IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 17, 2020, accepted October 14, 2020, date of publication October 19, 2020, date of current version October 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3032376

Convolutional Neural Network to Detect and Measure Fetal Skull Circumference in Ultrasound Imaging

EVERTON LEONARDO SKEIKA[1], **MATHIAS RODRIGUES DA LUZ**[2],
BRUNO JOSÉ TORRES FERNANDES[3], **(Member, IEEE),**
HUGO VALADARES SIQUEIRA[2],
AND MAUREN LOUISE SGUARIO COELHO DE ANDRADE[1]

[1]Department of Informatics, Graduate Program in Computer Science (PPGCC), Federal University of Technology—Paraná (UTFPR), Ponta Grossa 84017-220, Brazil
[2]Department of Electrical Engineering, Federal University of Technology—Paraná (UTFPR), Ponta Grossa 84017-220, Brazil
[3]Polytechnic School of Pernambuco (POLI), University of Pernambuco (UPE), Recife 50720-001, Brazil

Corresponding author: Mauren Louise Sguario Coelho de Andrade (mlsguario@utfpr.edu.br)

This work was supported in part by the Federal University of Technology—Parana, in part by the National Council for Scientific and Technological Development (CNPq), Brazil, under Grant 405580/2018-5, and in part by the Araucaria Foundation under Grant 51497.

ABSTRACT In obstetrics, ultrasound is used for assessment of fetal development during pregnancy. The images generated by ultrasound are used to obtain measurements of fetal head length, body size, and the analysis of fetal movements, to identify and prevent the onset of congenital disease. This work presents the development of a new method for the segmentation of two-dimensional ultrasound images of fetal skulls based on a V-Net architecture called Fully Convolutional Neural Network - Combination (VNet-c). We created a new combination of strategies using a 3D V-Net as base, such as pre-processing, use of Batch Normalization and Dropout, and evaluation of distinct activation layers, activation function, data augmentation, loss function, and network depth. The computational results reveal the feasibility of the proposal in the correct segmentation of fetal skulls and head circumference measurements, reaching up to 97.91% of correctness, overcoming states-of-the-art methods.

INDEX TERMS Convolutional neural networks, deep learning, fetal skull, measurement.

I. INTRODUCTION

Ultrasound is a pathology diagnosis method that uses ultrasonic waves for real-time imaging. Due to its noninvasive and non-radioactive nature and reduced cost compared to other imaging forms such as CT or MRI, it is the choice for many clinical applications [1], [2]. This technique assists in the medical diagnosis in several areas such as obstetrics, gynecology, ophthalmology, neurology, and cardiology. Besides that, it is used as a standard tool in therapeutic procedures [3].

In obstetrics, ultrasound is widely approached for the assessment of fetal development during pregnancy. The images generated by the ultrasound equipment are used by specialists to obtain measurements of fetal head length, body size, and the analysis of their movements, to identify and prevent the onset of congenital disease [4].

The main measures to evaluate fetal development are head circumference (HC), biparietal diameter, abdominal circumference, femur length, and humerus length [5]. The calculation of these values is performed by specific mathematical functions, helping the specialist estimate the fetus's gestational age and weight [6].

In practice, the delimitation of the area to be measured is done manually by an obstetrician. This process requires specialized knowledge and is a dull and time-consuming process [7]. Also, the contour extraction of the forming bones and organs is influenced by the experience of the evaluator [8].

In this regard, to facilitate such a process and assist in analyzing of the results, automatic image segmentation and measurement techniques are natural candidates [8]. However, ultrasound remains a challenging task because the generated image may present various intensity distributions due to different acquisition conditions. Besides, a series of noise, such as acoustic shadows, speckle noise, and low contrast, can make border recognition difficult [2], [8].

Among the various image segmentation techniques found in the literature, Convolutional Neural Networks (CNN) has proven effective [9]. It is an architecture based on the structure

The associate editor coordinating the review of this manuscript and approving it for publication was Sunil Karamchandani.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

of the mammalian visual cortex, which has become popular and successful in many tasks, as visual recognition and object detection *Faster-RCNN* [10], image classification *GoogLeNet* [11], and image segmentation [12].

In this sense, this work proposes developing a new method for the automatic segmentation of fetal skulls in two-dimensional (2D) ultrasound images. The proposed method is called Fully Convolutional Neural Network Combination (VNet-c), and it is based on the original V-NET to recognize and measure the circumference of the fetal skull. We used quantitative methods based on negative and positive rates to evaluate the performance of the developed method. A comparative analysis with some related works reveals the capacity to outperforming other segmentation techniques.

The main contributions of this paper are the improvement of original VNet version adding a bunch of adaptations such as pre-processing, use of Batch Normalization, change in layer and function of activation, dropout application, use of data augmentation, change in loss function and network depth, besides that we use a important step of post-processing. With that, we achieved better performance that is proved in Section V.

The rest of this article is organized as follows: in Section II, we show a review of the related works that use a Convolutional Neural Network to solve segmentation problems. Section III describes the background developed in this study. Next, in Section V we detail the proposed method and provide an analysis regarding the experimental results. Finally, the conclusion and future works perspective are shown in Section VII.

II. RELATED WORKS

Due to the relevance of the measures to evaluate fetal development problems, many recent investigations have addressed improvements in the segmentation of fetal organs and bones in two-dimensional ultrasound images. Li *et al.* [13] proposed using of a Fully Convolutional Neural Network (FCN) with an Encoder-Decoder structure for automatic segmentation of amniotic fluid and fetal tissues. They addressed 2D ultrasound images at different points of view and arbitrary positions.

The work from Sundaresan *et al.* [14] used a FCN to automatically localize the fetal heart in ultrasound (echocardiography) video frames and classify them as belonging to one of three standard visualization planes, namely: left ventricular outflow, three-vessel view, and four-chamber view of the heart. Their primary objective was assisting the identification of cardiac anomalies and congenital heart disease.

In 2017, Wu *et al.* [15] proposed a customization in the FCN, naming it as FCN cascade (casFCN) for fully automatic segmentation of the fetal skull and abdomen. According to the authors, the network was successful used to exploit feature extractions from multiple visual scales and distinguish the anatomy with a dense prediction map.

In 2018, Sinclair and collaborators [16] used a FCN and *Ramanujan Approximation* II [17] for segmentation and automatic measure of head circumference and biparietal diameter in 2D ultrasound images. Heuvel *et al.* used a Random Forest Classifier (RFC) to locate fetal skull throughout Haar-like features computed from ultrasound images [18].

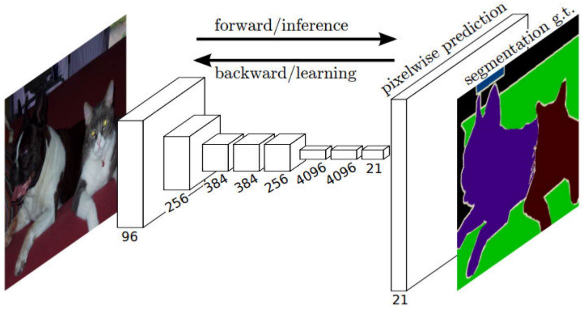


FIGURE 1. **Schematic representation of FCN network architecture proposed by Long; Shelhamer; Darrell (2015) [12].**

The work from Sobhaninia *et al.* [19] proposed a deep multi-task network based on the structure of the Link-Net network, introduced by Chaurasia and Culurciello [20], for segmentation and measurement of fetal skulls in two-dimensional ultrasound images.

Recently, in 2020, Sobhaninia and colleagues [21] presented a CNN multiscale based approach to localize the fetal head region in US imaging. Qu *et al.* also used a CNN-based method for the identification of fetal brain ultrasound. They proposed a differential convolutional neural network (differential-CNN) to identify six fetal brains automatically [22].

Wang *et al.* [23] published a review article of deep-learning-based methods for ultrasound image segmentation. They analyzed and summarized several algorithms, the performance of the methods, and their evaluation results.

III. FULLY CONVOLUTIONAL NEURAL NETWORK

Created by Long, Shelhamer, and Darrell in 2015 [12], the Fully Convolutional Network (FCN) presents similarities with the traditional Convolutional Network (CNN). However, in the FCN architecture, the fully connected layer, typically used for classification, is replaced by another convolution layer with a large ‘‘receptive field’’ used to classify each pixel of an image. The idea is to capture the overall context of the scene, i.e. to define which objects are in the image, and where they are located.

Figure 1 presents the architecture of a generic FCN used for the semantic segmentation task, that is, classifying each pixel of the input image according to the class it belongs to: cat, dog, sofa, window, or background.

According to the architecture presented in Figure 1, several convolution layers produce feature maps of different depths. At the end of the network, there is the pixel-wise prediction, which is also a type of convolution layer that makes a pixel-by-pixel prediction, that is, it assigns each pixel to a respective class. In this example, the pixel-wise prediction size is 21 due to the existence of 21 distinct classes in the dataset.

Usually, the architecture of an FCN network can be divided into two main parts:

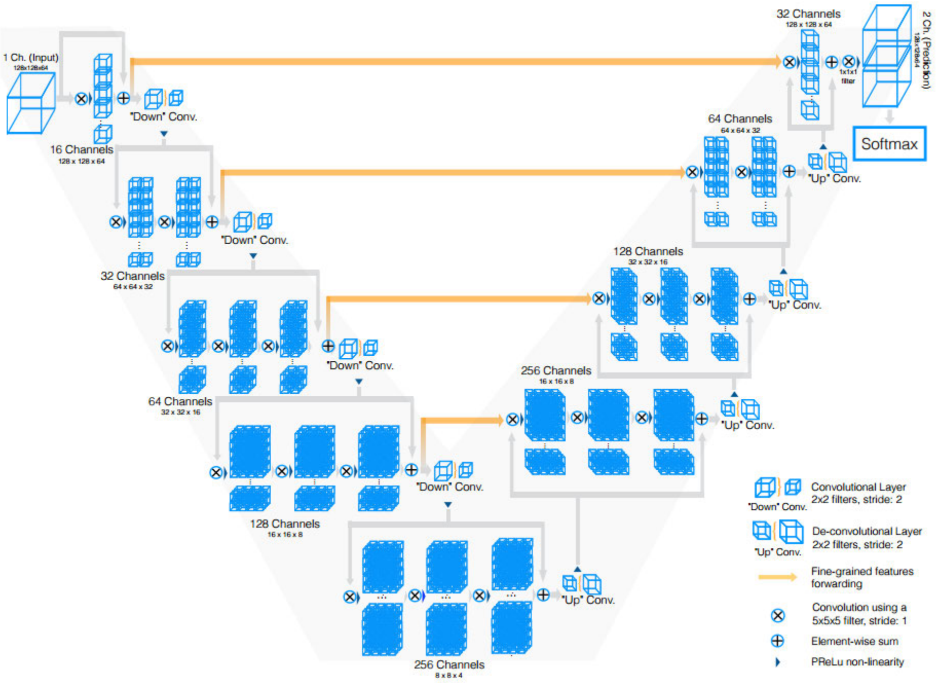


FIGURE 2. Schematic representation of the V-Net architecture proposed by Milletari, Nava, and Ahmadi [24].

- **downsampling path:** reduces image size using pooling steps;
- **upsampling path:** is the opposite of pooling layers. In its simplest form, it returns the image to its original resolution.

The FCN is widely used for image segmentation because this architecture can segment the network input and simultaneously classify it, without the use of other technique for sectioning the image.

A. V-Net FULLY CONVOLUTIONAL NEURAL NETWORK

Created by Milletari, Navab, and Ahmadi in 2016, the V-Net Fully Convolutional Neural Network is designed for segmentation of three-dimensional (3D) magnetic resonance imaging of the prostate [24]. This architecture performs convolutions to extract data characteristics, which can be described in two parts, the left part (compression) and the right part (decompression), as summarized in Figure 2.

The left part consists of a compression path. This side is divided into different stages operating at different resolutions, and these stages present one to three convolutional layers. As can be noted in Figure 2, the convolutions at each stage use volumetric kernels with $5 \times 5 \times 5$ voxels. As data proceeds at different stages along the compression path, its resolution is reduced. This is accomplished by convolutions with $2 \times 2 \times 2$ wide kernel voxels and using a stride of 2.

Figure 3 presents an example of the operation of an FCN. In Figure 3(a), for each operation, the size of feature maps is halved. This strategy has a similar purpose to pooling layers (not used in this network). The authors also applied the PReLU activation function, proposed by He *et al.* [25], across the network. The downsampling technique is also addressed to reduce the signal size presented as input and increases the receptive field of the characteristics, being computed in the network layers. Each of the stages on the left side of the network calculates several characteristics that are twice as large as the previous layer [24].

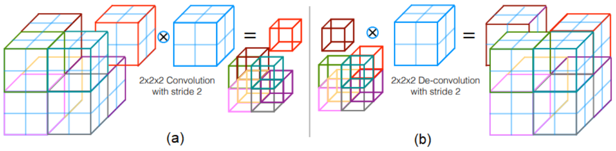


FIGURE 3. (a) Example of convolution using a $2 \times 2 \times 2$ kernel and stride of 2, (b) example of de-convolution using a $2 \times 2 \times 2$ kernel and stride of 2.

The right side of the network extracts the features and expands the low-resolution feature maps to gather and to assemble the information needed to perform the segmentation. After each step, a convolution operation is used to increase the output size, as shown in Figure 3(b), followed by one to three convolutional layers involving half the number of $5 \times 5 \times 5$ kernels resulting from the previous layer. At the end, the two feature maps computed by the last convolutional layer are $1 \times 1 \times 1$ kernel size and produce outputs the same size as the input volume. These are processed through a *Softmax* layer that generates the probability of each voxel belonging to the background or foreground [24].

IV. PROPOSED ARCHITECTURE: VNet-C

We propose a new architecture for automatic segmentation of fetal skulls in two-dimensional (2D) ultrasound images, and to recognize and measure the circumference of fetal skulls, the Fully Convolutional Neural Network - Combination (VNet-c). We created a solution using as base the V-Net FCN, including eight steps and techniques that can work

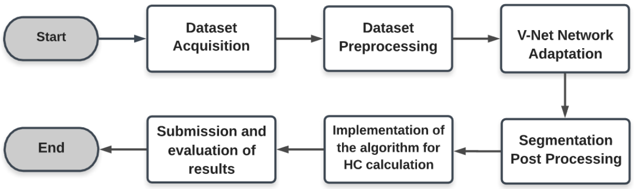

Start
Dataset Acquisition
Dataset Preprocessing
V-Net Network Adaptation
Segmentation Post Processing
Implementation of the algorithm for HC calculation
Submission and evaluation of results
End

FIGURE 4. Steps addressed to apply the proposed VNet-c architecture.

together to improve the accuracy in measuring the head circumference.

We present in the flowchart of Figure 4 the steps involved in the application of VNet-c: acquisition of the dataset, pre-processing dataset, the definition of the better combination of the V-Net network based on some adjustments, post-processing of segmentation, implementation of the algorithm to calculate the HC and evaluation of the results.

To elaborate the VNet-C, many adaptations were performed in the original V-Net 3D introduced by Milletari *et al.* [24]. The first is to change the network to receive 2D inputs, modifying the cost function, output layer, and activation functions.

A new pre-processing phase was introduced in this work. This step initially reduces the resolution of the original images to decrease the computational cost. After, we created an algorithm to fill the images, making them solid ellipses.

The network has been deepened, including more stages, to improve its learning capacity and implicit analysis of the characteristics. However, a deeper network necessarily increases the number of trainable parameters and may lead to an overfitted configuration [26]. To prevent this undesirable behavior, we added the Dropout mechanism [27], and we propose the use of Data Augmentation to create new input patterns, using techniques such as rotation and translation [28]. Batch-normalization was also addressed because it has a regularization effect and makes the most model more robust. Also, its application accelerates the learning process [11].

An unprecedented post-processing stage was also implemented to correct defective output images. The new algorithm generates ellipses, fitting them in the contours.

It is important to mention that we evaluated the use of distinct activation functions in the hidden layers and in the output layer. The final choice was the use of ELU function (intermediate layers) and hard sigmoid (in the last layer).

Many empirical adjustments we performed until we defined the final configuration of the VNet-c. The choices that guide our proposal are detailed in the next sections.

V. VNet-C ADAPTATION

In this Section we present in detail the steps involved in the development and application of the VNet-c to measure fetal skull using ultrasound images. Initially, we describe the database addressed, and the performance metrics used to evaluate our choices. After, we present the new pre and post-processes schemes, as well as the formative stages of the model.

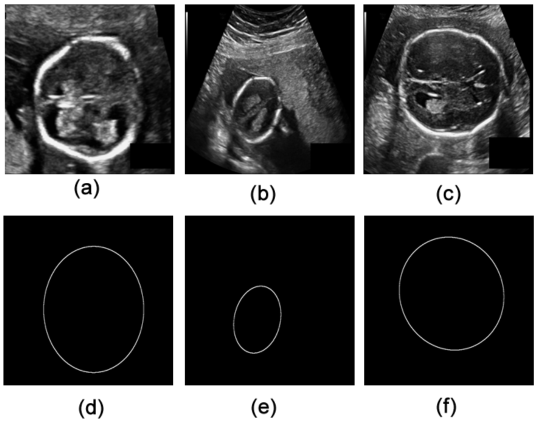


FIGURE 5. Representation of ultrasound images and ground truth contained in the dataset.

A. ACQUISITION OF DATASET

In this work, we addressed the dataset of the HC18 challenge, which was provided by Heuvel and collaborators [18]. Each image set is composed of 1,334 grayscale 2D ultrasound images, with a resolution of 800 × 540 pixels. To adjust the FCN, we use 999 samples, and other 297 are utilized for testing. The training set presents a ground truth image, a manual annotation of the head circumference performed by a trained sonographer. Figure 5 illustrates some of the images present in the training dataset, in which (a)-(c) are three 2D ultrasound images, and (d)-(f) their respective ground truth.

The HC18 challenge did not provide the ground truths of the test for outcome evaluation. Therefore, we randomly selected 297 images from the 999 of the training set (approximately 30%), with their respective pre-filled ground truths to make comparison possible. The results obtained by the network allowed the design of improvements in the architecture, leading to a segmentation closer to that specified by the specialist.

B. PERFORMANCE METRICS

To quantitatively evaluate the performance of fetal skull segmentation obtained by the proposed network, four metrics are addressed:

- **precision:** this metric measures the total of positive ground truth pixels (tp) that are also considered positive by segmentation, divided by the total number of pixels identified as belonging to the positive set (the sum of true positives (tp) and false positives (fn), which are the incorrectly labeled pixels set). Precision is given by Equation 1:

$$precision : \frac{|tp|}{|tp| + |fn|} \tag{1}$$

- **accuracy:** The accuracy is relative to the correct answers in the classification. It is the sum of true positives and true negatives (tn) divided by the whole outputs, or the sum of the true positives (tp), true negatives (tn), false positives (fp), and false positives (fn). This metric shows how the classifier has fared out [29]. accuracy i

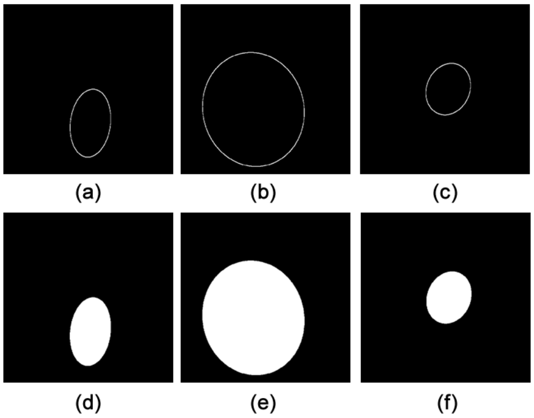


FIGURE 6. Result of filling in the ground truth.

given by Equation 2:

$$accuracy : \frac{|tp| + |tn|}{|tp| + |tn| + |fp| + |fn|} \tag{2}$$

- **dice:** dice Similarity Coeficient (DSC) [30] is defined as the intersection between two regions G and A, where $|G|$ is the ground truth, and $|A|$ is the image segmented by the algorithm. The symbol $\cap$ means the intersection of these two binary masks (sets), divided by the average volume of these two regions [31]. It can be expressed in therms of (tp), (tn), (fp), (fn), as in Equation 3:

$$dice : \frac{2|G \cap A|}{|G| + |A|} = \frac{2(tp)}{(tp + fp) + (tp + fn)} \tag{3}$$

- **Jaccard:** Jaccard Similarity Coefficient (JSC) is defined as the intersection between two regions G and A, being G the ground truth and A the image segmented by the algorithm. Again, the symbol $\cap$ represents the intersection of these two binary masks, being divided by their union, given by the symbol $\cup$ [31]. The Jaccard similarity coefficient is given by Equation 4:

$$Jaccard : \frac{|G \cap A|}{|G \cup A|} = \frac{tp}{tp + fp + fn} \tag{4}$$

C. PRE-PROCESSING

In this step two changes were made to the original dataset: first, the image size was reduced to a resolution of 512 × 512 pixels to decrease the computational cost of the training. The second was performed on ground truth images, and an algorithm was created to fill the 999 images, making them look like solid ellipses (Figure 6).

These modifications were necessary because some resulting ellipses from the segmentation presented incomplete edges. It makes impractical to accurately calculate the ellipse, as there is no way to delimit the missing area.

D. VNet-C DEFINITIONS

The network used for segmentation is an adaptation of the original V-Net 3D designed by Milletari *et al.* [24]. The source code is available for the segmentation of 2D images[1].

TABLE 1. Performance and number of epoch and steps during V-Net training.

V-Net	precision	accuracy	dice	jaccard	Ex. time
100st 500ep	0.97516	0.98748	0.97620	0.95382	~ 17 minutes
500st 100ep	0.97418	0.98697	0.97462	0.95136	~ 3,01 hours
500st 300ep	0.97697	0.98754	0.97552	0.95334	~8,33 hours
1000st 100ep	0.97468	0.98756	0.97611	0.95406	~ 5,50 hours
1000st 300ep	0.97699	**0.98778**	**0.97651**	**0.95447**	~ 16,25 hours
2000st 300ep	**0.97723**	0.98764	0.97648	0.95444	~ 36,00 hours

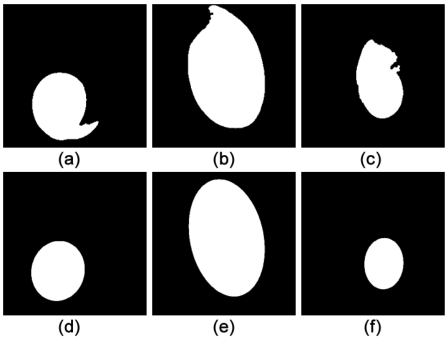


FIGURE 7. Initial results obtained by the network.

The network was built using the Keras[2] library, and Tensorflow was used as a backend with cuDNN for GPU network processing. The machine configuration used for the experiments was an Intel(R) Core(TM) I7-3370K processor, 8GB RAM, and a 6GB NVIDIA GeForce GTX TITAN graphics card.

Initially, basic changes were made to the execution of the FCN. As mentioned, we use images in 512 × 512 resolution. The network was trained using distinct numbers of epochs (*ep*) and steps (*st*) to find adequate values of these free parameters. Table 1 presents the results obtained considering precision, accuracy, dice, and Jaccard metrics considering the segmentation obtained by the FCN in the different training times, and the total execution time, according to the machine configuration used.

The evaluation of the segmentation indicates that the best result was obtained with 1000 steps and 300 epochs. We adopted these values as a standard for network training. Although the FCN achieved good results, Figure 7 illustrates three failed segmentations obtained by the net in 300 training epochs.

It can be seen in Figure 7 the results of the segmentation in (a)-(c) are inefficient compared to their ground truth (d)-(f). This inefficiency is related to overfitting, which is a term used to indicate that a model was overtrained. The consequence is that the model fits well with the previously observed dataset but is not effective in predicting outcomes outside of the training set. To improve the model's performance and reduce overfitting, some changes were made, as described below.

1) USE OF BATCH NORMALIZATION

The technique known as Batch Normalization (BN) was developed by Ioffe and Szegedy [32] to deal with

[1] https://github.com/FENGShuanglang/2D-Vnet-Keras

[2] https://keras.io/

TABLE 2. Performance and number of epoch and steps during V-Net training.

V-Net	precision	accuracy	dice	jaccard	Ex. time
without BN	0.97473	0.98697	0.97647	0.95438	~14,33
with BN	**0.97699**	**0.98778**	**0.97651**	**0.95447**	**~16,25**

TABLE 3. Comparative performance of the batch normalization positioning.

V-Net	precision	accuracy	dice	jaccard
with BN	0.97699	0.98778	0.97651	0.95447
edited BN	**0.97722**	**0.98807**	**0.97766**	**0.95662**

TABLE 4. Performances regarding the type of activation function used in the last layer.

V-Net	precision	accuracy	dice	jaccard	Ex. time
softmax	-	-	-	-	~18,25
sigmoid	**0.97722**	0.98807	0.97766	0.95662	~17,33
hard sigmoid	0.97681	**0.98809**	**0.97774**	**0.95665**	~15,91

initialization problems on CNN. A factor that hinders the training is the values of the previous layers' activation functions, which are always changing. In this sense, BN normalizes the output of an activation layer to follow a *Gaussian distribution*, which leads to an acceleration in the network learning process.

Although this technique was not described in the original 3D V-Net [24], it was applied to the 2D V-Net network. In this case, we analyze the performance during the training with and without the BN technique. The values of the metrics are in Table 2.

According to the superior results of precision, accuracy, dice, and Jaccard shown in Table 2, the use of BN is an advantage.

Another point evaluated was the positioning of the BN in the arranged code, since it is often applied before the PReLU activation function. However, the works [33] discusses changes in the position of using BN. Therefore, we tested if use the BN after PReLU could improve network performance. Table 3 shows the values obtained by the metrics in this regard.

Table 3 indicates that the use of BN after the PReLU function improved the performance of the segmentation. This change was aggregated in the network architecture.

2) CHANGE IN ACTIVATION LAYER

The last layer of an FCN has the important task of generating the probability that each pixel of the output image belonging to the background or foreground. The V-Net [24] in its original version, uses the *softmax* function. However, in the V-Net 2D network used as a base, the use of *sigmoid* function was employed. A comparative analysis is performed using other functions in the last layer, the *softmax*, and *hard_sigmoid*. It is noteworthy that the tests were applied to the network already with the changes imposed in the previous topics.

Table 4 expresses the results obtained between the segments obtained according to the activation function used in the last layer.

As shown in Table 4, it was not possible to make comparisons between the segmentations obtained by the network with the *softmax* function due to the low accuracy rate during their training. However, with the use of the *hard_sigmoid* function, there was a slight increase in segmentation accuracy and also a decrease in the training time. Therefore, the sigmoid function was replaced by the *hard_sigmoid* function.

TABLE 5. Performances achieved for distinct activation functions.

V-Net	precision	accuracy	dice	jaccard
PReLU (default)	0.97681	0.98809	0.97774	0.95665
ReLU	**0.97843**	0.98794	0.97753	0.95633
ELU	0.97747	**0.98812**	**0.97786**	**0.95693**
Leaky ReLU	0.97088	0.98730	0.97445	0.95152
Thresholded ReLu	0.77519	0.88165	0.78566	0.67029

TABLE 6. Performances achieved for distinct activation functions.

V-Net	precision	accuracy	dice	jaccard
without	0.97902	0.98793	0.97730	0.95593
0.2	0.97950	**0.98802**	**0.97747**	**0.95621**
0.5	0.98643	0.98597	0.97263	0.94752
0.7	**0.98652**	0.98563	0.97245	0.94698

3) CHANGE IN ACTIVATION FUNCTION

The activation function employed in the original network is PReLU, applied to all convolution steps. To further improve segmentation performance, tests were performed with four other activation functions in the Keras framework: ReLU, ELU, Leaky ReLU, and Thresholded ReLU (Table 5).

According to Table 5, the ELU function led to a better segmentation, given the highest average values in the accuracy, dice, and Jaccard metrics. Therefore, we adopted this function in the whole network.

4) DROPOUT APPLICATION

The dropout technique was proposed by Hinton *et al.* [27] to reduce overfitting during the training. This technique works by deactivating a set of neurons from the fully connected layer at each iteration of the adjustment phase.

The network used as a base applied this technique to the contraction path. Initially, the dropout was removed throughout to analyze if it is useful. We found that overfitting increased as the performance of segmentation worsened, as shown in Table 6. Empirical tests were conduced with the inclusion of this technique in the network expansion path, being applied after all the convolution steps. The network was trained with dropout values of 0.2, 0.5, and 0.7, ie 20%, 50%, and 70% of activations maintained.

As shown in Table 6, the increase in dropout usage did not outperform network segmentation, since the highest average accuracy value was 0.98812. In this sense, the dropout was kept only in the contraction layers.

5) USE OF DATA AUGMENTATION

The data augmentation technique is used to increase the training dataset by generating new artificial images from geometric transformations such as rotation, translation, noise injections, alteration in the color, brightness, contrast, cropping, among others. It creates new representations from the original images in the dataset [34]. Although the base network uses data augmentation, we perform some changes in order to minimize its overfitting.

TABLE 7. Operations and values used for synthetic image generation by data augmentation technique.

operation	value
rotation range	0.15
width shift range	0.05
height shift range	0.05
shear range	0.10
zoom range	0.10
horizontal flip	True
vertical flip	True

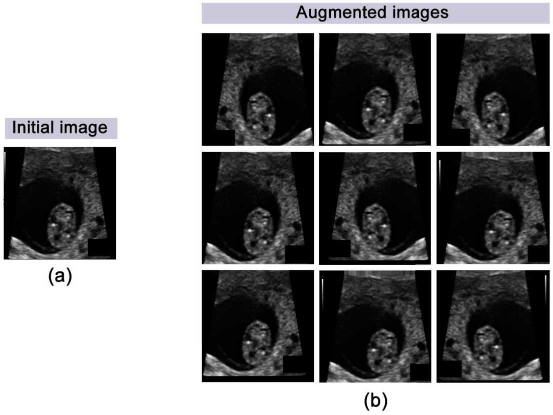


FIGURE 8. Example of applying data augmentation to the dataset used. (a) Original image (b) Augmented images.

TABLE 8. Performance metrics versus batch size edits.

V-Net	precision	accuracy	dice	jaccard	Ex. time
without	0.97717	0.97825	0.93339	0.93339	~12,66
2(default)	0.97577	0.98816	0.97794	0.95707	~16,00
4	**0.98009**	**0.98859**	**0.97842**	**0.95805**	~23,34
6	0.97768	0.98832	0.97793	0.95713	~31,75

Table 7 shows the values used for the generation of new synthetic images for network training. It is noteworthy that several empirical tests were performed, where distinct values were used, and other data augmentation operations were applied. However, the network with the best performance was achieved with the values described in this table.

As observed in Table 7 subtle values were used because the dataset used does not contain images so different from each other. That is, the new synthetic data generated from the transformations of the training samples had a remarkable resemblance to the original samples.

Figure 8 illustrates the results of some synthetic images generated by data augmentation operations from a reference image.

Also, other tests were performed to change the batch value from 2 to 4 and 6. However, this process further increased the number of artificial images generated for network training. As a consequence, there was a considerable increase in computational cost. Table 8 demonstrates network performance with changes in batch value and total time spent during the training.

From Table 8 it can be seen that changing the batch from 2 to 4, the processing time was increased about 45%, and from 2 to 6 about 95%. However, the modification from 2 to 4 led to an increase in segmentation performance. In this sense, we adopted this change in network.

TABLE 9. Performance metrics by batch size edits.

V-Net	precision	accuracy	dice	jaccard
dice (default)	0.98009	0.98859	0.97842	0.95805
binary cossentropy	0.97605	0.98744	0.97637	0.95414
Focal	0.98282	0.98861	0.97868	0.95849
Tversky	0.97608	**0.98866**	**0.97876**	**0.95866**
Lovasz-Softmax	**0.98386**	0.98821	0.97717	0.95607

TABLE 10. Performance metrics by number of network stage.

V-Net	precision	accuracy	dice	jaccard	Ex. time
2	0.79584	0.90482	0.81605	0.70813	~7,16
4	0.96605	0.98672	0.97158	0.94838	~16,75
5 (default)	0.97608	0.98866	0.97876	0.95866	~24,17
6	0.97379	0.98828	0.97777	0.95707	~29,70
7	**0.97666**	**0.98877**	**0.97894**	**0.95899**	**~ 53,35**

6) CHANGE IN LOSS FUNCTION

The loss function measures the difference between the expected output and the actual output obtained by anticipating network training. The higher the loss value, the less accurate the model is. Milletari *et al.* [24] proposed a new loss function based on the dice similarity metric for the V-Net network.

In this work, tests were also performed with other loss functions, such as the *binary cossentropy*, function used by the U-Net [28], the *Tversky* loss function proposed by Salehi *et al.* [35], the *Focal* loss function proposed by Lin *et al.* [36], and the *Lovasz-Softmax* loss designed by Berman *et al.* [37]. The results found using such functions are summarized in Table 9.

Table 9shows that the use of the Tversky loss function led to a better performance in fetal skull segmentation. We replaced the dice loss function by the Tversky loss function.

7) CHANGE IN NETWORK DEPTH

The last modification employed in the FCN addressed in this work was about using of a more or less dense "deep" network. By default, the network contains 5 stages. Tests were performed with different number of stages to analyze if the depth of the network influences the segmentation performance. The results are in Table 10.

From Table 10, it can be seen that the change in the depth of the network led to an increase in the segmentation performance, in which the 7-stage net reached the highest values in the metrics values. In this sense, this change was employed in network architecture.

However, the change in network depth resulted in a significant increase in computational cost due to the necessity of performing more convolutions. It is noteworthy that use more than 7 stages, would require a more powerful video card, with a higher amount of memory, which is not available.

E. POST-PROCESSING

To further improve the segmentation results, an algorithm was created in Python OpenCV[3] for post-processing, in order to

[3] https://opencv-python-tutroals.readthedocs.io/en/latest/

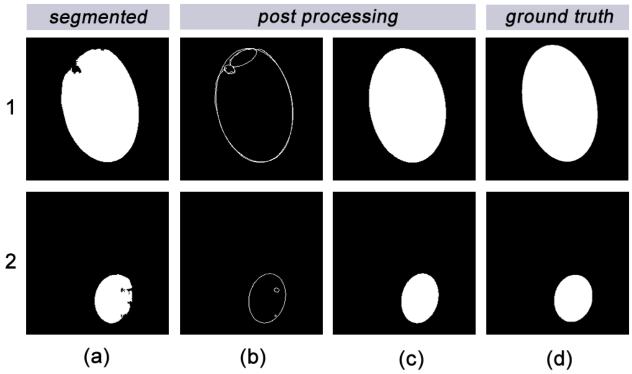


FIGURE 9. Example of the application of the post-processing algorithm.

TABLE 11. Comparison between the segmentations of Figure 9 (wopp) and with post-processing (wpp).

V-Net	precision	accuracy	dice	jaccard
1(a) wopp	**0.97141**	0.97366	0.96027	0.92358
1(c) wpp	0.96452	**0.97842**	**0.96794**	**0.93786**
2(a) wopp	**0.96616**	0.99448	0.95996	0.92301
2(c) wpp	0.96269	**0.99605**	**0.97181**	**0.94516**

TABLE 12. Comparison between the segmentations of Figure 9 (wopp) and with post-processing (wpp).

V-Net	precision	accuracy	dice	jaccard
wopp	0.97666	0.98877	0.97894	0.95899
wpp	**0.97677**	**0.98881**	**0.97918**	**0.95942**

correct defective segments. The goal is to generate a new ellipse fitting it to its contours. Figure 9 (a) illustrates a failed segmentation result obtained by the VNet-c; (b) presents the fitting of a new ellipse to the contours of the inefficient segmentation performed by the proposed algorithm; (c) the result of filling the ellipse; (d) illustrates ground truth used as an evaluative metric.

Table 11 makes a comparison between the segmentations present in Figure 9 with and without the post-processing technique.

According to Table 11, it can be observed that the use of post-processing improved the segmentation in terms of accuracy, dice, and Jaccard metrics compared to the respective ground truth, but a worsening in precision. This is because a uniform ellipse is generated in the contours, which ends up discarding some valid contour points obtained by the failed segmentation.

For a general comparison, Table 12 presents the result of segmentation obtained by the altered network, according to processes described in Section V-D.

According to the results shown in Table 12, it is clear that post-processing improved the performance of the segmentation.

F. ALGORITHM IMPLEMENTATION FOR HC CALCULATION

The challenge requires the developing of an algorithm to find five values for each ellipse generated by the segmentation:

1) center x: comprises the value of the distance in millimeters from the initial pixel on the x axis of the image to the pixel of the center of the ellipse;

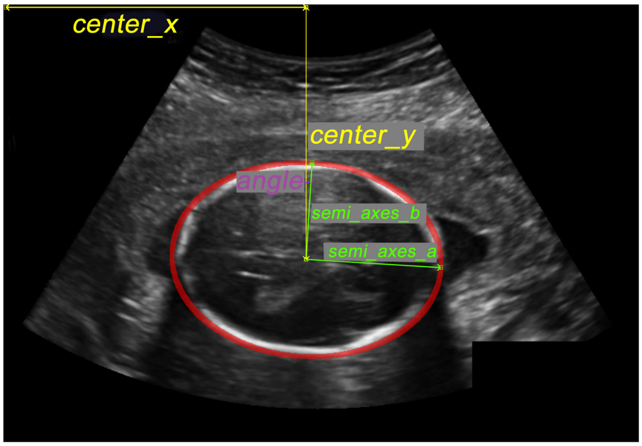


FIGURE 10. Illustration of the necessary measures.

2) center y: comprises the value of the distance in millimeters from the initial pixel of the image's y axis to the pixel of the center of the ellipse;
3) semi axes a: Once the center of the ellipse is found, the semi-axis measure the largest value of the radius from the distance from the center of the ellipse to its farthest point;
4) semi axes b: comprises the smallest value of the radius, from the distance from the center of the ellipse to its nearest point;
5) angle: comprises the value of the angle in radians between the center vector y and the semi-axis vector b.

Figure 10 illustrates how such values are determined in a image:

VI. COMPARATIVE ANALYSIS

In this section, we performed a comparative analysis of the results achieved by the new methodology developed to tune the FCN. Initially, we present Figure 11, which illustrates the increase in the accuracy considering each step covered in the Section V-D.

According to Figure 11, one can observe an evolution in the network performance after each modification proposed in its architecture. The increase in the set of synthetic images generated by data augmentation and the increase in network depth made a significant contribution to its performance.

We also compared our results with three other approaches that created solutions for segmentation and measurement of fetal skulls in two-dimensional ultrasound images. Table 13 shows a comparison of our method with other results reported in the literature.

It is possible to observe the promising results obtained by our proposal when compared to the state-of-the-art methods. The proposed system outperforms the others regarding the dice metric. In terms of mean difference, our results are below the work from Heuvel *et al.* [18]. However, they remain superior in terms of the dice performing 97.92% against 97.0%.

In terms of mean absolute difference, our work presented superior results. Sobhaninia *et al.* [19] reported a mean Abs

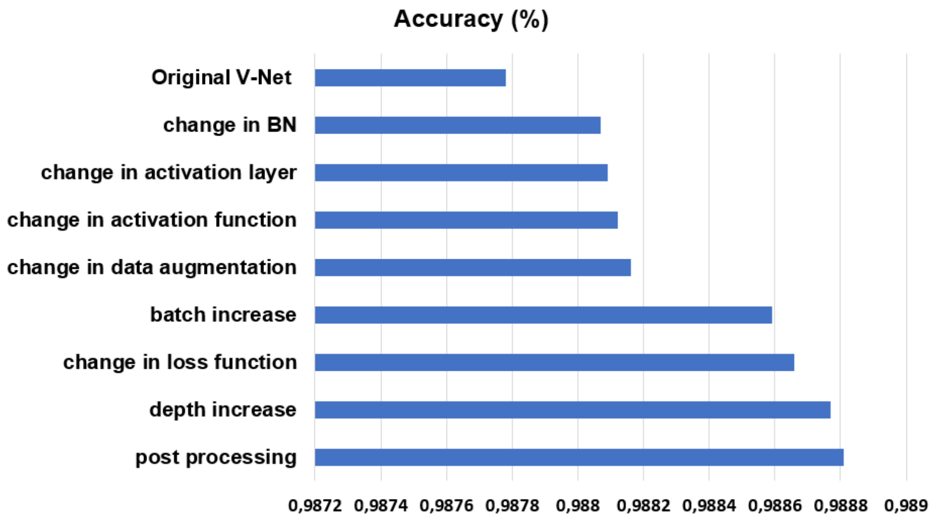


FIGURE 11. Evolution of the accuracy value according to the changes imposed on the network architecture.

TABLE 13. Comparative with State-of-Art methods.

Method	Dice	Mean Difference	Mean Abs difference
Proposed method	**97.92**	0.74 ± 2.41	**1.89 ±1.67**
Heuvel et. al. [18]	97.00	**0.60 ± 4.3**	2.80 ± 3.30
Sobhaninia et. al. [19]	96.84	1.13 ± 2.69	2.12 ± 1.87
Sobhaninia et. al. [21]	93.75	1.53	2.27

TABLE 14. Comparative results among V-Net and VNet-c.

Architecture	M.A.D. (mm) ± std	M.Di.(mm) ± std	M.D. (mm) ± std
VNet-c	1.89 ± 1.67	90.62 ± 2.45	0.74 ± 2.41
Vnet	2.18 ± 2.37	90.63 ± 2.45	0.83 ± 3.12

Difference of 2.12±1.87 mm, which is larger compared to our result.

Finally, we submitted CSV file containing all five fetal head measurements (as described in Section V-F) for automatic evaluation with the challenge metrics. Table 14 shows the values obtained by the original V-Net and VNet-c. The acronyms are: M.A.D. - Mean Abs difference, M.Di. - Mean Dice, and M.D.- Mean Difference

Table 14 depicted that the use of the proposed methodology, and the changes in the network architecture, were responsible for the performance gain.

VII. CONCLUSION

The use of ultrasound is essential for the measurement of fetal biometry during the gestation process. However, manual assessment of measurements is subjective and largely depends on the experience of the evaluator. In this sense, it is necessary to use computational techniques to obtain better results.

Given the problem, this paper presented the adaptation of a Deep Learning-based computational method for the segmentation of fetal skulls in two-dimensional ultrasound images, called Fully Convolutional Neural Network - Combination (VNet-c). A methodology was proposed for the adaptation of a completely convolutional neural network to increase its performance capability.

It was used as base the V-Net network, which was designed for the segmentation of 3D images. Many architecture changes were made to increase its performance and mitigate the overfitting, such as modifying batch normalization, changing activation functions, using of the dropout technique, and using a recent loss function instead of dice loss. Due to the small number of dataset images, changes were also made using data augmentation, increasing the number of batches. Also, the network depth was increased to the limit supported by the graphics card. Subsequently, with some of the resulting inaccurate segmentation an algorithm was created to perform post-processing and improve the quality of the segmentation.

We highlight that in the new VNet-c, two unprecedented stages were developed, a pre-processing to fill the ellipses, and a post-processing usign the technique to the output images.

The results obtained by the proposed methodology showed that it is possible to adapt the architecture of the Completely Convolutional Neural Network V-Net, which was designed for the segmentation of 3D images, to be used for 2D images. Our method obtained better results than other state-of-art methods, reaching up to 97.92% of correct segmentation.

The measurement values resulting from the segmentation stage can be used in the future as an auxiliary tool for medical specialists. Besides, further investigation can be developed with a more powerful computer.

REFERENCES

[1] S. Rueda *et al.*, "Evaluation and comparison of current fetal ultrasound image segmentation methods for biometric measurements: A grand challenge," *IEEE Trans. Med. Imag.*, vol. 33, no. 4, pp. 797–813, Apr. 2014.

[2] L. Wu, J.-Z. Cheng, S. Li, B. Lei, T. Wang, and D. Ni, "FUIQA: Fetal ultrasound image quality assessment with deep convolutional networks," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1336–1349, May 2017.

[3] A. M. Al-Karmi, M. A. Dinno, D. A. Stoltz, L. A. Crum, and J. C. Matthews, "Calcium and the effects of ultrasound on frog skin," *Ultrasound Med. Biol.*, vol. 20, no. 1, pp. 73–81, Jan. 1994.

[4] N. M. Zayed, A. M. Badwi, A. Elsayad, M. S. Elsherif, and A.-B.-M. Youssef, "Wavelet segmentation for fetal ultrasound images," in *Proc. 44th IEEE Midwest Symp. Circuits Syst. (MWSCAS)*, vol. 1, Aug. 2001, pp. 501–504.

[5] G. Carneiro, B. Georgescu, S. Good, and D. Comaniciu, "Detection and measurement of fetal anatomies from ultrasound images using a constrained probabilistic boosting tree," *IEEE Trans. Med. Imag.*, vol. 27, no. 9, pp. 1342–1355, Sep. 2008.

[6] R. C. Sanders and A. E. James, *The Principles and Practice of Ultrasonography in Obstetrics and Gynecology*. New York, NY, USA: Appleton, 1985.

[7] S. M. G. V. B. Jardim and M. A. T. Figueiredo, "Segmentation of fetal ultrasound images," *Ultrasound Med. Biol.*, vol. 31, no. 2, pp. 243–250, Feb. 2005.

[8] W. Lu, J. Tan, and R. Floyd, "Automated fetal head detection and measurement in ultrasound images by iterative randomized Hough transform," *Ultrasound Med. Biol.*, vol. 31, no. 7, pp. 929–936, Jul. 2005.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 91–99, 2015.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[12] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[13] Y. Li, R. Xu, J. Ohya, and H. Iwata, "Automatic fetal body and amniotic fluid segmentation from fetal ultrasound images by encoder-decoder network with inner layers," in *Proc. 39th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2017, pp. 1485–1488.

[14] V. Sundaresan, C. P. Bridge, C. Ioannou, and J. A. Noble, "Automated characterization of the fetal heart in ultrasound images using fully convolutional neural networks," in *Proc. IEEE 14th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2017, pp. 671–674.

[15] L. Wu, Y. Xin, S. Li, T. Wang, P.-A. Heng, and D. Ni, "Cascaded fully convolutional networks for automatic prenatal ultrasound image segmentation," in *Proc. IEEE 14th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2017, pp. 663–666.

[16] M. Sinclair, C. F. Baumgartner, J. Matthew, W. Bai, J. C. Martinez, Y. Li, S. Smith, C. L. Knight, B. Kainz, J. Hajnal, A. P. King, and D. Rueckert, "Human-level performance on automatic head biometrics in fetal ultrasound using fully convolutional neural networks," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2018, pp. 714–717.

[17] R. W. Barnard, K. Pearce, and L. Schovanec, "Inequalities for the perimeter of an ellipse," *J. Math. Anal. Appl.*, vol. 260, no. 2, pp. 295–306, Aug. 2001.

[18] T. L. A. van den Heuvel, D. de Bruijn, C. L. de Korte, and B. van Ginneken, "Automated measurement of fetal head circumference using 2D ultrasound images," *PLoS ONE*, vol. 13, pp. 1–20, Aug. 2018.

[19] Z. Sobhaninia, S. Rafiei, A. Emami, N. Karimi, K. Najarian, S. Samavi, and S. M. Reza Soroushmehr, "Fetal ultrasound image segmentation for measuring biometric parameters using multi-task deep learning," 2019, *arXiv:1909.00273*. [Online]. Available: http://arxiv.org/abs/1909.00273

[20] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.

[21] Z. Sobhaninia, A. Emami, N. Karimi, and S. Samavi, "Localization of fetal head in ultrasound images by multiscale view and deep neural networks," in *Proc. 25th Int. Comput. Conf., Comput. Soc. Iran (CSICC)*, Jan. 2020, pp. 1–5.

[22] R. Qu, G. Xu, C. Ding, W. Jia, and M. Sun, "Standard plane identification in fetal brain ultrasound scans using a differential convolutional neural network," *IEEE Access*, vol. 8, pp. 83821–83830, 2020.

[23] Z. Wang, Z. Zhang, J. Zheng, B. Huang, I. Voiculescu, and G.-Z. Yang, "Deep learning in medical ultrasound image segmentation: A review," 2020, *arXiv:2002.07703*. [Online]. Available: https://arxiv.org/abs/2002.07703

[24] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[27] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, *arXiv:1207.0580*. [Online]. Available: http://arxiv.org/abs/1207.0580

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. image Comput. Comput.-Assist. Intervent.* Cham, Switzerland: Springer, 2015, pp. 234–241.

[29] D. Powers, "Evaluation: From precision, recall and F-factor to ROC," Informedness, Markedness & Correlation, School Inform. Eng., Flinders Univ., Adelaide, SA, Australia, Tech. Rep. SIE-07-001, 2007.

[30] L. R. Dice, "Measures of the amount of ecologic association between species," *Ecology*, vol. 26, no. 3, pp. 297–302, Jul. 1945.

[31] G. S. Khooshabi, "Segmentation validation framework," in *Proc. ICML*, 2015.

[32] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

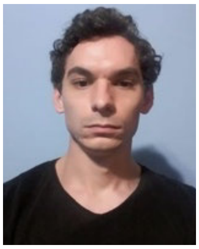
[33] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5927–5935.

[34] H. Lee, S. Tajmir, J. Lee, M. Zissen, B. A. Yeshiwas, T. K. Alkasab, G. Choy, and S. Do, "Fully automated deep learning system for bone age assessment," *J. Digit. Imag.*, vol. 30, no. 4, pp. 427–441, Aug. 2017.

[35] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, "Tversky loss function for image segmentation using 3D fully convolutional deep networks," in *Proc. Int. Workshop Mach. Learn. Med. Imag.* Cham, Switzerland: Springer, 2017, pp. 379–387.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[37] M. Berman, A. R. Triki, and M. B. Blaschko, "The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4413–4421.



EVERTON LEONARDO SKEIKA received the B.Sc. and M.Sc. degrees in computer science from the Federal University of Technology—Paraná (UTFPR), Ponta Grossa, Brazil, in 2017 and 2019, respectively. His research interests include machine learning, computer vision, image processing, and neural networks.

MATHIAS RODRIGUES DA LUZ received the B.Sc. degree in electrical engineering from the Federal University of Technology—Paraná (UTFPR), Brazil, in 2020, where he is currently pursuing the master's degree with the Graduate Program in Electrical Engineering.

BRUNO JOSÉ TORRES FERNANDES (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the Federal University of Pernambuco, Recife, Brazil, in 2007, 2009, and 2013, respectively. He is currently an Associate Professor with the University of Pernambuco, where he received the title of Livre-Docente at the end of 2017. He is also a Coordinator of the Graduate Program in Computer Engineering (master's and Ph.D.), UPE, and the Computer Vision Laboratory, Instituto de Inovação Tecnológica (IIT -UPE), and the Head of the Pattern Recognition and Digital Image Processing Research Group, UPE. His research interests include machine learning, computer vision, image processing, and neural networks. He was a recipient of awards, including the 2008 Google Academic Prize as the Top M.Sc. Student in the Federal University of Pernambuco and the Science and Technology Award for Outstanding Research in the Polytechnic School at the University of Pernambuco, in 2011 and 2017.

HUGO VALADARES SIQUEIRA received the B.Sc. degree in electrical engineering from the State University of São Paulo, in 2006, and the master's and Ph.D. degrees from the University of Campinas, in 2009 and 2013, respectively. He realized his first Postdoctoral stage at the University of Campinas, in 2014, and the second at Illinois State University, USA, and at the University of Pernambuco, in 2017. He has worked on electric engineering and computer science, acting on the following subjects: neural networks, evolutionary algorithms, immune algorithms, swarm intelligence, time series forecasting, pollutant impact on human health, and clustering tasks, among others. He is currently an Adjunct Professor with the Federal University of Technology—Paraná (UTFPR). He is also an Adviser of the Graduate Programs in Computer Science (PPGCC) and Production Engineering (PPGEP). He is a Coordinator of the Interdisciplinary Group of Computational Intelligence and the Laboratory of Computational Intelligence and Advanced Control (LICON), UTFPR-PG.

MAUREN LOUISE SGUARIO COELHO DE ANDRADE received the B.Sc. degree in informatics from The State University of Ponta Grossa (UEPG), in 1995, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Federal University of Technology—Paraná (UTFPR), Curitiba, Brazil, in 2011 and 2015, respectively. She has worked on computer science, acting on the following subjects: neural networks, image processing, and computer graphics. She is currently an Adjunct Professor with the Federal University of Technology—Paraná (UTFPR). She is also an Adviser of the Graduate Programs in Computer Science (PPGCC). She is a Coordinator of the Computer Science degree at the Federal University of Technology—Paraná (UTFPR).