Feature extraction with mixture gaussian for stroke classification

Williana L. S. Leite
Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará
Fortaleza, Brazil
williana.leite60@aluno.ifce.edu.br

Roger M. Sarmento
Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará
Fortaleza, Brazil
rogerms@ifce.edu.br

Carlos M. J. M. Dourado Junior
Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará
Fortaleza, Brazil
mauriciodourado@ifce.edu.br

***Abstract*—The stroke remains the second leading cause of death in the world. The stroke diagnosis is usually obtained by neuroimaging analysis, and among the main techniques, Computed Tomography (CT) is the most used. A quick diagnosis of stroke can generally contribute positively to the patient's recovery. CAD systems that analyze CT scan images become extremely important when diagnosis speed is a relevant factor for patient recovery. This work presents a new feature extractor using gaussian mixtures, called Mixture Gaussian Analysis of Brain Tissue Density (MGABTD). The MGABTD achieved accuracy and f1-score of 99.9%. The results demonstrated the effectiveness of the method in extracting features used to determine whether a CT scan is normal or shows an ischemic or hemorrhagic stroke.**

***Index Terms*—Stroke, Classification, Feature extractor, Computed tomography, Radiological density.**

I. Introduction

The Cerebrovascular Accident (CVA) or Stroke, is a cardiovascular disease that represents the sudden loss of neurological function from any cause [32]. According to Feigin et al. [10], strokes remain the second leading cause of death in the world. According to the World Health Organization (WHO), approximately 15 million people suffer a stroke in the world every year. Out of this contingent, 5 million die and another 5 million get permanently disabled. Despite the high mortality rate, if diagnosed quickly, treatment can prevent serious permanent injuries and even death. There are two types of stroke: The hemorrhagic stroke, caused by the rupture of a blood vessel, and the ischemic stroke, which takes place when the bloodstream gets interrupted [20].

The diagnosis can be made by means of the analysis of neuroimages. There are several available techniques, such as: Computed Tomography (CT) and Magnetic Resonance Imaging (MRI). MR images are of better quality than CT scans [17]. However, they require equipment that is usually available only in large hospitals, so CT scans are usually the choice. That said, and considering the importance of rapid diagnosis for the patient's recovery, systems that analyze CT scans to assist physicians in their decision-making concerning treatments for the disease are extremely relevant [1].

Computer-Aided Diagnosis (CAD) has been the focus of several research works, some factors that support the relevance of such systems are: improvement in diagnostic accuracy, reduction of time and the ability to deal with large amounts of information [31]. Among other modalities of CAD systems, diagnostic imaging has also shown satisfactory results for various pathologies [15], [21], [33].

A CT image is composed of 16 bits, and pixel intensities are measured on the Hounsfield Units (HU) scale, which measures radiodensity. With this scale, HU intervals are defined. They represent several common substances, such as white matter, which is represented by an interval ranging from 23HU to 34HU. However, some of these substances or pathologies have different or overlapping definitions in the literature, making the class intervals subjective, such as the definition of white matter and gray matter class (32-41 HU), values 32, 33 and 34HU belong simultaneously to two distinct classes [32].

In order to search for better results in the classification of stroke in CAD systems, this research proposes a new method for extracting features from CT images. The proposed method is an improvement of the Analysis of Brain Tissue Density (ABTD) [23] technique that uses percentages to minimize the subjectivity of brain tissue density intervals, and the new method uses a probability estimator. Sarmento et al. [26] obtained satisfactory results using the Parzen Window as a probability estimator in the classification of stroke by CT. Considering the lessons learned through experiments [2], the proposed method uses Gaussian Mixing as a probability estimator and will be called Mixture Gaussian Analysis of Brain Tissue Density (MGABTD). Considering the 3 classes: hemorrhagic stroke, ischemic stroke and normal, the new method presented satisfactory results when compared to the other extractors with an average accuracy of 99.9%, with the classifiers: Extremely Randomized Trees (Extra Tree), Support Vector Machine (SVM), K-Nearest Neighbors (KNN) and Random Forest (RF). The main contribution of this work is to make HU intervals less susceptible to subjectivity, creating a more reliable method using the Gaussian mixture estimation to measure the probability of each pixel belonging to a HU region.

II. Related Works

Over the years, several research works have been developed with the objective of making the diagnostic process more efficient and robust through computer systems [28].

Sarmento et al. [26] proposed a new method for feature extraction using the parzen window called PABTD, to estimate the probability of each pixel belonging to predefined ranges that delimit each of the classes using the HU scale. The configuration that obtained the best performance was using the SVM classifier with RBF, reaching an accuracy of 98.41%. In Vasconcelos et al. [30], a new extractor called ABTD-Adaptive was also proposed, which considers the neighborhood of pixels and class interval information to perform the extraction. The classifier with the best accuracy was the SVM kernel RBF with 98.13%.

In Kanchana and Menaka [16], the objective was the segmentation and classification of ischemic stroke. A new approach was employed to segment image regions into normal and abnormal. After segmentation, the region is sent to extractors: First Order Features (FOF), Gray Level Run Length Matrix (GLRLM) features, Gray Level Co-occurrence Matrix (GLCM) features, and HU Moments (HUM). Then, in order to select only the most important features, the feature ranking technique was used, with the ANOVA statistical method. The best accuracy was 99.79%, obtained in Random Forest (RF), and in the neural network. Badriyah et al. [3] also proposed the use of GLCM after image processing, and obtained an accuracy of 95.97% with RF.

Dorado et al. proposed 2 approaches for medical image classification using Convolutional Neural Networks (CNN). In the first [8], the authors used the medical bases for stroke and lung cancer to present the efficiency of the tool. Of the best-known CNN architectures, the one that obtained the best result was VGG16 together with KNN, achieving 100% accuracy. In the second [9], the authors proposed the use of the NASNet Large extractor and the Bayesian classifier for stroke detection, also reaching 100% accuracy. Following a similar approach, Chen et al. [5] proposed the use of CT without contrast, the combination that obtained the best result was extraction with Resnet50, reaching 98.7% accuracy, but the author warns of the delay in the feature extraction process.

The authors Gautam and Raman [11] also a new feature extractor: Local Neighborhood Pattern (LNP) to extract spatial and quantum information from images. The extraction is based on comparing diagonal pixels with the average of the entire intensity of the image, and the remaining pixels are compared with diagonal pixels on its left side. The best accuracy obtained was 82.65% with the Ensemble Bagged Trees (EBT), using 10-fold cross-validation. The same authors also proposed another format for classification strokes by using image fusion and RNC, with even better results, where they developed an RNC with a new 13-layer architecture, with a size much smaller than the most famous networks in the literature such as Alexnet, Resnet among others. It made classification much faster. The best accuracy obtained in one of the test sets was 98.77% [12].

Multimodal learning, with information fusion techniques in deep learning architectures, was also explored in the literature by researchers Roder et al. [25], where they proposed to apply Fourier transforms as input to the network in addition to the images, so that the network can learn simultaneously. The classifier used was the Restricted Boltzmann Machines (RBM), recognized for being simpler, it obtained the best accuracy of 99.72%. Lo et al. [18] proposed the use of deep CNN on non-contrast CT for the classification between healthy and stroke patients. With 1,254 images from 96 patients, the highest accuracy obtained was 97.12% with AlexNet.

III. Methodology

The steps of the proposed method for each CT of the database are: select the region of interest, extract the features with the MGABTD method and, finally, perform the training and projection of the classes: normal, hemorrhagic, or ischemic stroke. In the following sections, the main steps are presented in detail.

A. Image acquisition

The database consists of 473 brain CT images, divided into images of 174 healthy patient images, 157 ischemic strokes and 142 hemorrhagic strokes. The database was obtained with the support of Clínica Trajano Almeida - Diagnostic Imaging and Hospital do Coração [23]. For ethical reasons, the database does not provide any patient identification. The images do not have patterns or regular lighting, the structural features are different and have 512x512 pixels stored in DICOM format.

B. Selection of the region of interest

The inner part of the skull is the most important region on CT for the diagnosis of stroke. To segment this region of interest, the method presented in [26] was used. Segmentation is basically divided into:

- Load the image;
- Apply a mask that highlights only pixels with HU values between 0 and 120, inner region of the skull; [32]
- With skull pixels highlighted, a subtraction is made from the original image, thus removing the skull;
- Finally, the largest connected component is selected and then the intersection with the original image is made.

At the end of the process, the resulting image does not have the skull or any noise that could interfere with the further analysis of the proposed method.

C. Gaussian Mixture Model

A Gaussian Mixture Model (GMM), is a parametric probability density function, which represents a weighted sum of densities of the Gaussian components [24]. A GMM can be defined by:

$$p(x) = \sum_{i=1}^{K} w_i \; g(x|\mu_i, \Sigma_i) \tag{1}$$

Where x is a sample from the database, a D-dimensional vector. In this case, w_i is the mixture coefficient, that is,

the weight that each Gaussian component will have, and the K represents the number of components. A component is defined by a multivariate Gaussian, $g(x|\mu_i, \Sigma_i)$, where μ_i means the mean vector and Σ_i the covariance matrix of the i-th component. For classification, the Gaussian Mixture can be used as a probability estimator:

$$p(c_y|x) = \frac{p(x|c_y) * p(c_y)}{\sum_{y=1}^{C} p(x|c_y) * p(c_y)} \quad (2)$$

Where $p(c_y|x)$ represents the probability that the sample x belongs to class c_y, where $y \in \{1, ..., C\}$, where C is the total of classes. Here, $p(x|c_y)$ represents the a priori probability of x belonging to c_y and $p(c_y)$ represents the probability density function obtained in the GMM, described in equation 1. And the denominator represents a normalization factor, to ensure that the probabilities of each class when added together add up to 1. To determine the class of the sample, just calculate the probability of belonging to each of the classes and select the class that has the highest probability.

There are several methods to estimate the parameters of a GMM [19], the most used method in the literature is the maximum likelihood estimation, which uses training data for maximization. The parameters can be obtained iteratively using the algorithm Expectation-Maximization (EM) or the Maximum A Posteriori (MAP) [7].

The EM algorithm, used in this research, is used to define the means, covariance matrices and mixing coefficients of each of the components of each class, the algorithm is divided into 2 steps: E-Step and M-Step. The E-step is responsible for calculating the responsibility, which is the probability of a sample belonging to a component, considering the current value of the parameters (means, covariance matrices and mixture coefficient) of the iteration. And the M-step is responsible for updating the parameters in each iteration. After t iterations the EM algorithm places the components for each class into position.

D. Mixture Gaussian Analysis of Brain Tissue Density

After segmenting the region of interest, the image is divided into two hemispheres: left and right, and a GMM for classification is created considering the following classes based on Yousem [32] definitions: Class 1: Cerebrospinal Fluid (-99 at 5 HU); Class 2: Ischemic Stroke (6 to 22 HU); Class 3: White Mass (23 to 34 HU); Class 4: Gray Matter (32 to 41 HU); Class 5: Hemorrhagic Stroke (50 to 81 HU), and Class 6: Calcification (130 to 250 HU). Disregarding the Skull class, since it was removed in the region segmentation step, and adding Class 0 that represents the background, there are 7 classes in total.

With GMM trained, feature extraction is applied in each hemisphere. Three variations of the use of Gaussian Blend for feature extraction were proposed: MGABTD-pixel, MGABTD-percent and MGABTD-neighborhood.

1) Analysis of Brain Tissue Densities: ABTD is a feature extraction method proposed by Rebouças Filho et al. [23] in which the feature vector is represented by the percentage of pixels in each class, considering the HU intervals presented in the literature. As we have mentioned before, using the intervals proposed in the literature directly can be problematic, due to cases of overlap and different thresholds in the same region, which is the main limitation of this technique.

2) MGABTD-pixel: In this approach, instead of directly using the thresholds from the literature to determine the pixel class as in ABTD, the class predicted by the GMM is used. In this method, the classes extracted from each pixel are already considered as characteristics. Therefore, it is possible to observe that the number of features generated is a risk factor for time performance if the images are not resized to a smaller size, since the feature vector size will be equivalent to the number of image pixels.

3) MGABTD-percent: In this method, as in ABTD, the percentage of pixels in each class is also calculated and this represents the vector of image characteristics. The difference is found in the determination of the class of a pixel that in ABTD uses the thresholds of the literature and in MGABTD-percent uses the predictions of the GMM.

4) MGABTD-neighborhood: As in [30], MGABTD-neighborhood also analyzes the neighborhood of pixels to define image characteristics. After the prediction of the classes of each pixel by the GMM, a post-processing is done to redefine the class of each pixel considering its neighborhood. The class redefinition of a pixel is done by analyzing its 8-neighborhood, and then calculating the class that has the highest occurrence in the neighborhood. The 8-neighborhood of a pixel at position (x, y), are pixels of positions: $\{(x+1, y), (x-1, y), (x, y+1), (x, y-1), (x+1, y+1), (x-1, y-1), (x+1, y-1), (x-1, y+1)\}$. After redefining the pixel class, the process is similar to the MGABTD-percent, where the percentage of pixels in each class is calculated and this represents the characteristics of the image.

With the features extracted from each of the hemispheres, this information is joined in a vector forming the final features of the image.

IV. Results and discussion

This section contains the results obtained in the performance analysis of the 3 proposed extraction methods. To determine the efficiency and robustness of the extractors, they will be compared to the results obtained in the most recent related works. Tests were performed on an Intel(R) Xeon(R) CPU @2.00GHz and 12GB of RAM. The proposed methods were implemented using Python, version 3.6.9.

A. Classifiers

The efficiency of extractors were evaluated by means of experiments using classifiers to identify the best configurations, the classifiers used are presented below.

1) K-Nearest Neighbors: The K-Nearest Neighbors (KNN) algorithm for classification uses the distance between samples to predict the class, observing the classes of the k nearest neighbors. KNN, despite being simple, obtains satisfactory results in many [27] applications. The default configuration used was: k=5.

2) Multilayer Perceptron: Multilayer perceptron (MLP) have been widely used to solve classification and regression problems. MLP is a neural network with 1 or more hidden layers [4], with 1 hidden layer capable of approximating any continuous function and with 2 two layers able to approximate any function [6]. The default configuration used was: max_iter=200, solver=adam, neurons=100, activation=relu.

3) Support Vector Machine: The Support Vector Machine (SVM) is an algorithm that seeks to find a hyperplane that separates two classes. The algorithm is based on Statistical Learning Theory, which seeks to minimize structural risk and establishes principles for a classifier to have good generalization [29]. The default configuration used was: C=1.0, kernel=rbf, gamma=scale.

4) Optimum Path Forest: Optimum Path Forest (OPF) is a non-parametric classification algorithm based on Graph Theory [22]. Because it has a lower computational cost and it is not necessary to analyze the separability of classes, this algorithm presents an alternative to classical methods. The default configuration used was: distance=log_squared_euclidean.

5) Random Forest: Random Forest (RF) is an ensemble algorithm that creates several decision trees in randomly selected subspaces of the feature space. The RF is proposed to improve the generalization capability of decision-tree-based classifiers following the principles of stochastic modeling [14]. The default configuration used was: n_estimators=100.

6) Extremely Randomized Trees: Extremely Randomized Trees (ERT) or Extra Tree, is also a tree and ensemble-based algorithm. The difference for RF is that *Extra Tree* does the separation of samples without replacement. In RF, it is done by means of replacement, and the nodes are randomly divided [13]. The default configuration used was: n_estimators=100.

B. Best MGABTD Extractor

A grid search was performed for each extractor, the GMM parameters tested were: number of components per class and the maximum number of iterations in the EM process. Following the lessons learned about the quality of classification even after resizing the images, discussed in [26], percentages of reduction in image size were tested. The classifiers were used in their default settings. Each possible configuration of each extractor was evaluated using 3 different techniques: Hold Out (HO), Cross Validation (CV) with stratified sampling and Leave One Out (LOO). In the following sections, the configurations of each extractor that obtained the best results considering the 3 validation techniques will be presented. The best configuration obtained for each extractor is shown in Table I. Note that the MGABTD-neighborhood extractor has a much longer time than the other tested extractors.

TABLE I
BETTER CONFIGURATION OF PARAMETERS OF EACH EXTRACTOR.

Best Configuration	MGABTD-pixel	MGABTD-percent	MGABTD-neighborhood
No. of Components per Class	Classe 0: 2 Classe 1: 2 Classe 2: 4 Classe 3: 2 Classe 4: 2 Classe 5: 4 Classe 6: 4	Classe 0: 2 Classe 1: 2 Classe 2: 2 Classe 3: 2 Classe 4: 2 Classe 5: 2 Classe 6: 2	Classe 0: 2 Classe 1: 2 Classe 2: 2 Classe 3: 2 Classe 4: 2 Classe 5: 2 Classe 6: 2
Iterations EM-step	5	5	5
Size Reduction	30%	30%	40%
Extract Time	0.44s	0.37s	18.9s

C. Best Classification

After obtaining the best configuration of each extractor, a grid search of the classifiers was performed. The chosen parameters correspond to the most relevant ones described in the previous sections and are presented in Table II. Determining the best configuration of each classifier according to the extractor is necessary.

From the best configuration of parameters of each extractor classifier, the general performance of each proposed approach can be compared. The 3 validation techniques were also used: HO, CV, and LOO. For the HO technique, two forms of separation of the base were used: 80-20 and 70-30, and for the CV technique, 3 types of separation were used: 5, 10 and 15 folds. In Figures 1, 2 and 3, the average accuracies obtained by each extractor considering each technique of validation. This view is important when it comes to observing that, in general, there is little variation in performance among validation techniques.

Fig. 1. Average accuracy of extractor MGABTD-pixel

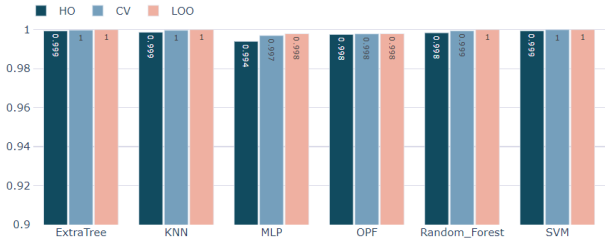
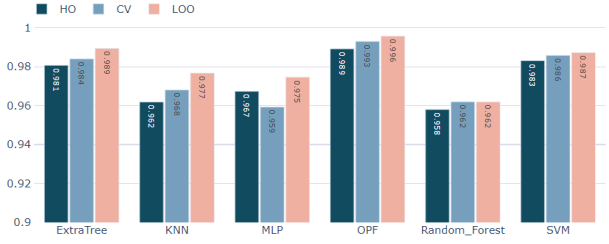


Fig. 2. Average accuracy of extractor MGABTD-percent

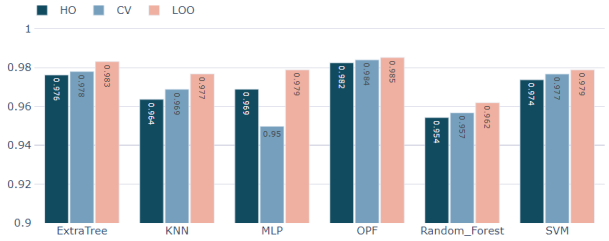


To make the comparison between extractors more robust, instead of choosing the results of only one of the validation

TABLE II
RESULTS OBTAINED WITH THE OPTIMAL SETTINGS OF EACH EXTRACTOR-CLASSIFIER.

Extractor	Classifier	Best Configuration	ACC	Precision	Recall	F1-Score	PPV	NPV	Jaccard	Train (s)	Test (ms)
MGABTD-pixel	**Extra Tree**	**n_estimators=100**	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**	**0.999**	**0.34**	**2**
	KNN	k=5	0.999	0.999	0.999	0.999	0.999	0.999	0.998	0.07	4
	OPF	distance=euclidean	0.997	0.997	0.997	0.997	0.997	0.998	0.996	14.85	15
	RF	n_estimators=300	0.999	0.999	0.999	0.999	0.999	0.999	0.998	0.80	3
	SVM	C=10;kernel=rbf;gamma=0.0001	0.999	0.999	0.999	0.999	0.999	0.999	0.999	1.02	4
	MLP	max_iter=10000;solver=sgd neurons=50;activation=tanh	0.996	0.996	0.996	0.996	0.995	0.997	0.993	31.69	0.007
MGABTD-percent	Extra Tree	n_estimators=100	0.984	0.985	0.984	0.984	0.983	0.987	0.973	0.09	0.1
	KNN	k=5	0.968	0.970	0.968	0.968	0.969	0.976	0.947	0.001	0.003
	OPF	**distance=euclidean**	**0.992**	**0.992**	**0.992**	**0.992**	**0.991**	**0.994**	**0.987**	**0.71**	**0.9**
	RF	n_estimators=100	0.960	0.960	0.960	0.960	0.961	0.972	0.936	0.14	0.3
	SVM	C=1000;kernel=rbf;gamma=scale	0.985	0.985	0.985	0.985	0.986	0.989	0.975	0.007	0.002
	MLP	max_iter=10000;solver=adam neurons=100;activation=relu	0.967	0.971	0.967	0.966	0.966	0.973	0.945	4.25	0.002
MGABTD-neighborhood	Extra Tree	n_estimators=100	0.979	0.980	0.979	0.979	0.979	0.984	0.964	0.09	0.1
	KNN	k = 5	0.969	0.971	0.969	0.969	0.969	0.976	0.949	0.001	0.003
	OPF	**distance = euclidean**	**0.983**	**0.984**	**0.983**	**0.983**	**0.985**	**0.988**	**0.973**	**0.55**	**0.9**
	RF	n_estimators=150	0.957	0.959	0.957	0.957	0.959	0.968	0.931	0.21	0.2
	SVM	C=1000;kernel=rbf;gamma=1.0	0.976	0.977	0.976	0.976	0.978	0.983	0.961	0.005	0.002
	MLP	max_iter=10000;solver=adam neurons=150;activation=relu	0.965	0.974	0.965	0.965	0.963	0.969	0.941	6.77	0.004

Fig. 3. Average accuracy of extractor MGABTD-neighborhood



techniques, an arithmetic mean of the metrics was calculated considering all 3 validation techniques: HO, CV and LOO. For example, the final accuracy will be calculated as the arithmetic mean of the average accuracy obtained in HO, CV and LOO, so all 3 techniques impact the final results, instead of choosing just one of them. In addition to accuracy, the following metrics were also calculated: precision, recall, f1-score, Positive Predictive Value (PPV), Negative Predictive Value (NPV) and Similarity Jaccard, all following the same logic of the average between the validation techniques. Table II presents the results of each metric considering the optimal configurations of extractors and classifiers. Even considering 3 validation techniques to compute the metrics, apparently some classifiers have very similar performance, for example the performance of the classifiers of the MGABTD-pixel extractor.

To verify if the classifiers of each extractor have equivalent performance and that the accuracy variations presented in Table II and Figures 1, 2, 3 could be the result of chance, the hypothesis test 5x2 cross-validation paired t-test was applied. The significance level used was 0.05, with the null hypothesis that the classifiers have equal performance. For the MGABTD-pixel extractor, the test proved that there are no significant differences between the performance of the classifiers tested, therefore they are equivalent. For the classifiers of the MGABT-percent and neighborhood extractors, the conclusions were similar, only the MLP and SVM classifiers showed significant differences, and the classifiers: KNN, Extra Tree, OPF, and RF, proved to be equivalent for these extractors.

Considering that some classifiers presented equivalent performance, time will be used as a selection criterion. Analyzing training and testing times shown in Table II, the KNN, RF, and Extra Tree classifiers have an advantage over the other classifiers, regardless of the extractor used. The MGABT-pixel, as expected, had the longest training time compared to the other extractors, as it has more features. The MGABTD-neighborhood extractor did not present any benefit when compared to the others, since it has a longer extraction time and lower average accuracy, and therefore its use is not indicated.

The Friedman hypothesis test was applied to compare the performance of the MGABTD extractors with the ABTD base method. The significance level used was 0.05, with the null hypothesis that there is no difference in the accuracies obtained between extractors for a given classifier. Using KNN, since it was the fastest in training and testing, the null hypothesis was rejected, so there is a statistically significant difference between at least two extractors. In the related works, only the solutions with deep learning presented by Dourado et al. [8], [9] obtained higher accuracy than the MGABTD-pixel. Five of the 12 architectures that achieved 100% accuracy, DenseNet (121, 169, and 201), InceptionResNetV3, and NasNetLarge, had longer extraction and testing times than MGABTD-pixel and percent.

V. CONCLUSION AND FUTURE WORK

In this research, 3 variations of feature extraction were presented using gaussian mixture and analysis of brain tissue density. The extractors: MGABTD-pixel, MGABTD-percent

and MGABTD-neighborhood represent an evolution of the ABTD method using GMM. As presented, the main contribution of this approach is the use of a probability estimator to determine the pixel class. The thresholds presented in the literature usually come from medical observations, which may vary from research to research. With the probability estimator, the definition of the pixel class tends to become more robust and reliable.

After the grid search, the best extractor configurations were analyzed in order to identify which would have the best performance. The MGABTD-neighborhood extractor did not present significant advantages when compared to other extractors. The MGABTD-pixel extractor takes an average of 0.44s to extract and with the Extra Tree, KNN, RF and SVM classifiers it achieves 99.9% accuracy. The MGABTD-percent extractor does the extraction in an average of 0.37s and with the OPF it obtains an accuracy of 99.2%. Thus, it is concluded that both extractors: MGABTD-pixel, MGABTD-percent are efficient methods for feature extraction using CT neuroimaging, considering the extraction performance and speed.

For future research, the objective is: the acquisition of more images for further validation of the proposed method, include parallelization techniques to make the extraction faster, so it is not necessary to reduce the image size and test with state-of-the-art classifiers.

REFERENCES

[1] Z. Akkus, P. Kostandy, K. A. Philbrick, and B. J. Erickson, "Robust brain extraction tool for ct head images," *Neurocomputing*, vol. 392, pp. 189–195, 2020.

[2] C. Archambeau, M. Valle, A. Assenza, and M. Verleysen, "Assessment of probability density estimation methods: Parzen window and finite gaussian mixtures," in *2006 IEEE International Symposium on Circuits and Systems*. IEEE, 2006, pp. 4–pp.

[3] T. Badriyah, N. Sakinah, I. Syarif, and D. R. Syarif, "Machine learning algorithm for stroke disease classification," in *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. IEEE, 2020, pp. 1–5.

[4] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[5] Y.-T. Chen, Y.-L. Chen, Y.-Y. Chen, Y.-T. Huang, H.-F. Wong, J.-L. Yan, and J.-J. Wang, "Deep learning–based brain computed tomography image classification with hyperparameter optimization through transfer learning for stroke," *Diagnostics*, vol. 12, no. 4, p. 807, 2022.

[6] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[8] C. M. Dourado, S. P. P. da Silva, R. V. M. da Nobrega, P. P. Reboucas Filho, K. Muhammad, and V. H. C. de Albuquerque, "An open ioht-based deep learning framework for online medical image recognition," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 541–548, 2020.

[9] C. M. Dourado Jr, S. P. P. da Silva, R. V. M. da Nobrega, A. C. d. S. Barros, P. P. Reboucas Filho, and V. H. C. de Albuquerque, "Deep learning iot system for online stroke detection in skull computed tomography images," *Computer Networks*, vol. 152, pp. 25–39, 2019.

[10] V. L. Feigin, M. Brainin, B. Norrving, S. Martins, R. L. Sacco, W. Hacke, M. Fisher, J. Pandian, and P. Lindsay, "World stroke organization (wso): Global stroke fact sheet 2022," *International Journal of Stroke*, vol. 17, no. 1, pp. 18–29, 2022.

[11] A. Gautam and B. Raman, "Brain strokes classification by extracting quantum information from ct scans," *Multimedia Tools and Applications*, pp. 1–17, 2021.

[12] ——, "Towards effective classification of brain hemorrhagic and ischemic stroke using cnn," *Biomedical Signal Processing and Control*, vol. 63, p. 102178, 2021.

[13] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[14] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.

[15] M. A. Inamdar, U. Raghavendra, A. Gudigar, Y. Chakole, A. Hegde, G. R. Menon, P. Barua, E. E. Palmer, K. H. Cheong, W. Y. Chan *et al.*, "A review on computer aided diagnosis of acute brain stroke," *Sensors*, vol. 21, no. 24, p. 8507, 2021.

[16] R. Kanchana and R. Menaka, "Ischemic stroke lesion detection, characterization and classification in ct images with optimal features selection," *Biomedical Engineering Letters*, vol. 10, no. 3, pp. 333–344, 2020.

[17] Q. Ke, J. Zhang, W. Wei, R. Damaševičius, and M. Woźniak, "Adaptive independent subspace analysis of brain magnetic resonance imaging data," *Ieee Access*, vol. 7, pp. 12 252–12 261, 2019.

[18] C.-M. Lo, P.-H. Hung, and D.-T. Lin, "Rapid assessment of acute ischemic stroke by computed tomography using deep convolutional neural networks," *Journal of Digital Imaging*, vol. 34, no. 3, pp. 637–646, 2021.

[19] G. McLachlan and K. Basford, "Mixture models marcel dekker," *New York*, 1988.

[20] S. Mendis, P. Puska, B. e. Norrving, W. H. Organization *et al.*, *Global atlas on cardiovascular disease prevention and control*. World Health Organization, 2011.

[21] H. Oliveira, L. Penteado, J. L. Maciel, S. F. Ferraciolli, M. S. Takahashi, I. Bloch, and R. C. Junior, "Automatic segmentation of posterior fossa structures in pediatric brain mris," in *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2021, pp. 121–128.

[22] J. P. Papa and A. X. Falcao, "Optimum-path forest: a novel and powerful framework for supervised graph-based pattern recognition techniques," *Institute of Computing University of Campinas*, vol. 4148, 2010.

[23] P. P. Rebouças Filho, R. M. Sarmento, G. B. Holanda, and D. de Alencar Lima, "New approach to detect and classify stroke in skull ct images via analysis of brain tissue densities," *Computer methods and programs in biomedicine*, vol. 148, pp. 27–43, 2017.

[24] D. A. Reynolds, "Gaussian mixture models." *Encyclopedia of biometrics*, vol. 741, no. 659-663, 2009.

[25] M. Roder, G. H. Rosa, J. P. Papa, D. Carlos *et al.*, "Enhancing shallow neural networks through fourier-based information fusion for stroke classification," in *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. IEEE, 2021, pp. 378–385.

[26] R. M. Sarmento, F. F. Vasconcelos, P. P. Rebouças Filho, and V. H. C. de Albuquerque, "An iot platform for the analysis of brain ct images based on parzen analysis," *Future Generation Computer Systems*, vol. 105, pp. 135–147, 2020.

[27] G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-neighbor methods in learning and vision: theory and practice (neural information processing)*. The MIT press, 2006.

[28] M. S. Sirsat, E. Fermé, and J. Câmara, "Machine learning for brain stroke: a review," *Journal of Stroke and Cerebrovascular Diseases*, vol. 29, no. 10, p. 105162, 2020.

[29] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.

[30] F. F. Vasconcelos, R. M. Sarmento, P. P. Reboucas Filho, and V. H. C. de Albuquerque, "Artificial intelligence techniques empowered edge-cloud architecture for brain ct image analysis," *Engineering Applications of Artificial Intelligence*, vol. 91, p. 103585, 2020.

[31] J. Yanase and E. Triantaphyllou, "A systematic survey of computer-aided diagnosis in medicine: Past and present developments," *Expert Systems with Applications*, vol. 138, p. 112821, 2019.

[32] D. M. Yousem, *Neuroradiologia*. Elsevier Brasil, 2011.

[33] S. Zahoor, I. U. Lali, M. A. Khan, K. Javed, and W. Mehmood, "Breast cancer detection and classification using traditional computer vision techniques: a comprehensive review," *Current Medical Imaging*, vol. 16, no. 10, pp. 1187–1200, 2020.