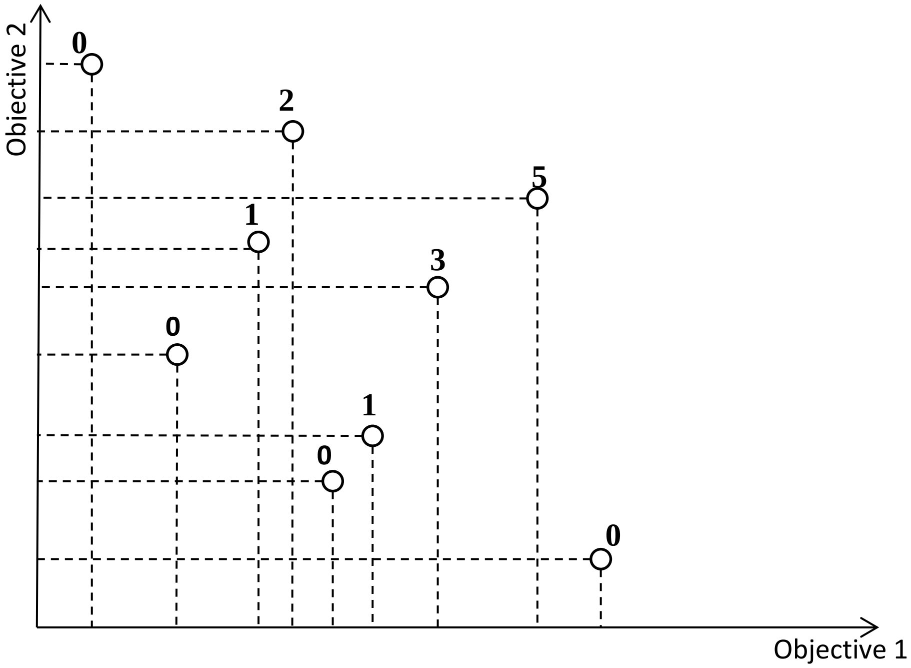


Figure 6. Pareto ranking [26, 34]

If there exists any preference such as assigning different priorities to each objective or defining a desired level of performance for each objective (i.e., restrictions), the ranking technique is slightly modified to take the restrictions and priorities into account. Suppose that c_1 and c_2 are the corresponding restrictions of objectives 1 and 2. In the case that both objectives have the same priorities, the individuals who satisfied the restrictions are assigned a rank equal to the number of individuals by which they are dominated. The individuals which do not meet the restrictions are penalized by assigning a higher rank. Figure 7 illustrates this situation.

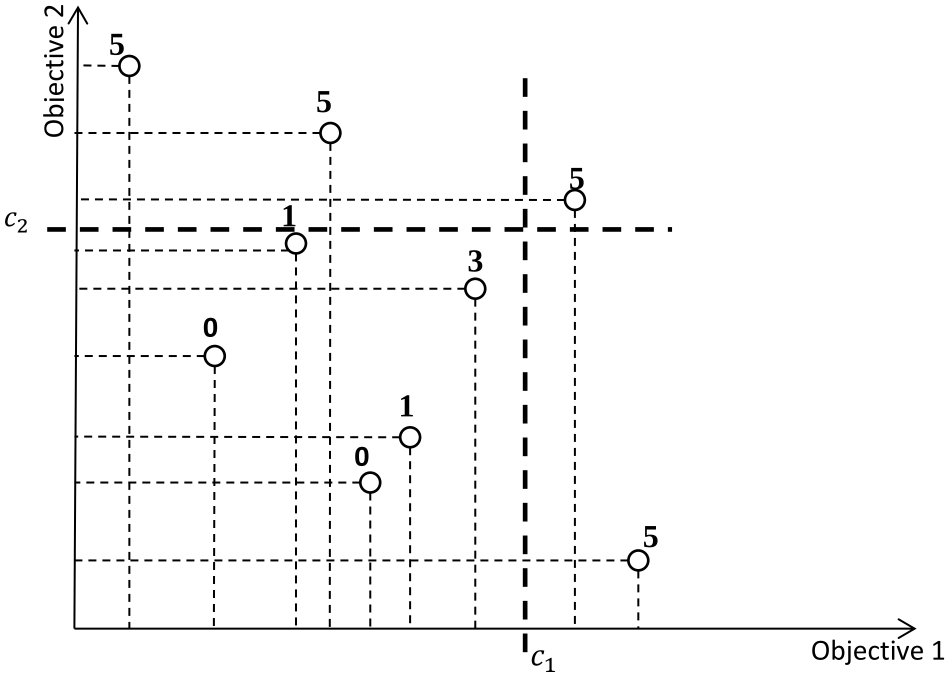


Figure 7. Pareto ranking in the case that both objectives have equal priorities. Both objectives should meet the defined restrictions [27, 33].

Figure 8 illustrates a situation in which objective 2 has a higher priority than objective 1. In this case, individuals which do not meet restriction c_2 are the worst, independently of their performance according to objective 1. Once c_2 is met, individuals are ranked based on how well they optimized objective 1 [27, 33].

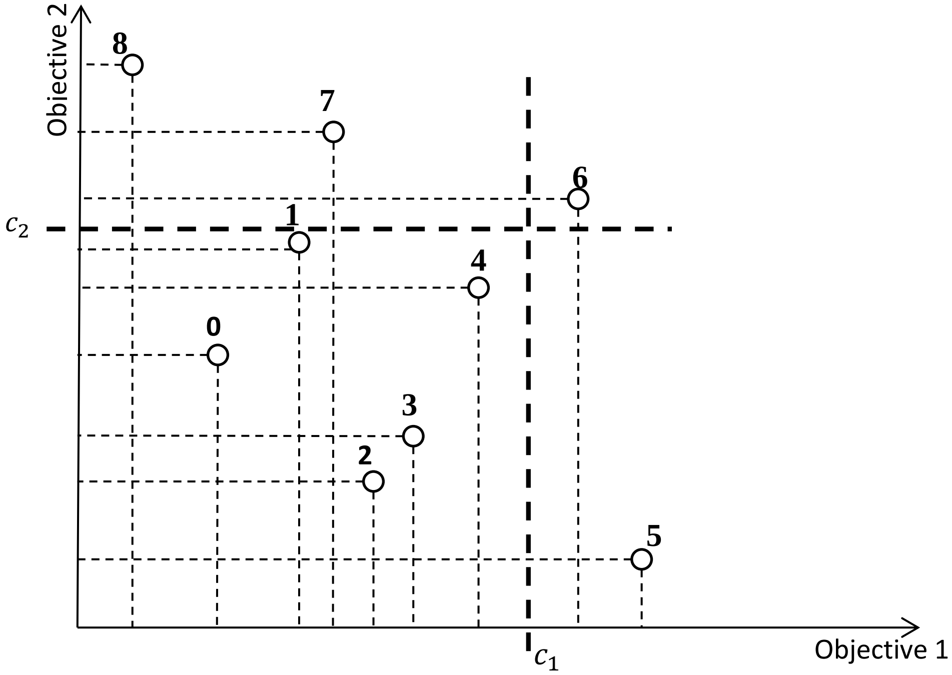


Figure 8. Pareto ranking in the case that objective 2 has higher priority than objective 1. Both objectives should meet the defined restrictions [27, 33].

Having ranked the individuals, MOGA assigns a fitness value to each individual based on its corresponding rank. To do that, the individuals are sorted based on the ranks and the fitness is assigned by interpolating from the best individual (i.e., rank=0) to the worst according to a linear or exponential function. Finally, a single value of fitness is calculated for the individuals with the same rank by the means of averaging. Assigning the average value to those with the same rank will guarantee the same probability of being selected as the parent of next generation [33-35].

The mating procedure uses the fitness values to generate a new population, ensuring that the individuals with higher fitness have a larger probability of breeding. To generate a new population, a small percentage of random immigrant from the previous generation is also introduced into the population, which makes the genetic algorithm more likely to recover information lost through selection and thus, from genetic drift [33]. The pairs selected for mating exchange part of their chromosome (i.e., based on the given probability crossover rate), to produce two offsprings for each pair, in the recombination phase. Parent recombination is done in a way that the offspring respect the maximum model length. The resulting offspring may be longer, shorter or equally sized as their parents. Once the new population is generated after recombination, mutation is applied to randomly selected individuals. The mutation operator is implemented by three basic operations: substitution, deletion and addition of one element. The number of neurons is mutated, with a given probability, by adding or subtracting one neuron to the model, verifying boundary conditions such that no neural network can have fewer or more neurons than pre-specified values. Each model input term in the chromosome is tested and, with a given probability, is either replaced by a new term not in the model, or deleted. Finally, a new term may be appended to the chromosome.

In each MOGA iteration, as shown in Figure 9, the non-dominated set is updated based on the individuals in current generation. It is expected that, after a sufficient number of generations, the population has evolved to achieve a non-dominated set which is not going to be altered; in this stage, the user must select the best model, among the final non-dominated set.

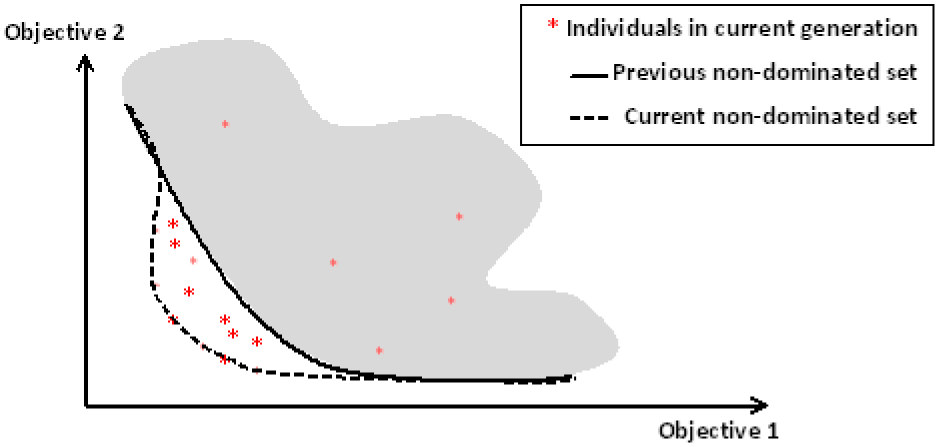

Objective 2
* Individuals in current generation
Previous non-dominated set
Current non-dominated set
Objective 1

Figure 9. The update of the non-dominated set on arrival of new points. The gray area denotes all generated models up to the current generation.

6. Experimental results

6.1. Constructing the input dataset for MOGA

Using the MOGA approach, the system must train a considerable amount of RBFNN models to be able to construct the final non-dominated set (please recall that the training process is done $\boldsymbol{\alpha}$=10 times for each chromosome). As a result, in practice, some constraints should be imposed on the size of the datasets that will be provided to MOGA, otherwise the process would not be finished in a reasonable time. As mentioned in section 3, we have 1,867,602 pixels (hereby called as $\boldsymbol{BIG\_DS}$) whose status (i.e., normal or abnormal) is already determined by the Neuroradiologist. Among these pixels 1,802,816 are normal (96.53% of the data samples) and 64,786 are abnormal (3.47% of the data samples). Hence, $\boldsymbol{BIG\_DS}$ is an imbalanced dataset whose size is $\mathbf{1,867,602 \times 52}$ (i.e., 51 features and 1 target column). To enable MOGA to generate models applicable to the whole range of data where the classifier is going to be used, we included all convex points [36] of $\boldsymbol{BIG\_DS}$ into the training set. To obtain the convex points, the Approxhull algorithm [37, 38] is used, resulting in 13023 samples, among which 11732 were normal and 1291 abnormal. The convex points along with 6977 random data samples (50% normal and 50% abnormal) constitute our training set whose size is 20,000. After excluding the training data samples from $\boldsymbol{BIG\_DS}$, 6666 random data samples were selected as a test set, and additional 6666 random data samples as a validation set. In both test and validation sets 50% of data samples were normal and 50% were abnormal. As a result, the input dataset for MOGA, hereafter called $\boldsymbol{MOGA\_DS}$ has 33,332 data samples including 60% training, 20% test and 20% validation data samples. $\boldsymbol{MOGA\_DS}$ is normalized between [-1, 1] before being passed to MOGA, since this process reduces the chance of encountering numerical problems in the training.

A flowchart, illustrating the different steps carried out for designing an RBFNN classifier for CVA detection using MOGA, is shown in Figure 10.

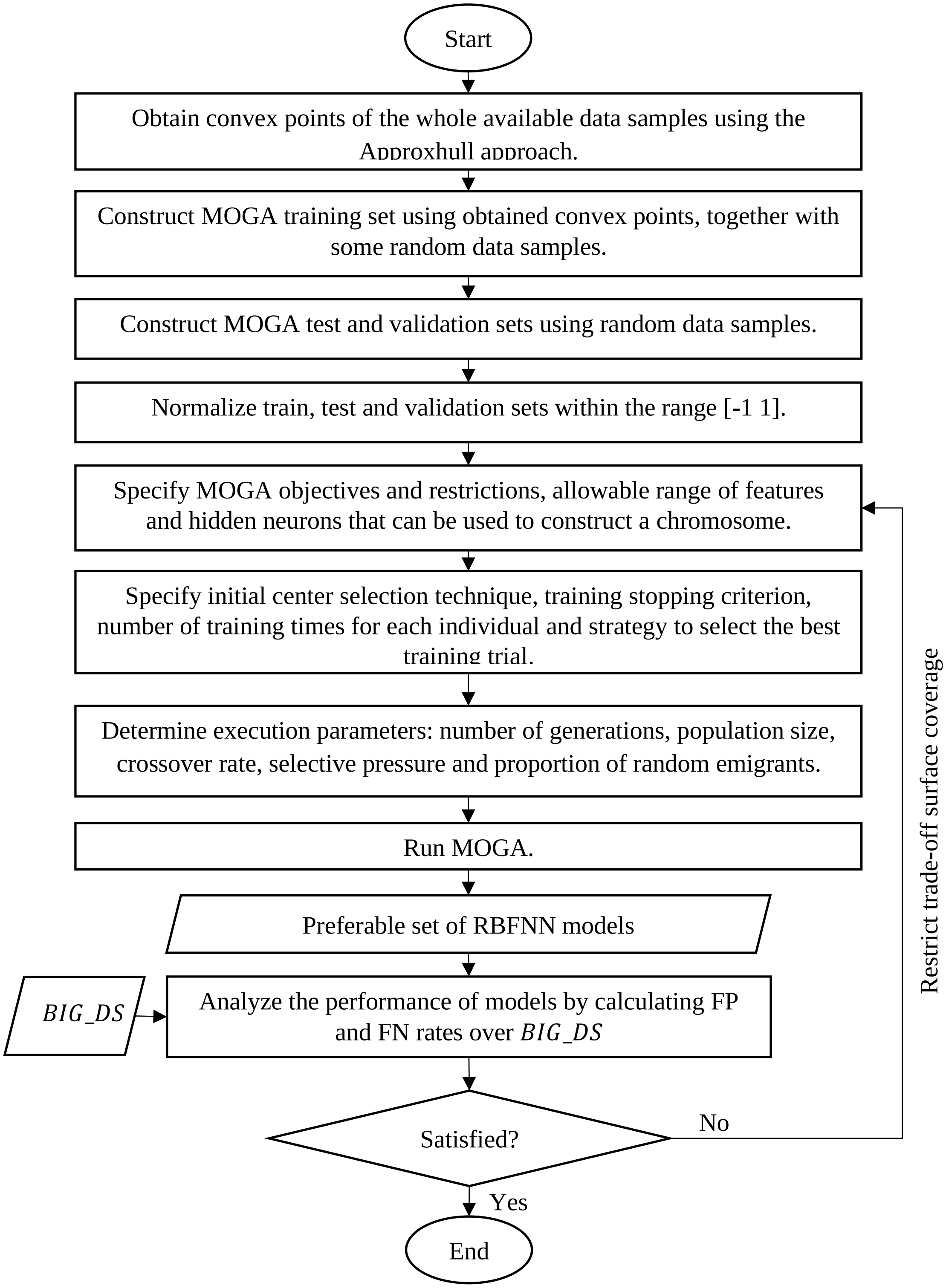

Start
Obtain convex points of the whole available data samples using the Approxhull approach.
Construct MOGA training set using obtained convex points, together with some random data samples.
Construct MOGA test and validation sets using random data samples.
Normalize train, test and validation sets within the range [-1 1].
Specify MOGA objectives and restrictions, allowable range of features and hidden neurons that can be used to construct a chromosome.
Specify initial center selection technique, training stopping criterion, number of training times for each individual and strategy to select the best training trial.
Determine execution parameters: number of generations, population size, crossover rate, selective pressure and proportion of random emigrants.
Run MOGA.
Preferable set of RBFNN models
BIG_DS
Analyze the performance of models by calculating FP and FN rates over BIG_DS
Satisfied?
No
Restrict trade-off surface coverage
Yes
End

Figure 10. Different steps carried out for designing an RBFNN classifier for CVA detection using MOGA

6.2 Experiment formulations

To identify the best possible RBFNN models two scenarios were conducted whose objectives are shown in Table II.

Table II. Objectives for the MOGA experiments

Exp.	**Objectives**
1	$FN_{TR}, FP_{TR}, FN_{TE}, FP_{TE}, MC$
2	$FN_{TR} < 129, FP_{TR} < 121, FN_{TE}, FP_{TE}, MC$

For both experiments, the system was allowed to choose the number of neurons in the hidden layer and the number of input features from the ranges [2,30] and [1,30], respectively. The number of generations and number of individuals in each generation were both set to 100. Early stopping with a maximum number of 100 iterations was used as a termination criterion for the training of each individual. The number of training trials for each individual, α, was set to 10 and the nearest to the origin strategy was used to select the best training trial. The proportion of random immigrants was 10%, the selective pressure was set to 2 and the crossover rate to 0.7.

The only difference between the two experiments is that restrictions were applied to the FN_{TR} and FP_{TR} objectives in the second experiment, based on the results obtained from the first experiment. To select the best model of experiment 1, we evaluated all non-dominated models on BIG_DS and then picked the model whose number of False Positives (FP) and False Negatives (FN) on BIG_{DS} were minimum.

406 non-dominated models were obtained as result of experiment 1 (since there are no restrictions on the objectives of this experiment, its preferable set is the same as the non-dominated set). Table III shows the Minimum, Average and Maximum FP and FN rates, as well as the model complexity of the non-dominated models of experiment 1. In the following Tables TR, TE, V and MC denote the training, test, validation sets, and the model complexity, respectively. Moreover, FD is the number of False Detections ($FP + FN$).

Table III. Min, Avg. and Max false positive and false negative rates as well as model complexity of 406 non-dominated models obtained in experiment 1.

	TR_{MOGA_DS}			TE_{MOGA_DS}			V_{MOGA_DS}			BIG_DS			MC
	FP (%)	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	
Min.	0	1.86	1.08	0	1.80	2.39	0	2.28	2.91	0	2.20	2.33	6
Avg.	2.13	23.41	7.21	3.83	21.50	12.67	4.16	21.60	12.88	4.09	21.78	4.71	199.8
Max.	8.47	100	24.16	12.27	100	50.03	13.47	100	50	12.49	100	12.74	900

Table IV shows the models whose False Positive and False Negative rates are less than 3% in BIG_DS. As it can be seen, both models have an equal percentage of FP (i.e., 2.96%) within BIG_DS but the FN percentage of model 1371 within BIG_DS is slightly smaller than that of model 6009. Hence, the statistics of model 1371, shown in Table V, were used as restrictions on experiment 2.

Table IV. Models of experiment 1 whose False Positive and False Negative rates are less than 3% in BIG_DS

Model No.	TR_{MOGA_DS}			TE_{MOGA_DS}			V_{MOGA_DS}			BIG_DS			MC
	FP (%)	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	
1371	0.80	2.70	1.25	2.43	2.58	2.51	3.27	3.24	3.26	2.96	2.88	2.96	702
6009	0.74	2.53	1.17	2.85	2.46	2.66	3.36	3.06	3.21	2.96	2.89	2.96	870

Table V. Statistics of model 1371.

Model No.	TR_{MOGA_DS}			TE_{MOGA_DS}			V_{MOGA_DS}			BIG_DS		
	FP	***FN***	***FD***	***FP***	***FN***	***FD***	***FP***	***FN***	***FD***	***FP***	***FN***	***FD***
1371	121	129	250	81	86	167	109	108	217	53442	1868	55310

Experiment 2 resulted in 281 non-dominated models from which 69 models are in the preferable set. Table VI shows the Minimum, Average and Maximum FP and FN rates as well as the model complexity over this set. Table VII shows the preferable models of experiment 2 whose FP and FN rates are less than 2.6% in BIG_DS. Analyzing the results of Table VII, one can see that employing restrictions on FN_{TR} and FP_{TR} resulted in models with smaller number of false detections in all sets, including in BIG_DS. Among the models in Table VII, model 3726 has the minimum percentage of FP and model 3055 has the minimum percentage of FN on BIG_DS.

Table VI. Min, Avg. and Max false positive and false negative rates as well as model complexity of 69 models in the preferable set of experiment 2.

	$\boldsymbol{TR_{MOGA\_DS}}$			$\boldsymbol{TE_{MOGA\_DS}}$			$\boldsymbol{V_{MOGA\_DS}}$			$\boldsymbol{BIG\_DS}$			$\boldsymbol{MC}$
	FP (%)	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	
Min.	0.49	1.38	0.77	2.19	2.06	2.36	2.24	1.75	2.05	2.40	1.98	2.40	750
Avg.	0.60	1.90	0.89	2.76	2.65	2.71	2.71	2.45	2.58	2.78	2.43	2.76	862.3
Max.	0.67	2.51	1.04	3.37	3.37	3.25	3.24	2.97	2.97	3.20	2.91	3.17	900

Table VII. Preferable models of experiment 2 whose false positive and false negative rates are less than 2.6% in BIG_DS

Model No.	$\boldsymbol{TR_{MOGA\_DS}}$			$\boldsymbol{TE_{MOGA\_DS}}$			$\boldsymbol{V_{MOGA\_DS}}$			$\boldsymbol{BIG\_DS}$			$\boldsymbol{MC}$
	FP (%)	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	
3726	0.60	1.79	0.87	2.27	2.90	2.58	2.58	2.37	2.48	**2.40**	2.34	2.40	870
4812	0.60	1.87	0.89	2.71	2.30	2.50	2.61	2.71	2.66	2.60	2.43	2.59	900
3863	0.59	1.52	0.80	2.32	2.71	2.52	2.43	2.56	2.49	2.55	2.45	2.55	900
3055	0.50	1.73	0.77	2.71	2.84	2.78	2.43	2.09	2.26	2.56	**2.31**	2.55	900

6.3 Ensemble of models in the preferable set of experiment 2

Having selected a model (model 3726 shown in Table VII) with acceptable rates of specificity 97.60% (i.e., 2.40 % FP rate) and sensitivity 97.66% (i.e., 2.34% FN rate) at pixel level, an ensemble of the preferable models obtained in experiment 2 was also considered as a classifier. Each data sample is fed to all 69 preferable models and then a majority vote determines whether the pixel is considered normal or abnormal. Table VIII shows the results obtained on $MOGA\_DS$ and BIG_DS. Comparing the results with the ones obtained from model 3726 in Table VII, it can be seen that 0.41% and 0.56% reductions could be obtained in the FP and FN rates over BIG_DS, respectively. Hence, the ensemble approach achieves a specificity of 98.01% (i.e., 1.99 % FP rate) and a sensitivity of 98.22% (i.e., 1.78% FN rate) at pixel level over BIG_DS.

Table VIII. Results of the ensemble of preferable models of experiment 2 on $MOGA\_DS$ and BIG_DS

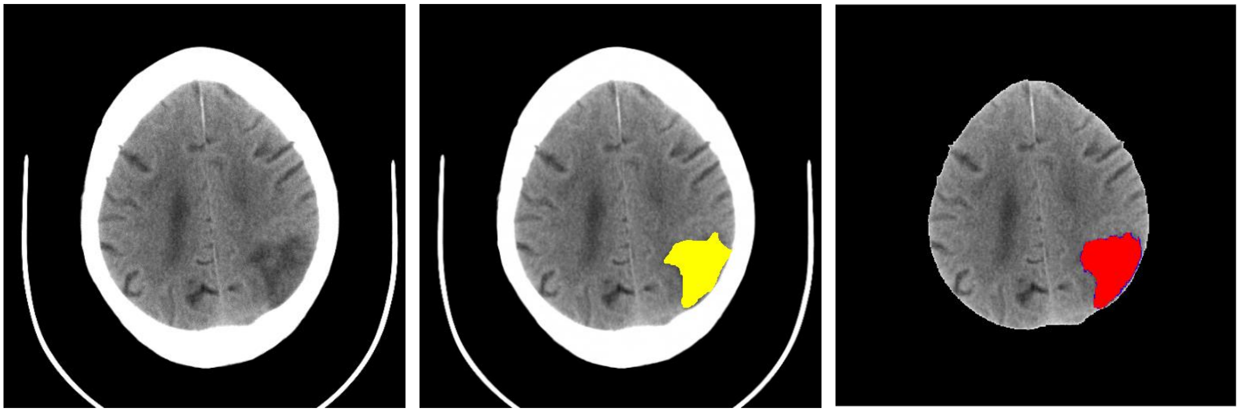
Ensemble of preferable models in experiment 2	$\boldsymbol{TR_{MOGA\_DS}}$			$\boldsymbol{TE_{MOGA\_DS}}$			$\boldsymbol{V_{MOGA\_DS}}$			$\boldsymbol{BIG\_DS}$		
	FP (%)	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***
	0.44	1.18	0.61	1.90	2.03	1.96	1.93	1.74	1.83	**1.99**	**1.78**	**1.99**

6.4 Visualizing abnormal regions in CT images using the ensemble of preferable models

Figure 11 shows the results of applying the ensemble of preferable models on some CT images, where the output images of the classifier were marked with different colors, depending on the classifier output for each tested pixel. The color code is shown in Table IX.

Table IX. Colour code used for marking pixels based on the percentage of preferable models with a positive output

Percentage of preferable models with a positive output	Colour code	Description
[66% 100%]	Red	Clear presence of pathology
[50% 66%)	Blue	Cannot decide whether the pixel is normal or abnormal
[0% 50%)	---	Clear absence of pathology



(a) (b) (c)

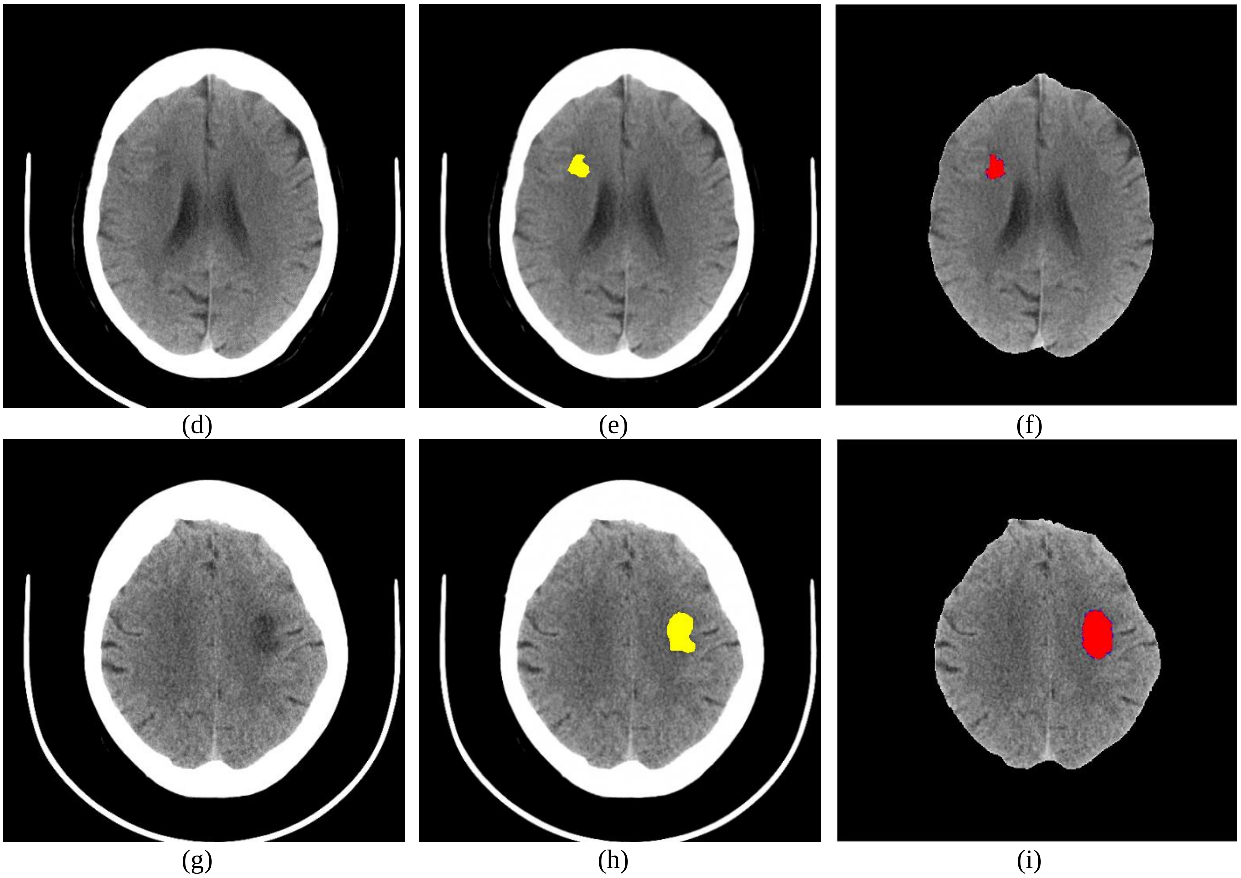


Figure 11. The result of applying the ensemble of preferable models on CT images. (a), (d) and (g) are the original images. (b), (e) and (h) are marked by the Neuroradialogist. (c), (f) and (i) are marked by the classifier.

6.5 Features use

To understand which features are the most frequent in the preferable models of experiment 2, the relative frequency of each feature fi within the 69 models is shown in Figure 12. One can see that, among the allowable 30 features within the 51 features considered, features $\{f2, f4, f5, f7, f12, f33, f41, f42, f44, f45\}$ are the ones that have been employed in more than 80% of the models. Among this set, features $\{f2, f4, f5, f7, f12, f41\}$ belong to the set of first order statistics, feature $f33$ is a second order statistic and features $\{f42, f44, f45\}$ are symmetry features.

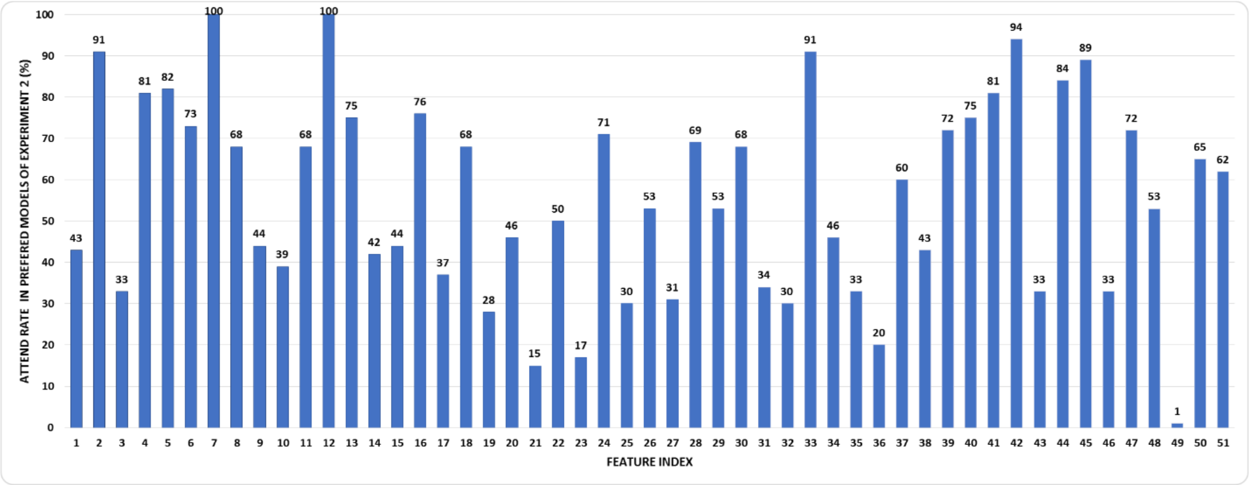


Figure 12. Relative frequency of each feature in the preferable models of experiment 2.

7. Discussion

As was shown before, the proposed approach achieves an excellent performance, both in terms of sensitivity and specificity. Moreover, as it is a pixel-based classifier, lesion contours are obtained, very close to the ones marked by the Neuroradiologist. A comparison of this approach with other works is, however, difficult as, to the best of our knowledge, there is no alternative system that uses a pixel-based classification.

As Support Vector Machines (SVM) are frequently used in classification purposes, in the next subsection we change our model, a MOGA-designed RBFNN to a SVM, and compare its use as a pixel-based classifier. The subsequent subsection will compare the proposed approach with three existing alternatives found in the literature. As pointed out before, these systems work at a slice level, which means that they can identify lesions but not draw their contours.

7.1 Comparing MOGA RBF classifiers with Support Vector Machines

In order to compare the obtained results with a SVM [39], the MATLAB SVM tool with Gaussian RBF (Radial Basis Function) kernel was used. For determining the best penalty parameter (C) and the spread, 121 possible combinations obtained by selecting 2 values from the set {0.003, 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300} were used for SVM training, and the combination (C=3, spread=1) whose error on the test set was minimum, was selected. In this experiment, 69.8% of the data samples in the training set were considered as support vectors. Table X shows the FP and FN rates when this SVM was applied.

Comparing the results with the ones obtained with the ensemble of models, shown in Table VIII, and also with model 3726, shown in Table VII, one can see that even with a huge complexity of the SVM model (139,600 support vectors), its FP and FN rates in BIG_{DS} are not only higher than the ones obtained with the ensemble of preferable models, but also the ones achieved by model

3726. Notice that a SVM model, with Gaussian Kernel can be considered a RBFNN model, where the centers of the Gaussians are the support vectors, and with a common spread to all the neurons. In this case, all the features (51) were considered as inputs and 13960 support vectors were employed. This is translated into a complexity of 711,960 parameters, determined by the SVM algorithm. This figure should be compared with a complexity of 870 (around 0.1%), for model 3726 in Table VII.

Table X. FP and FN rates using SVM

TR_{MOGA_DS}			TE_{MOGA_DS}			V_{MOGA_DS}			BIG_DS		
FP (%)	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***	***FP (%)***	***FN (%)***	***FD (%)***
0.16	0	0.13	2.6	2.42	2.51	2.32	2.26	2.29	**2.5**	**2.37**	2.5

7.2 Comparing the results obtained with other approaches

The authors in [40] presented a Computer Aided Detection (CAD) method for early detection of CVAs from CT images where, in the same way as this work, in a preprocessing phase artifacts are removed and tilted CT images are realigned. In order to find the regions that have higher probability of being considered lesions, a Circular Adaptive Region Of Interest (CAROI) algorithm is applied on each CT slice, which aims to draw a circular border around areas with sudden change of intensity values. Each circular region is then compared with its corresponding region in the other side of the brain using the Pearson correlation coefficient. Those circular areas which have the smallest PCC values are selected for further investigation. Eight second order features are calculated from the GLCM matrix of previously selected circular regions and are passed to a 3-layer feed-forward back propagation neural network which was trained using 10 normal and 20 abnormal cases in a round robin (leave-one-out) fashion. The output of neural network identifies whether the circular region is a lesion or not.

In order to evaluate their CAD system, 31 positive cases containing 82 ischemic strokes (39 acute and 43 chronic) were used as validation set. A sensitivity of 76.92% (i.e., 30/39 lesion areas correctly detected) for acute ischemic strokes and a sensitivity of 90.70% (i.e., 39/43 lesion areas correctly detected) for chronic strokes were reported. This gives a total sensitivity of 84.14% (i.e., $\frac{30+39}{82} \times 100$).

In spite of the differences of this approach and ours, in order to be able to compare the accuracy obtained in terms of lesions sensitivity, this measure has been calculated. A total number of 35 ischemic lesions within 150 CT images were marked by our collaborating Neuroradiologist. The ensemble of preferable models in experiment 2 detected 30 lesions correctly, which is translated in a sensitivity of 85.71%, slightly higher than approach [40].

The authors in [41] developed a CAD system for detecting hemorrhagic strokes in CT images. After removing the artifacts and realigning the tilted images, the hemorrhagic areas are segmented by employing a threshold on the pixels' intensity values. To detect the edema regions, a higher contrast ratio of a given CT image is firstly obtained using a local histogram equalization. A thresholding method is then applied to segment the edema region from the normal tissue. The accuracy of the CAD system is evaluated by comparing the area of bleeding region (ABR) and edema region (AER) that are detected by the CAD and the ones that are marked by the doctor using data from 8 spontaneous hemorrhagic stroke patients. It is reported that the average difference of ABRs is 8.8%, and the average of the degree of coincidence is 86.4%, while the average difference of AERs is 14.1%, with an average of degree of coincidence of 77.4%.

The results obtained by this approach cannot be exactly compared with the approach presented here, as [41] deals with hemorrhagic strokes, which typically are much easier to detect and mark than ischemic strokes. In spite of that, the average difference of the areas as well as the average degree of coincidence have been computed for the cases presented here, for the lesions both marked by the doctor and detected by our system. The average difference is 11.4%, and the average degree of coincidence is 88.6%. These figures are better than the values obtained for AER, in approach [41].

The authors in [42] utilize a combination of 2D and 3D Convolutional Neural Networks (CNN) to cluster brain CT images into 3 groups: Alzheimer's disease, lesion and normal ageing. The best classification accuracy rates using the proposed CNN architecture are 85.2%, 80% and 95.3% for the classes of Alzheimer's disease, lesion and normal, respectively, with an average of 87.6%.

To be able to compare the classification accuracy rates of our work with the ones in [42], we calculated this metric in a CT slice level, by labeling each CT slice as normal or abnormal (i.e., having one or more lesions). Our system was able to correctly identify all 24 CT slices that had lesions within which translates into 100% classification accuracy rate for the abnormal group. Among the remaining 126 normal CT slices, our system identified small false lesions within 7 CT slices, which means that we obtained a 94.4 % classification accuracy rate for the normal group.

8. Conclusions

In this work, a RBFNN based system for automatic identification of CVA through the analysis of brain CT images is presented. Considering a set of 51 features, the MOGA design framework was employed to find the best possible RBFNN structure and its corresponding parameters. Two experiments were conducted in MOGA. The best result is obtained from an ensemble of preferable models of experiment 2, where the FN_{TR} and FP_{TR} objectives were restricted based on the results obtained by the best model from the first experiment. Values of specificity of 98.01% (i.e., 1.99 % FP) and sensitivity of 98.22% (i.e., 1.78% FN) were obtained at pixel level, in a set of 150 CT slices (1,867,602 pixels).

Comparing the classification results with SVM over BIG_DS, despite the huge complexity of the SVM model, the accuracy of the selected model in experiment 2, as well as the ensemble of preferable models, are superior to that of SVM model.

The present approach compares also favorably with other similar published approaches, achieving improved sensitivity at lesion level than [40], better average difference and degree of coincidence than [41], as well as superior classification accuracy rate than [42]. It should be stressed than none of these methods are able to draw the lesion(s) contour(s), as it is achieved by the proposed approach.

As the number of abnormal pixels is much smaller than the number of normal pixels in the existing dataset, at the present stage the classifier is able to detect the great majority of the lesions, but sometimes will identify false lesions. Current research is tackling this problem. Additionally, as the proposed classifier was designed and tested only with CT images including ischemic CVAs, we consider enlarging the CT database to include other brain lesions, and apply the same methodology to design an additional classifier capable of discriminating brain lesions with similar image patterns.

Acknowledgments

The authors would like to acknowledge the support of FCT, through IDMEC, under LAETA, project UID/EMS/50022/2013 and Dr. Luis Cerqueira, from Centro Hospitalar de Lisboa Central, Portugal, for marking the exams.

References

[1] A. S. Go, D. Mozaffarian, V. L. Roger, E. J. Benjamin, J. D. Berry, W. B. Borden*, et al.*, "Heart disease and stroke statistics--2013 update: a report from the American Heart Association," *Circulation,* vol. 127, pp. e6-e245, Jan 1 2013.

[2] P. M. Ferreira and A. E. Ruano, "Evolutionary Multiobjective Neural Network Models Identification: Evolving Task-Optimised Models," *New Advances in Intelligent Signal Processing,* vol. 372, pp. 21-53, 2011.

[3] E. Hajimani, M. G. Ruano, and A. E. Ruano, "MOGA design for neural networks based system for automatic diagnosis of Cerebral Vascular Accidents," in *9th IEEE International Symposium on Intelligent Signal Processing (WISP)*, 2015, pp. 1-6.

[4] E. Hajimani, A. Ruano, and G. Ruano, "The Effect of Symmetry Features on Cerebral Vascular Accident Detection Accuracy," presented at the RecPad 2015, the 21th edition of the Portuguese Conference on Pattern Recognition, Faro, Portugal, 2015.

[5] X. Llado, A. Oliver, M. Cabezas, J. Freixenet, J. C. Vilanova, A. Quiles*, et al.*, "Segmentation of multiple sclerosis lesions in brain MRI: A review of automated approaches," *Information Sciences,* vol. 186, Mar 1 2012.

[6] T. Hachaj and M. R. Ogiela, "CAD system for automatic analysis of CT perfusion maps," *Opto-Electronics Review,* vol. 19, Mar 2011.

[7] L. E. Poh, V. Gupta, A. Johnson, R. Kazmierski, and W. L. Nowinski, "Automatic Segmentation of Ventricular Cerebrospinal Fluid from Ischemic Stroke CT Images," *Neuroinformatics,* vol. 10, Apr 2012.

[8] W. L. Nowinski, G. Qian, K. N. B. Prakash, Q. Hu, and A. Aziz, "Fast Talairach Transformation for magnetic resonance neuroimages," *Journal of Computer Assisted Tomography,* vol. 30, pp. 629-641, Jul-Aug 2006.

[9] N. Otsu, "A Threshold Selection Method From Gray-level Histogram," *IEEE Transactions on Systems, Man, and Cybernetics,* 1978.

[10] M. Gao and S. Chen, "Fully Automatic Segmentation of Brain Tumour in CT Images," *European Journal of Cancer,* vol. 47, Sep 2011.

[11] A. P. Nanthagopal and R. S. Rajamony, "Automatic classification of brain computed tomography images using wavelet-based statistical texture features," *Journal of Visualization,* vol. 15, pp. 363-372, Nov 2012.

[12] T. J. Devadas and R. Ganesan, "Analysis of CT Brain images using Radial Basis Function Neural Network," *Defence Science Journal,* vol. 62, Jul 2012.

[13] O. Freifeld, H. Greenspan, and J. Goldberger, "Lesion detection in noisy MR brain images using constrained GMM and active contours," *4th IEEE International Symposium on Biomedical Imaging : Macro to Nano, Vols 1-3,* pp. 596-599, 2007.

[14] H. Greenspan, A. Ruf, and J. Goldberger, "Constrained Gaussian mixture model framework for automatic segmentation of MR brain images," *IEEE Transactions on Medical Imaging,* vol. 25, pp. 1233-1245, Sep 2006.

[15] B. J. Bedell and P. A. Narayana, "Automatic segmentation of gadolinium-enhanced multiple sclerosis lesions," *Magnetic Resonance in Medicine,* vol. 39, pp. 935-940, Jun 1998.

[16] A. O. Boudraa, S. Mohammed, R. Dehak, Y. M. Zhu, C. Pachai, Y. G. Bao*, et al.*, "Automated segmentation of multiple sclerosis lesions in multispectral MR imaging using fuzzy clustering," *Computers in Biology and Medicine,* vol. 30, pp. 23-40, Jan 2000.

[17] E. Hajimani, C. A. Ruano, M. G. Ruano, and A. E. Ruano, "A software tool for intelligent CVA diagnosis by cerebral computerized tomography," in *8th IEEE International Symposium on Intelligent Signal Processing (WISP)*, 2013, pp. 103-108.

[18] X. Qi, A. Belle, S. Shandilya, W. Chen, C. Cockrell, Y. Tang*, et al.*, "Ideal Midline Detection Using Automated Processing of Brain CT Image," *Open Journal of Medical Imaging,* vol. 3, p. 9, 2013.

[19] L. He, Y. Chao, K. Suzuki, and K. Wu, "Fast connected-component labeling," *Pattern Recognition,* vol. 42, pp. 1977-1987, Sep 2009.

[20] A. Usinskas, R. A. Dobrovolskis, and B. F. Tomandl, "Ischemic stroke segmentation on CT images using joint features," *Informatica,* vol. 15, pp. 283-290, 2004.

[21] L. Ribeiro, A. E. Ruano, M. G. Ruano, and P. M. Ferreira, "Neural networks assisted diagnosis of ischemic CVA's through CT scan," *IEEE International Symposium on Intelligent Signal Processing, Conference Proceedings Book,* pp. 223-227, 2007.

[22] W. Chen, R. Smith, S.-Y. Ji, K. R. Ward, and K. Najarian, "Automated ventricular systems segmentation in brain CT images by combining low-level segmentation and

high-level template matching," *BMC Medical Informatics and Decision Making,* vol. 9, p. 14, 2009.

[23] Wei-Li Zhang and Xi-Zhao Wang, "Feature Extraction and Classification for Human Brain CT Images," in *International Conference on Machine Learning and Cybernetics*, 2007, pp. 1155-1159.

[24] R. M. Haralick, Shanmuga.K, and I. Dinstein, "Textural Features for Image Classification," *IEEE Transactions on Systems Man and Cybernetics,* vol. SMC3, pp. 610-621, 1973.

[25] D. A. Clausi, "An analysis of co-occurrence texture statistics as a function of grey level quantization," *Canadian Journal of Remote Sensing,* vol. 28, pp. 45-62, Feb 2002.

[26] C. A. Teixeira, M. G. Ruano, A. E. Ruano, and W. C. A. Pereira, "A soft-computing methodology for noninvasive time-spatial temperature estimation," *IEEE Transactions on Biomedical Engineering,* vol. 55, pp. 572-580, Feb 2008.

[27] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: A unified formulation," *IEEE Transactions on Systems Man and Cybernetics Part a-Systems and Humans,* vol. 28, pp. 26-37, Jan 1998.

[28] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal of the Society for Industrial and Applied Mathematics,* vol. 11, pp. 431-441, 1963.

[29] K. Levenberg, "A method for the solution of certain problems in least squares," *Quart. Applied Math.,* vol. 2, pp. 164-168, 1944.

[30] P. M. Ferreira and A. E. Ruano, "Exploiting the separability of linear and nonlinear parameters in radial basis function networks," *IEEE Proceedings on Adaptive Systems for Signal Processing, Communications, and Control Symposium,* pp. 321-326, 2000.

[31] A. E. Ruano and I. o. E. Engineers, *Intelligent Control Systems Using Computational Intelligence Techniques*: Institution of Engineering and Technology, 2005.

[32] S. S. Haykin, *Neural Networks: A Comprehensive Foundation*: Prentice Hall, 1999.

[33] C. M. M. d. Fonseca, "*Multiobjective genetic algorithms with application to control engineering problems*," PhD, Department of Automatic Control and Systems Engineering, University of Sheffield, 1995.

[34] C. M. Fonseca and P. J. Fleming, "Multiobjective Genetic Algorithms Made Easy: Selection, Sharing and Mating Restriction," presented at the Genetic Algorithms in Engineering Systems: Innovations and Applications, UK, 1995.

[35] C. A. D. Teixeira, "Soft-computing techniques applied to artificial tissue temperature estimation," PhD, FCT, University of Algarve, 2008.

[36] H. R. Khosravani, A. E. Ruano, and P. M. Ferreira, "A simple algorithm for convex hull determination in high dimensions," presented at the 8th International Symposium on Intelligent Signal Processing (WISP), Portugal, 2013.

[37] A. Ruano, H. R. Khosravani, and P. M. Ferreira, "A Randomized Approximation Convex Hull Algorithm for High Dimensions," *IFAC-PapersOnLine,* vol. 48, pp. 123 - 128, 2015.

[38] H. Khosravani, A. Ruano, and P. Ferreira, "A Convex Hull-based Data Selection Method for Data Driven Models," *Applied Soft Computing,* 2016 (in press).

[39] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning,* vol. 20, pp. 273-297, Sep 1995.

[40] K.-s. D. Ng, "Computer aided detection method for early detection of cerebrovascular accident," PhD, Dept. of Health Technology and Informatics, The Hong Kong Polytechnic University, The Hong Kong Polytechnic University, 2009.

[41] J.-G. Gan, Y.-W. Wang, J.-L. Su, L. Chan, and Ieee, "The Development of CAD system for Hemorrhagic Stroke in Computed Tomography Images," *2014 Ieee International Symposium on Bioelectronics and Bioinformatics (Isbb)*, 2014 2014.

[42] X. H. W. Gao, R. Hui, and Z. M. Tian, "Classification of CT brain images based on deep learning networks," *Computer Methods and Programs in Biomedicine*, vol. 138, pp. 49-56, Jan 2017.