GUDU: Geometrically-constrained Ultrasound Data augmentation in U-Net for echocardiography semantic segmentation

Christoforos Sfakianakis

Thesis submitted in partial fulfillment of the requirements for the

Masters' of Science degree in Computer Science and Engineering

University of Crete
School of Sciences and Engineering
Computer Science Department
Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Emeritus Professor *Georgios Tziritas*

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

University of Crete
Computer Science Department

GUDU: Geometrically-constrained Ultrasound Data augmentation in U-Net for echocardiography semantic segmentation

Thesis submitted by
Christoforos Sfakianakis
in partial fulfillment of the requirements for the
Masters' of Science degree in Computer Science

THESIS APPROVAL

Author: _____
Christoforos Sfakianakis

Committee approvals: _____
Georgios Tziritas
Emeritus Professor, Thesis Supervisor

_____
Antonios Argyros
Professor, Committee Member

_____
Nikolaos Komodakis
Assistant Professor, Committee Member

Departmental approval: _____
Polyvios Pratikakis
Associate Professor, Director of Graduate Studies

Heraklion, February 2022

GUDU: Geometrically-constrained Ultrasound Data augmentation in U-Net for echocardiography semantic segmentation

Abstract

Echocardiography is a very important medical examination that helps in the computation of critical heart functions. Boundary identification, segmentation and estimation of the volume of key parts of the heart, especially the left ventricle, is an important but difficult and time-consuming process, even for the most experienced cardiologists, due to shadows and speckle noise that characterize ultrasound images. In recent years, research has focused on the automatic segmentation of heart through artificial intelligence techniques and especially with the use of deep learning. Our work is part of this research framework.

We implemented a neural network based on U-Net and trained it, using a large public dataset of cardiac ultrasound images (CAMUS dataset), to extract the areas of the left ventricle, myocardium and left atrium. In order to optimize the training process, we have developed a data augmentation method based on the medical practice in echocardiography.

The evaluation of our method by the independent platform of the public competition CAMUS, showed an overall improvement in the segmentation accuracy but also in the estimation of the volume and the ejection fraction of the left ventricle. Specifically using the metric Dice for geometric metrics, the performance of our method for the epicardium reached 0.956 for the end-diastolic phase and 0.950 for the end-systolic phase. For the clinical metrics of the left ventricle volume, the Pearson correlation coefficient was used where our method gave 0.973, 0.974, 0.871 for the end-diastolic, end-systolic phase and ejection fraction respectively.

GUDU: Γεωμετρικά προσδιορισμένη αύξηση δεδομένων για την σημασιολογική τμηματοποίηση εικόνων υπερήχου καρδιάς με χρήση συνελικτικών νευρωνικών δικτύων

Περίληψη

Η υπερηχοκαρδιογραφία είναι μια πολύ σημαντική ιατρική εξέταση που βοηθάει στον υπολογισμό κρίσιμων καρδιακών λειτουργιών. Η οριοθέτηση, η τμηματοποίηση και ο υπολογισμός του όγκου των βασικών μερών της καρδιάς και ιδιαίτερα της αριστερής κοιλίας είναι μια σημαντική αλλά δύσκολη και χρονοβόρα διαδικασία, ακόμα και για τους πιο έμπειρους καρδιολόγους, λόγω των σκιών και του αυξημένου κοκκώδη θορύβου που χαρακτηρίζουν τις εικόνες υπερήχου. Τα τελευταία χρόνια έχει στραφεί η έρευνα στην αυτόματη τμηματοποίηση των μερών της καρδιάς μέσω τεχνικών τεχνητής νοημοσύνης και ειδικά με την χρήση της βαθιάς μάθησης. Σε αυτό το πλαίσιο εντάσσεται η δουλειά μας.

Έχουμε υλοποιήσει ένα τεχνητό συνελικτικό νευρωνικό δίκτυο και το έχουμε εκπαιδεύσει, χρησιμοποιώντας ένα μεγάλο δημόσιο σύνολο εικόνων υπερήχου καρδιάς (τράπεζα CAMUS), ώστε να εξάγει τις περιοχές της αριστερής κοιλίας, του μυοκαρδίου και του αριστερού κόλπου. Για την καλύτερη και εξειδικευμένη εκπαίδευση του, έχουμε αναπτύξει μια αύξηση δεδομένων βασιζόμενοι στην ιατρική πράξη της υπερηχοκαρδιογραφίας.

Η αξιολόγηση της μεθόδου μας από την ανεξάρτητη πλατφόρμα του δημόσιου διαγωνισμού CAMUS, έδειξε σημαντικά ποσοστά βελτίωσης στην τμηματοποίηση αλλά και στον υπολογισμό του όγκου και του κλάσματος εξώθησης της αριστερής κοιλίας. Συγκεκριμένα χρησιμοποιώντας την μετρική Dice για τα γεωμετρικά μεγέθη, η επίδοση της μεθόδου μας για το επικάρδιο έφτασε στο 0.956 για την διαστολική φάση και 0.950 για την συστολική. Για τα κλινικά μεγέθη του όγκου της αριστερής κοιλίας χρησιμοποιήθηκε ο συντελεστής συσχέτισης Pearson όπου η μέθοδος μας απέδωσε 0.973, 0.974, 0.871 για την διαστολική φάση, τη συστολική φάση και το κλάσμα εξώθησης αντίστοιχα.

Acknowledgements

I would like to thank my supervisor Prof. Georgios Tziritas for his guidance throughout all of my studies. His support, the inspired discussions between us, were instrumental for the completion of my thesis.

I would also like to thank my colleague and PhD candidate, Georgios Simantiris for his help and support in programming tools.

Last but not least, I would like to thank my wife Georgia Chronaki for her unconditional love and support.

στη μνήμη του Γιώργου Γκουντουλογιάννη (Gudu)

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 General

Cardiovascular diseases (CVDs) have become the leading cause of death in industrialized countries [5]. Major advancements in cardiovascular research and practice have been made in recent decades, with the goal of improving heart illness detection and treatment, as well as lowering CVD mortality. Modern medical imaging techniques, such as magnetic resonance imaging (MRI), computed tomography (CT), and ultrasound (US), are now widely used because they allow for non-invasive qualitative and quantitative assessment of cardiac anatomical structures and functions, as well as support for diagnosis, disease monitoring, treatment planning, and prognosis.

Even with the advancement of new technology, the ultimate choice on analysis is heavily reliant on operator expertise. Diagnostic mistakes are a significant unsolved issue. Furthermore, not only may cardiologists disagree on picture interpretation, but the same observer might get various conclusions when a reading is repeated. High workloads in clinical practice on a daily basis may contribute to this inaccuracy, and all cardiologists require accurate perception in this sector [19]

Cardiac image segmentation is a critical initial step and can help cardiologists in their observations and decisions in a variety of applications. Separates the picture anatomically into a number of semantically significant sections from which quantitative measurements such as myocardial mass, wall thickness, left ventricle (LV) and right ventricle (RV) volume, ejection fraction (EF), and so on may be retrieved. The LV, RV, left atrium (LA), right atrium (RA), and coronary arteries are typically the anatomical features of interest for cardiac image segmentation. Figure 1.1 shows an overview of typical activities related to cardiac image segmentation, including applications for the three most often utilized modalities, MRI, CT, and ultrasound. In our work, we focus on Cardiac ultrasound imaging, commonly known as echocardiography, a vital clinical technique for evaluating cardiovascular function. Because of its mobility, low cost, and real-time capabilities,

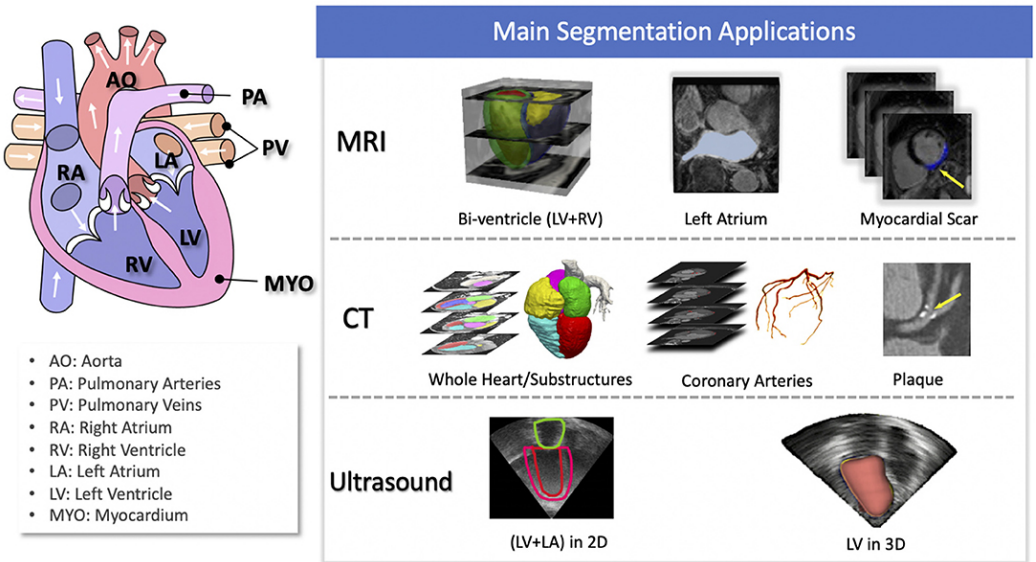


Figure 1.1: Overview of cardiac image segmentation tasks for different imaging modalities. Figure source: "Deep Learning for Cardiac Image Segmentation: A Review" [5]

it is frequently employed as the first imaging assessment in clinical settings. Traditional methods such as active contours, level-sets, and active shape models have been used to automate the segmentation of anatomical structures in ultrasound images [26], but the achieved accuracy is limited by various ultrasound imaging problems such as low signal-to-noise ratio, varying speckle noise, low image contrast (especially between the myocardium and the blood pool), edge dropout, and shadows cast by structures such as dense muscle. This characteristic is common to all kind of echocardiographic views (Figure 1.2), and is crucial to B-mode view because is the most usual in clinical practice (B-mode in known as 2D view).

Deep Convolutional Neural Networks (CNNs) have recently obtained cutting-edge outcomes in biomedical image segmentation applications [5] [19]. The U-net design [30] in particular, proven to be irrespective and fully effective and could be extensively implemented with slight or large modifications. The U-net design inspires the majority of the highest performing ventricular segmentation algorithms and is also the base in our proposed segmentation method (see Chapter 2).

1.2 Dataset

One of the challenges in applying deep learning algorithms for ventricle segmentation was the lack of an adequate and accurate dataset for training. However, Leclerc et al. [23] offered a labeled dataset containing a range of echocardiograms

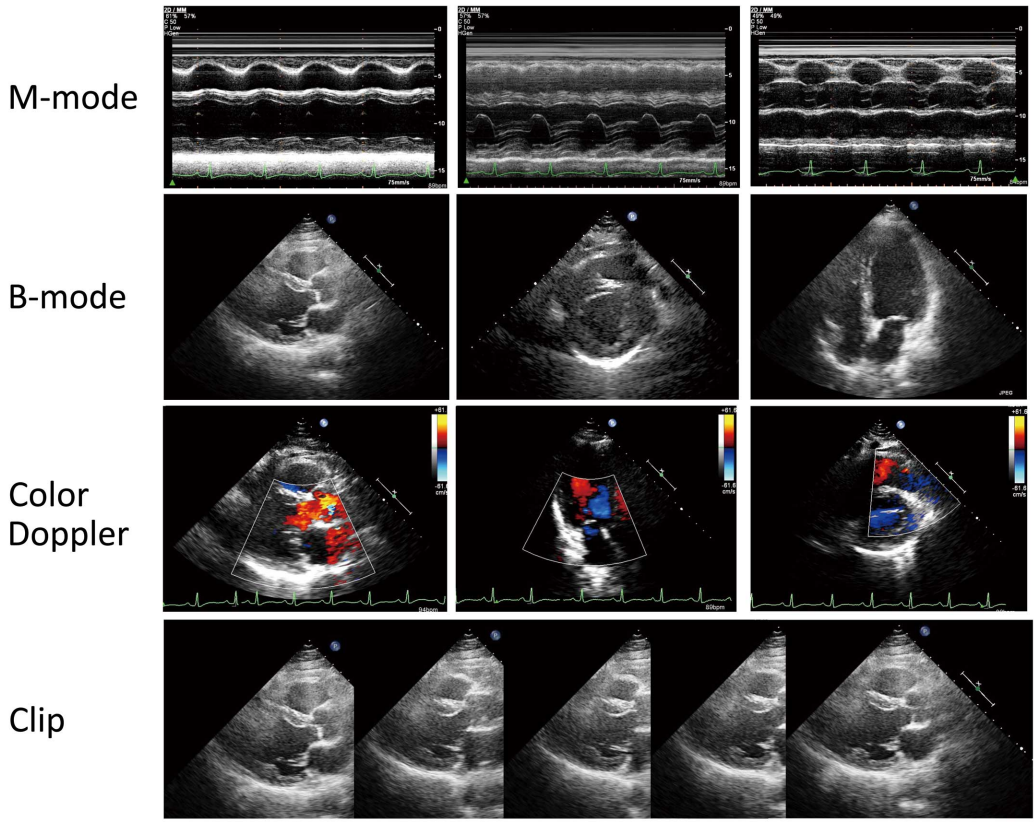


Figure 1.2: "Variety of echocardiographic images needed to be recognized by artificial intelligence systems." Figure source: "Utilization of Artificial Intelligence in Echocardiography" [19]

of varying quality (good, medium, and poor). Furthermore, in order to provide a trustworthy dataset, Leclerc et al. [23] incorporated shadows and dropouts in echocardiograms. Poor quality echocardiograms, on the other hand, were overlooked during the training phase, according to their work. Their CAMUS dataset includes four- and two-chamber acquisitions from 500 patients, as well as manual segmentation (references) of the Left Ventricle (LV_{Endo}), Myocardium (MYO), and Left Atrium (LA). Echocardiographic pictures were obtained using GE Vivid E95 ultrasound scanners equipped with the GE M5S probe. Each patient's apical four-chamber and two-chamber view sequences were exported and manually annotated at end diastole (ED) and end systole (ES). As a result, the CAMUS dataset contains 2000 echocardiographic pictures.

Three cardiologists worked together to manually annotate the 2D echocardiographic pictures in the CAMUS dataset, adhering to the same segmentation procedure.The first cardiologist established a consistent segmentation methodology,

which the other cardiologists followed (Figure 1.3). The annotations of 50 patients from the CAMUS dataset are not available, and their segmentations must be evaluated online by participating in their well organized evaluation platform (see Chapter 3).

1.3 Related work

According to last published systematic review of Deep Learning echocardiography image analysis [10], there is a limitation in the field due to the lack of public datasets. A Google Scholar research in echocardiography image segmentation gives methods and experimental investigations that employ private datasets and is hard to compare our work with other methodologies. So it is only possible to present the state-of-the-art for articles that used the CAMUS Dataset [23] and DICE coefficient metric for evaluation of LV, MYO and ATR segmentation accuracy or the crucial LV Volume in ml (for both End Diastolic and End Systolic phase) and Ejection Fraction in same public data and more specific, in the same test set that is available in CAMUS online platform after the registration. The mentioned last paper Review (2021) [10], presents works that use CAMUS dataset for Left ventricle ejection volume end fraction. Some works uses the non-notated test dataset, we used in our proposed method. Some other uses only a part of training dataset as validation and test set:

Leclerc et al. [23] who published the CAMUS dataset, assessed the extent to which state-of-the-art Deep Convolutional Neural networks (DCNN) Encoder-Decoder methods can evaluate 2D echocardiographic images, that is, segment cardiac structures and estimate clinical indexes in a single data set and they work on two versions of U-Net. U-Net 1 optimized for speed, and U-Net 2 optimized for accuracy. They also implement a Anatomically Constrained Neural Network (ACNN) original published by Oktay et al. [27] fitted to CAMUS dataset. Segmentation results for geometrical and clinical metrics is presented analytically in Tables 3.1, 3.2 discussed in Results Chapter 3 compared with our proposed method results, because, by the time this text is written, these was the only results published in Leader board in CAMUS online evaluation platform.

Leclerc et al. [21] presented a new mechanism of attention for refining the endocardium segmentation and epicardium in 2D echocardiography. The model used two U-Net networks to derive the region of interest from the image before segmentation. The model used parameterized sigmoids to perform threshold operations. The architecture was trained from end-to-end and named Refining U-Net (RU-Net). DICE results were (0.921 ± 0.054) for VE-Endo and (0.948 ± 0.006) for VE-Epi.

Smistad et al. [34] transferred learning from a trained model to segment views from A2C/A4C echocardiographic window data from 106 patients with ALAX vision in conjunction with the CAMUS Dataset, which had 500 patients with A2C/A4C views. However, the results were unsatisfactory, reducing accuracy.

They thus proposed a network with A2C, A4C and ALAX Multi-view segmentation to segment the LV, Myocardium and Atrium, respectively with DICE of (0.921 ± 0.03), (0.786 ± 0.08) and (0.892 ± 0.08).

Leclerc et al. [22] presented a new multi-stage care network to improve the robustness of the segmentation of ECHO 2D LV structures. The network was built around the U-Net architecture and consisted of two stages: The first network extracted the LV region and its mask. The second network used the extracted image to segment the region. The solution's performance was assessed with the most extensive set of current open access 2D echocardiographic data, the CAMUS Dataset. The average Correlation Coefficient result was 0.96 to detect EDV and ESV, and the result for MAE was 7.6 ml. For EF, the correlation coefficient was 0.83 and 5.0 for MAE.

Amer et al. [3] proposed a new method based on Deep Learning called Res-DUnet for LV segmentation and to estimate EF. The model was based on embedded U-Net with extended convolution, where residual blocks were used instead of U-net network units. Result was a DICE of 0.951 ± 0.030.

Zyuzin et al. [40] trained a model by combining the U-Net architecture with Residual Blocks, and the U-net ResNet-34 architecture obtained respective DICE results of 0.9348, 0.9459, 0.9038 for EDV, ESV and EF.

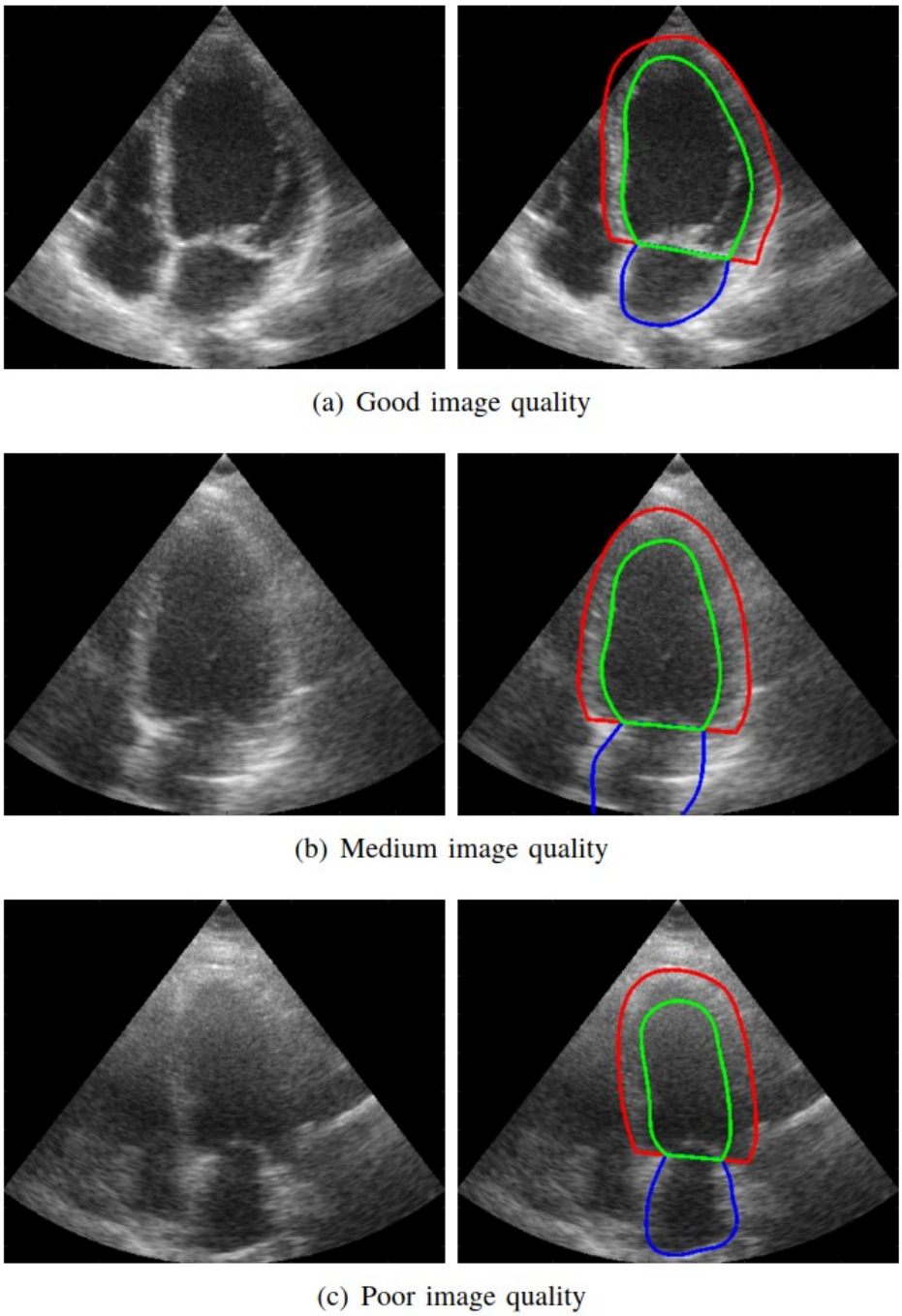
Furthermore search for recently related work, lead us to present papers or articles that uses CAMUS dataset and have an interesting proposal and common characteristics with our work:

Yasser Ali et al. [2] introduced a hybrid net, denoted by ResU (using U-Net and a modified version of Res-Net), and an efficient automatic segmentation approach for echocardigraphic images. They had a random rotation augmentation and the model produced a average DICE score of 0.97 on the testing set for ED and ES.

Fei Liu et al. [24] proposed a novel PLANet method for the semantic segmentation of 2D echocardiographic images. They evaluated the proposed PLANet on the CAMUS dataset and they achieved DICE score 0.951 for ED and 0.931 for ES on the testing set.

Nathan Painchaud et al. [29] proposed apost-processing pipeline to enforce temporal consistency in 2D+time echocardiography segmentation. The temporal consistency is enforced as a constrained regularization on the curves w.r.t. time of seven clinically relevant. The evaluation of their method on CAMUS dataset gave an average DICE Score 0.951.

Truong Dang et al. [9] presented a novel weighted ensemble of deep learning models for the problem of medical image segmentation. The probability predictions by the segmentation algorithms are combined based on weighted combining for a final prediction. The evaluation of their method on CAMUS dataset gave an average DICE Score 0.929 for LV, 0.954 for MYO and 0.935 for ATR.



(a) Good image quality

(b) Medium image quality

(c) Poor image quality

Figure 1.3: Typical images extracted from the CAMUS dataset [23]. Endocardium and epicardium of the left ventricle and left atrium wall are shown respectively in green, red and blue. [Left] input images; [Right] corresponding manual annotations.

Chapter 2

Proposed Learning Method

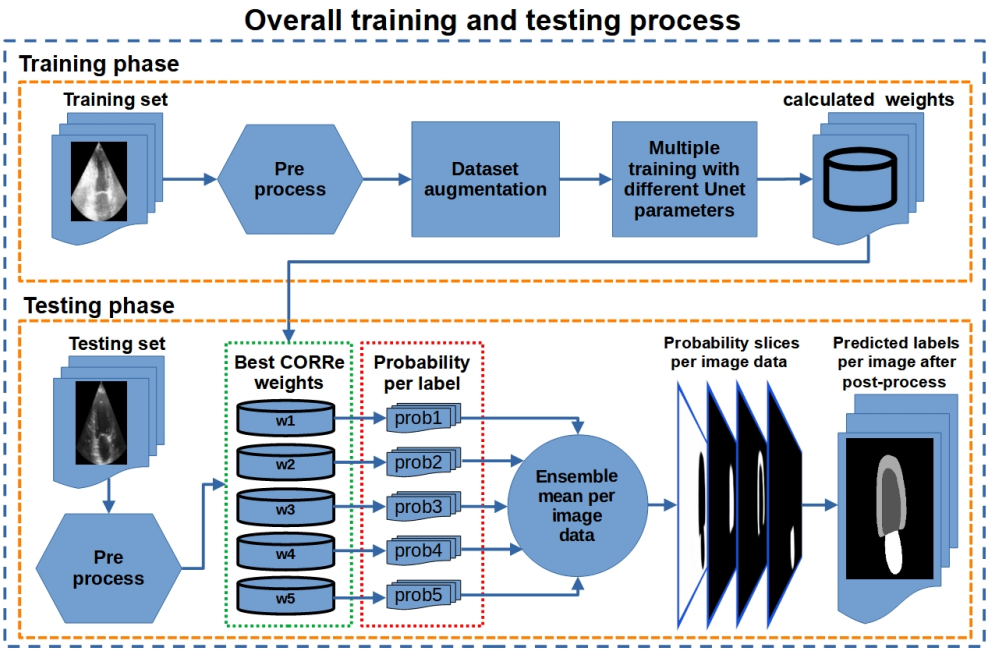


Figure 2.1: Overall training and testing method including the Ensemble Mean prediction schema we used.

2.1 Data pre-processing

CAMUS dataset [23] include images of different resolution. Our training implementation, depends on U-Net architecture, with Convolutional, MaxPooling and

Batch Normalization layers, all of which are invariant to the size of the input image. Also U-Net architecture presupposes input size multiple of 2 [30]. In order to overcome this issue, for every image, we make a new square image with the desired size, resize and paste the old image in the center, keeping the same ratio and pad with zero the non ROI (Region Of Interest) area as shown in Figure 2.2.

This procedure is applied to whole training dataset, including two-chamber (2CH), four-chamber (4CH) view acquisition for both End Diastole (ED), End Systolic (ES) and the corresponding ground truth images.

Considering that dataset contains images with width larger than 512 pixels, we choose Nearest Neighbor Interpolation (NN) for downscaling resampling filter, in resize procedure.

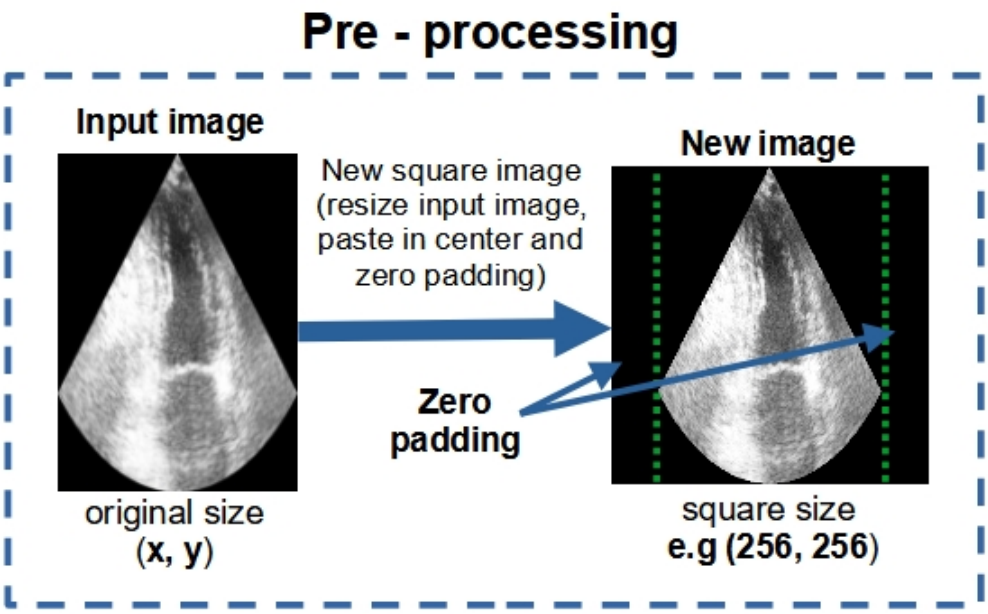


Figure 2.2: Pre-processing for two chamber, four chamber, both ED and ES and the corresponding ground truth images of the dataset.

In order to succeed in all above image size transformations, resize etc, we use the State of the art Python image processing tools, Pillow [7], a friendly Python Imaging Library (PIL) fork by Alex Clark and Contributors and SimpleITK image analysis library [38], a simplified, open-source interface to the Insight Segmentation and Registration Toolkit (ITK). SimpleITK help us to read the original medical image data from the dataset and Pillow to do the necessary transformations. NumPy [14], the fundamental package for scientific computing with Python also used.

2.2 Data augmentation

The importance of data augmentation is commonly accepted, especially for CNN that are heavily reliant on big data to avoid overfitting. A successful data augmentation process can improve both performance and generalization in case of limited training data [32]. There are general and simple techniques in implementation that work in most cases like geometric transformations, color space augmentations, kernel filters, mixing images, random erasing, feature space augmentation, adversarial training and recently generative adversarial networks (GAN) that produce new pseudo realistic images.

We go one step further from the basics and we designed a specialized data augmentation adapted to the echocadiographic image data. The central idea is based on the fact that human hand is involved in echocardiography procedure and is highly subjective the final ultrasound sequence result. Cardiologists and radiologists follows a protocol during the medical examination. There are special recommendations for the echocardiographic assessment of regional LV function, LV mass, LV size etc and probe placement and orientation is highly crucial [20]. Small movements of the ultrasound probe with respect to tissue during the medical examination, can give total different depiction of the cardiac parts (LA, LV, RV, ATR etc). Also low contrast settings in ultrasound scanner can effects the human eye ability to distinguish the borders between the cardiac parts, eg between LV cavity and myocardium.

Based on the above all factors, we designed specifically for the cardiac ultrasound images three novel augmentation techniques, based on a virtual probe orientation and a virtual low-contrast scanner setting that produces an intensity transformation (Figure 2.3).

In order to succeed in three data augmentation techniques, we use the State of the art Python image processing tools, Pillow [7], a friendly Python Imaging Library (PIL) fork by Alex Clark and Contributors and NumPy [14], the fundamental package for scientific computing with Python.

2.2.1 "Cone" random position augmentation

Instead of doing random rotations from the center of the images, we choose a more specific way. We call "cone" the ROI, because of the shape the echo gives to the image. A virtual move of the probe can gives us different augmented images as a result of the rotation from the top of the "cone". This is actually the rotation around the axis, which is perpendicular to the image plane, passing through the beginning of the "cone". We choose 6 random rotations (-3, -6, -9, 3, 6, 9 degrees) in order the ROI to fit to the square frame of the image (Figure 2.4). This virtual probe movement gives new images that are commonly produced via patient clinical examination.

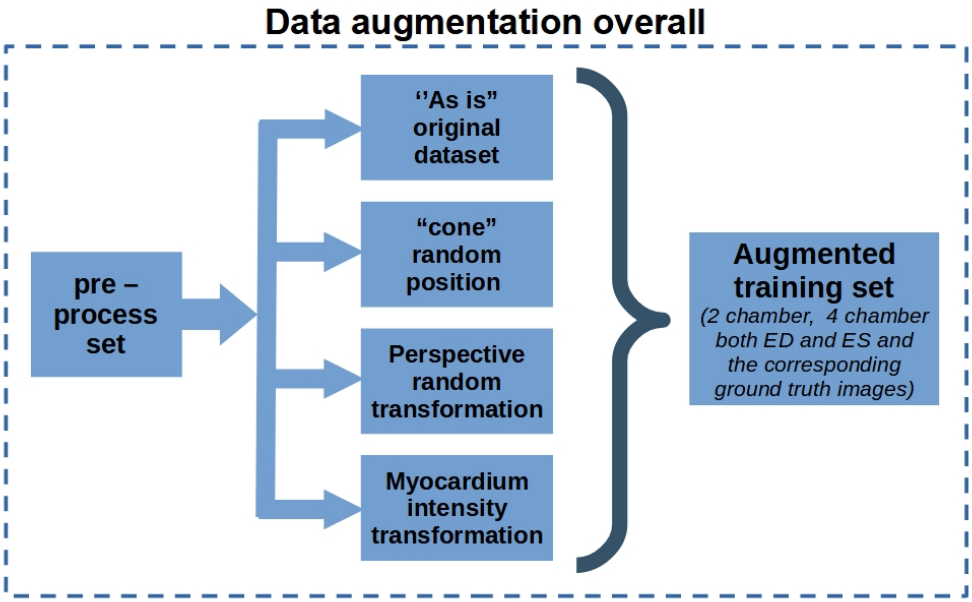


Figure 2.3: Three augmentation techniques

2.2.2 Perspective random position augmentation

As above, we designed a second virtual movement of the probe, a twist around vertical axes. We assume that this twist of the probe corresponds to a shift of the ultrasound projection to a different plain. This small move can give a completely different ultrasound image. (Figure 2.5).

Perspective Transformation data augmentation for object detection has already performed with improvements in performance and the robustness of the detection model on small datasets [36]. In our case we had to mimic images taken at the angle that the camera (probe) moved from the original position in such a way as shown in the figure 2.5. In order the final augmentation result to have a realistic ultrasound image, we used a special designed plane measuring method [8]. In [8] there is a detailed plane to plane homographies development based on the camera model [31]. The mentioned method help us to determine the necessary coefficients needed for performing the perspective transformation by giving the vertices of the current and the resulting plane using a random choose of a plane offset (Figure 2.6). This is actually the rotation around the axis which is vertical on to the image, passing from the middle of the image top side. In case of 256×256 images, the plane pixel offset is a random choose from the set (-50, -40, -30, -20, -10, 10, 20, 30, 40, 50). This method gives us eight possible new augmented images (Figure 2.5).

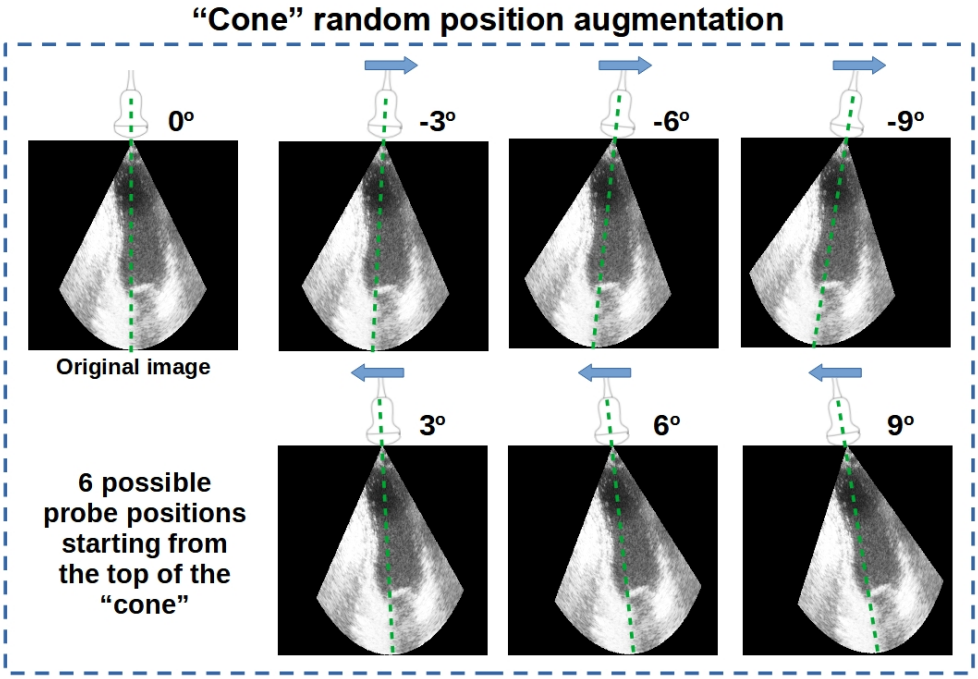


Figure 2.4: Probe virtual movement in a way shown in figure gives as 6 possible augmented images. The top of the "cone" is the center of the random rotation.

2.2.3 Myocardium intensity augmentation

Low contrast settings in ultrasound scanners can change the final image view result and can make it difficult for the human eye to discern between cardiac sections, such as the LV cavity and myocardium. For this type of augmentation we used an already tested method by Simantiris et al. [33] based on intensity transformation that was developed specifically for cardiac MR images, carefully customized in our ultrasound image case. The intensities of the two locations are altered to provide augmented images in order for the network to learn how to accurately identify the left ventricular (LV) cavity from the myocardium. The probability density functions of the two areas, the myocardium and the LV cavity, are first calculated and the intensity contrast between the two distributions is then decreased using the Bhattacharyya distance [4]. Detailed steps and used formulas are in mentioned technique [33]. Figure 2.7 shows a myocardium intensity transformation augmentation example.

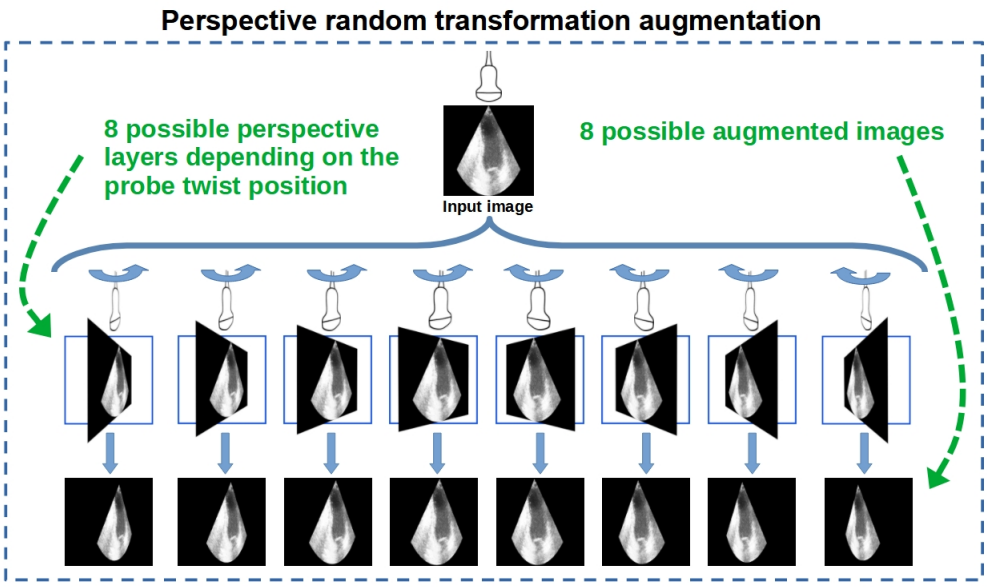


Figure 2.5: Probe virtual twist movement in a way shown in figure gives as 8 possible augmented images.

2.3 Proposed Network Architecture

We propose a neural network based on the original U-Net architecture [30], with the ability to add one more down-sampling and up-sampling layer. U-net design comprises of an encoder path (contracting down-sampling path) and a decoder path (expansive up-sampling path) (Figure 2.8). Convolution operation, max pooling, ReLU activation, concatenation, and up convolution layers make up the network, with the ability to include Batch Normalization and Dropout layers in our implementation. The input image is initially fed into the network. Image data is subsequently sent throughout the network via all conceivable routes. The four probability slices per class will show at the last layer, as an output of the Softmax function. Each block represents a multi-channel feature map, with the spatial dimension of each box shown as the third dimension number. Blue and gray boxes show convolution and Batch Normalization respectively, followed by a nonlinear activation function, a rectified linear unit (ReLU in our case).

Max pooling is next, a type of non-linear down-sampling that decreases the size of a feature map. The number of feature channels is expanded by a factor of two after each max pooling process. Convolution and max-pooling are combined to create a contracting path, which increases the number of feature maps. Furthermore, the resolution of feature maps has been reduced. Usually in common CNN networks, all feature vectors are mapped to a single output vector. To build

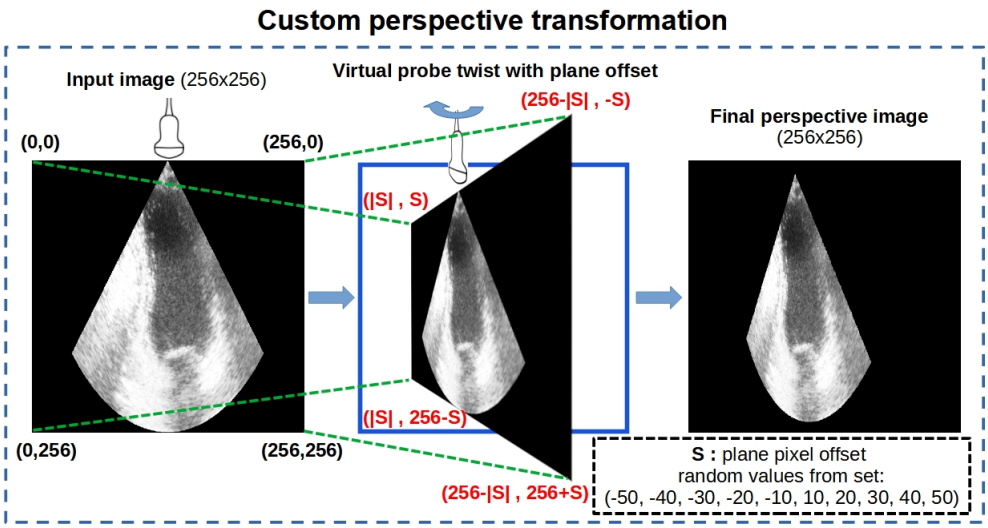


Figure 2.6: Custom perspective transformation based on a virtual probe twist and a plane to plane homographies method (a 256×256 image case paradigm).

a high-resolution segmentation map, the U-net has an extra expansive path. This expansion path is made up of a series of up-convolutions (transposed convolution) and concatenation of features maps obtained from the contracting path.

The up-convolution layer, maps each feature vector to the 2×2 output window using a learning kernel, followed by the activation function (eg ReLU). For better localization and learning representations, higher resolution features are copied from the contracting path to the matching expansion level. By concatenating the output of the transposed convolution layer with the feature maps from the down-sampling at the same level at every block of the up-sampling, we get more accurate localization. The segmentation map in our case consists 4 channels (classes/labels): Background, Left Ventricle, Myocardium, Left atrium. It is computed by a pixel-wise soft-max function over the final feature map.

Our method is based on the ability to use multi-training parameters in training procedure (2.5), including all the crucial network architecture parameters, such as the depth of the U-Net (number of layers), the capability of Dropout and Batch Normalization layers, and the number of filters. This fully parametric availability of the network architecture gives us the ability to determine the lowest resolution in feature maps, the total number of training parameters and increases the flexibility to choose parameters that effects the accuracy and robustness during the training process.

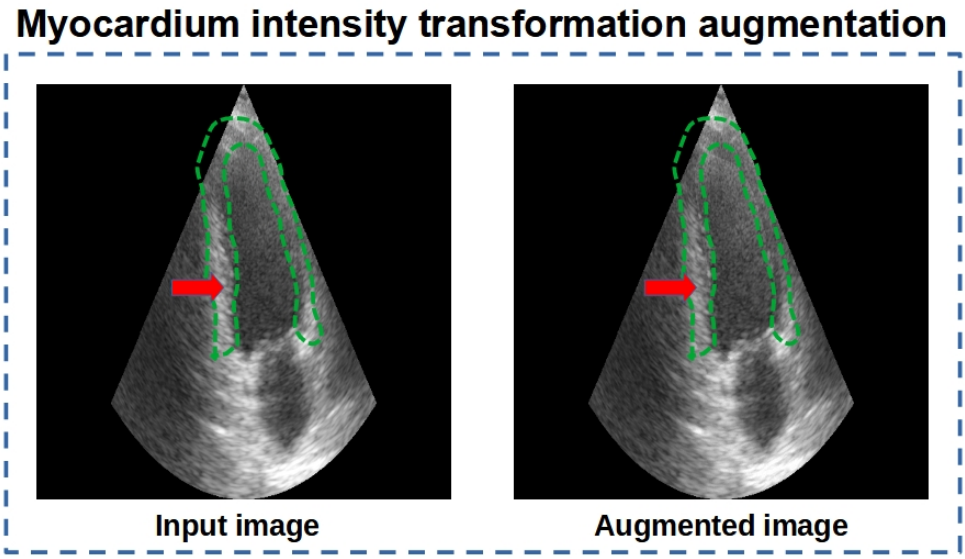


Figure 2.7: Myocardium intensity transformation augmentation paradigm.

2.4 Loss Function

We used 2 different loss functions in our training process (2.5). First the Keras [6] build in Categorical Cross Entropy Loss function and then a custom Dice loss function based on Simantiris et al. [33] tested method, customized for our case.

The Categorical Cross Entropy Loss function formula is:

$$L_{CCE} = -\frac{1}{N}\sum_{i}^{N}\sum_{c=0}^{3} T_{ic}\log(Y_{ic}) \tag{2.1}$$

where N is the number of batch samples, c the label of the class (4 classes), T_{ic} is 1 if sample i is in class c and 0 otherwise and Y_{ic} is the predicted probability that sample i is in class c.

Above definition shows that loss depends only on the probability assigned to the true class and has no dependence on how the rest of the probability is distributed. Cross entropy assumes that all misclassification costs are of equal importance. So for a given sample of class 1 if an algorithm assigns $p_1 = 0.2$ to class 1 then it doesn't matter how the rest of the probability mass is distributed across the rest of the classes, the loss will be the same.

We need a more flexible loss function that is closer to Dice Similarity used for the Accuracy measure and also to give us the ability to penalize misclassifications based on the characteristics of the echocardiography.

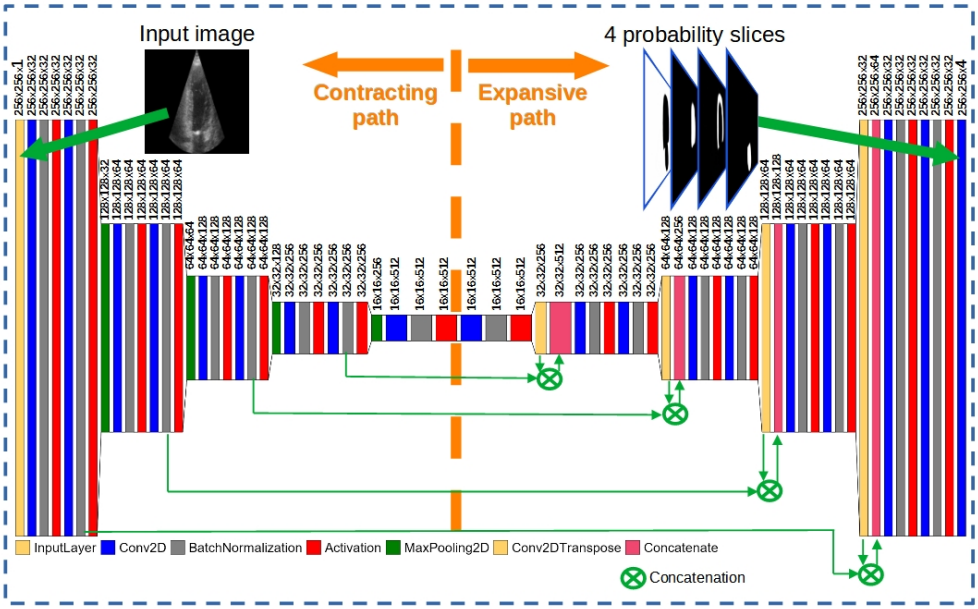


Figure 2.8: Instance of the proposed U-Net architecture, with Batch Normalization layers, without Dropout layers, depth: 5 (No extra block layer), Input image size: 256×256, number of filters: 32. This parameters give us lower feature map size: 16×16 and total number of training parameters: 8M. *The main structure of this figure is automatically designed by VisualKeras [12], a Python package that visualize Keras neural network architectures*

So used a loss function based on the soft Dice similarity:

$$L_D = \sum_{c=0}^{3} \left(1 - \frac{2 \sum_s Y(s,c)T(s,c) + e}{\sum_s (Y^2(s,c) + T(s,c)) + e} \right) \tag{2.2}$$

where the summation on s is over all the pixels of the images in a mini-batch. Y denote the networks prediction and T the ground truth. The c parameter takes values $0, 1, 2, 3$, denoting the labels for Background, Left Ventricle, Myocardium and Left Atrium respectively.

Also we use a method proposed by Simantiris et al [33] to penalize the case of pixels belongs to the Left Ventricle that are neighbors with pixels of the background, because the LV cavity is always surrounded by the myocardium and not adjacent to background. This penalty is simply calculated my the formula:

$$L_{LB} = \sum_{u} \sum_{t \in N(u)} Y(u,1)Y(t,0) \tag{2.3}$$

where the first summation on u is again over all the pixels of each image in the mini-batch for the predictions of label '1' (LV) and the second summation of t is over the immediate neighborhood of pixel u of each image in the mini-batch for predictions of label '0' (Background).

With the same thinking, we penalize the case of pixels belongs to the Left atrium that are neighbors with pixels of the Myocardium:

$$L_{AM} = \sum_{u} \sum_{t \in N(u)} Y(u,3)Y(t,2) \tag{2.4}$$

where also the first summation on u is again over all the pixels of each image in the mini-batch for the predictions of label '3' (ATR) and the second summation of t is over the immediate neighborhood of pixel u of each image in the mini-batch for predictions of label '2' (Myocardium).

Finally the total loss function formula is:

$$L = L_D + \alpha L_{LB} + \beta L_{AM} \tag{2.5}$$

2.5 Training

Training process starts with the augmentation method 2.1 in the original CAMUS training dataset [23] consisting of 450 cardiac ultrasound image sets per patient. Every patient set includes four images, two corresponding to the two-chamber (2CH) view acquisition for both End Diastole (ED) and End Systolic (ES) and two corresponding to the four-chamber (4CH) view acquisition for both End Diastole (ED) and End Systolic (ES). We deal the training process separate for two-chamber (both ED and ES) and separate for four-chamber (both ED and ES). This leads us to two independent training procedures that produces two different calculated weights files, one for the two-chamber training images and a second one for four-chamber (the same in validation/testing process 2.6).

So, for every training procedure, we have 900 original images available (450 for ED and 450 for ES), pre-process them as stated in 2.1 and obtain another 2700 images augmented in three different ways as presented in 2.1 (900 images per every augmentation method). We then distribute them into 9 folds (400 images per fold), making sure each fold approximately contains data from each one of the 3 quality groups (good, medium and poor image quality). Performing 9-fold cross validation scheme for our training, gives us 8-folds for training set (3200 images) and 100 original, not augmented images for validation set. The validation set remains as was (original), so that images in this set were never introduced to the network during the training stage, as original or augmented. Also during training, we noticed that both validation loss and validation accuracy remain in same levels for all validation folds, proving that our augmentation method lead to a stable performance for all the folds. This made us to decide to make the multi – training process with the same validation set (fold 8) in order to speed up the whole process.

For all the training instances, we used the Adam optimizer [18] in order to have the same base algorithm for all the different training parameters. The first part of the Table 2.1 shows the parameters that can change to implement the multi – training schema of our method.

Image size is the input image square dimension after pre-procces (see 2.1). We used 3 different image dimensions, 128×128, 256×256 and 512×512. During validation process, we noticed that 256×256 dimension was most convenient for our H/W resources (RAM, processing time etc) with actually same results as 512×512, so most of the training instances was with this dimension.

Weights balance is mapping class indices (integers) to a weight (float) value, used for weighting the loss function (during training only). This can be useful to tell the model to "pay more attention" to samples from an under-represented class. Actually this parameter has the form $w(c) = [abcd]$ where the $w(0) = a$ is the "attention" value for Background, $w(1) = b$ for LV, $w(2) = c$ for MYO and $w(3) = d$ for ATR, and consists of a multiply parameter to the loss function (eg $w(c)*(2.1)$). In our implementation we have to give more attention to Left Ventricle area, so the $w(1) = b$ is higher in most training cases (eg 1511).

Loss func is the loss function used for training, Categorical cross entropy or Soft Dice. Weights balance works for both loss functions, but Soft Dice has the extra penalty properties described in 2.4.

U-Net depth describes the number of down-sampling block layer including the bottom. Original U-Net architecture has 5 layers depth. We add the capability to have one more down-sampling block in order to further decrease the lowest resolution and to increase the number of feature maps. Increasing the depth, increases the total number of training parameters and the complexity of the total network.

Learning rate (lr) is the value for learning rate in Adam optimizer. We tested only two values 0.001 and 0.0001.

Batch size is the number of samples per gradient update. We tested only two values, 10 and 32 as proposed in Lecrec et al. [23]

Batch norm is the network capability to includes Batch Normalization layers after every Convolutional layer (gray blocks in Figure 2.8)

No filters is an integer that gives the dimensionality of the output space, i.e. the number of output filters in the convolution. In U-Net architecture that we follow in our implementation, this dimensionality is doubling after every down-sampling block and is divisible by two in up-sampling (2.8).

A sample of the learning curves is depicted in Figure 2.9, showing the loss and accuracy history (both for training and validation set). We used Keras [6] capability to early stop the training process after 7 epochs that the validation loss don't improve. This method decreases the total epoch number needed to converge (in about 25 epochs, as shown in Figure 2.9. As mentioned before, all the folds show similar behavior during the training and our network was able to train the whole set with a particular validation fold in approximately 20 minutes for 256 × 256 size images with 32 starting filters on an NVidia Titan V GPU

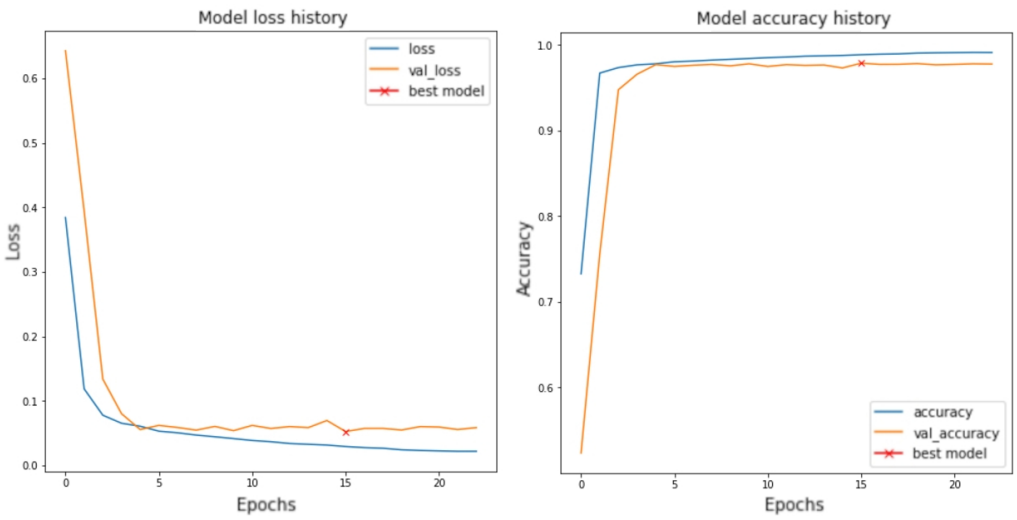


Figure 2.9: Training loss and accuracy for fold 8. *Figure created with Matplotlib python library* [15]

using Keras/Tensorflow ([6], [1]) and about 10 minutes for 16 starting filters. This time doubles with 512×512 size images. We also used Keras capability to store the calculated weights only for the best validation loss instance during converging process (see the red x notation in Figure 2.9). As mentioned before, we used fold 8 for our training method, but all the folds had approximately the same metrics. Table 2.1 shows 20 training instances with approximately the same validation loss and validation accuracy for both 2CH and 4CH (average 0.062, 0.976 for 2CH and average $0,055$, $0,979$ for 4CH respectively).

2.6 Validation

As the evaluation metrics (validation loss, validation accuracy) is pretty the same, for different training parameters, we decided to measure the more morphological characteristics using the calculated weights on the validation set (100 images, 50 2CH and 50 4CH). For this reason, and for the same training instances, through the calculated weights, we measure the pixel-wise Confusion Matrix and 3 clinical indices: i) the ED volume (LV_{ED} in ml), ii) the ES volume (LV_{ES} in ml) iii) the ejection fraction (LV_{EF} as a percentage) for which we computed two metrics: the Pearson correlation coefficient (corr) and the Mean Absolute Error (MAE) (See the second part of Table 2.2).

Confusion Matrix is a summary of prediction results on a classification problem. In our semantic segmentation problem, we have 4 classes available for each pixel (pixel-wise segmentation). So in confusion matrix, the number of correct and

Table 2.1: Training history in the same validation set (fold 8), with twenty training instances of different U-Net parameters. Validation loss and validation accuracy results remain in same levels. This lead us to extract more morphological characteristics.

										Two Chamber		Four Chamber	
	image size	weights balance	loss func	unet depth	lr	batch size	batch norm	No Filters	No params	Val Loss	Val Acc	Val Loss	Val Acc
1	256	1353	Cross entropy	6	1e-4	10	Yes	32	35M	0,079	0,975	0,070	0,979
2	512	1551	Cross entropy	6	1e-3	10	No	32	35M	0,061	0,976	0,050	0,980
3	256	None	Dice	6	1e-3	32	No	16	8M	0,062	0,975	0,058	0,979
4	512	None	Cross entropy	6	1e-3	32	No	16	8M	0,062	0,975	0,051	0,979
5	512	1551	Cross entropy	6	1e-3	32	No	16	8M	0,061	0,975	0,050	0,980
6	512	1941	Cross entropy	6	1e-3	32	No	16	8M	0,060	0,975	0,050	0,979
7	512	1511	Cross entropy	6	1e-3	32	No	16	8M	0,058	0,975	0,052	0,979
8	256	1511	Dice	6	1e-3	32	No	16	8M	0,064	0,975	0,055	0,979
9	256	None	Dice	6	1e-3	32	No	32	35M	0,060	0,976	0,056	0,979
10	256	1511	Dice	6	1e-3	32	No	32	35M	0,060	0,976	0,059	0,979
11	256	None	Dice	5	1e-4	10	No	64	35M	0,063	0,975	0,056	0,979
12	256	1511	Dice	5	1e-4	10	No	64	35M	0,063	0,975	0,055	0,980
13	256	1511	Dice	6	1e-4	10	No	64	135M	0,062	0,976	0,055	0,980
14	256	1511	Dice	6	1e-4	32	No	64	135M	0,062	0,975	0,056	0,979
15	256	1511	Dice	6	1e-3	32	Yes	32	35M	0,056	0,978	0,053	0,981
16	256	1511	Dice	6	1e-3	32	Yes	16	8M	0,061	0,976	0,056	0,980
17	256	1711	Dice	6	1e-3	32	Yes	32	35M	0,057	0,977	0,051	0,981
18	256	1911	Dice	6	1e-3	32	Yes	32	35M	0,058	0,977	0,057	0,980
19	128	None	Dice	5	1e-3	32	Yes	32	8M	0,062	0,976	0,059	0,979
20	128	1511	Dice	5	1e-3	32	Yes	32	8M	0,059	0,976	0,057	0,980

incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix, that it shows the ways in which our classification model is confused when it makes predictions. It gives us insight not only into the errors being made by our classifier but more importantly the types of errors that are being made. This means that overcomes the limitation of using classification loss and accuracy alone. Confusion matrix is calculated from the expected outcomes and predictions by counting the number of correct predictions for each class and the number of incorrect predictions for each class, organized by the class that was predicted. These numbers are then organized into a matrix, where each row corresponds to a predicted class (predicted label) and each column corresponds to an actual class (True label). The counts of correct and incorrect classification are then filled into the table. The total number of correct predictions for a class go into the expected row for that class value and the predicted column for that class value. In the same way, the total number of incorrect predictions for a class go into the expected row for that class value and the predicted column for that class value.

Figure 2.10 shows the Confusion Matrix that corresponds to the 18th calculated

Table 2.2: Validation metrics table. After the calculation of weight files, we work on validation set (fold 8) to produce the Confusion Matrix (table show only LV area Confusion Matrix scores), and Pearson correlation coefficient (corr) and the Mean Absolute Error (MAE) for LV_{ED}, LV_{ES}, LV_{EF}

										Confusion Matrix		CORRe			MAE		
	image size	weights balance	loss func	unet depth	lr	batch size	batch norm	No Filters	No params	LV_{ed}	LV_{es}	LV_{edv}	LV_{esv}	LV_{ef}	LV_{edv}	LV_{esv}	LV_{ef}
1	256	1353	Cross entropy	6	1e-4	10	Yes	32	35M	0,911	0,912	0,970	0,970	0,903	7,478	5,210	3,730
2	512	1551	Cross entropy	6	1e-3	10	No	32	35M	0,935	0,942	0,976	0,974	0,914	6,201	4,203	3,537
3	256	None	Dice	6	1e-3	32	No	16	8M	0,942	0,936	0,969	0,964	0,888	6,543	5,254	4,016
4	512	None	Cross entropy	6	1e-3	32	No	16	8M	0,924	0,932	0,976	0,970	0,912	7,022	4,818	4,089
5	512	1551	Cross entropy	6	1e-3	32	No	16	8M	0,947	0,949	0,978	0,974	0,915	6,002	5,061	4,106
6	512	1941	Cross entropy	6	1e-3	32	No	16	8M	0,939	0,945	0,974	0,971	0,930	6,329	4,699	3,582
7	512	1511	Cross entropy	6	1e-3	32	No	16	8M	0,930	0,940	0,975	0,974	0,915	6,376	4,369	3,616
8	256	1511	Dice	6	1e-3	32	No	16	8M	0,940	0,940	0,969	0,968	0,899	6,514	5,008	4,084
9	256	None	Dice	6	1e-3	32	No	32	35M	0,948	0,944	0,977	0,975	**0,924**	5,912	4,413	**3,446**
10	256	1511	Dice	6	1e-3	32	No	32	35M	0,950	0,945	0,978	0,969	0,904	5,627	3,745	3,794
11	256	None	Dice	5	1e-4	10	No	64	35M	0,939	0,938	0,976	0,969	0,911	6,052	4,512	3,528
12	256	1511	Dice	5	1e-4	10	No	64	35M	0,942	0,942	0,977	0,974	0,920	6,263	4,928	4,163
13	256	1511	Dice	6	1e-4	10	No	64	135M	0,948	0,942	0,974	0,973	0,912	6,276	4,671	3,975
14	256	1511	Dice	6	1e-4	32	No	64	135M	0,951	0,942	0,972	0,970	0,907	6,554	4,845	3,551
15	256	1511	Dice	6	1e-3	32	Yes	32	35M	0,946	0,935	0,980	0,980	**0,932**	5,998	3,702	**3,225**
16	256	1511	Dice	6	1e-3	32	Yes	16	8M	0,929	0,929	0,977	0,977	**0,930**	6,182	3,784	**3,312**
17	256	1711	Dice	6	1e-3	32	Yes	32	35M	0,931	0,931	0,979	0,976	**0,920**	5,730	4,217	**3,453**
18	256	1911	Dice	6	1e-3	32	Yes	32	35M	0,957	0,954	0,978	0,976	**0,914**	5,821	4,624	**3,593**
19	128	None	Dice	5	1e-3	32	Yes	32	8M	0,952	0,949	0,971	0,969	0,911	6,648	4,806	4,075
20	128	1511	Dice	5	1e-3	32	Yes	32	8M	0,944	0,939	0,974	0,966	0,896	5,910	5,249	4,403

weights file of the Table 2.2. The matrix values is based on the total images of the validation set. This batch of images consists 6.553.600 pixels (50 (2CH images) + 50 (4CH images) multiply by 256×256 dimension). The left part of the matrix shows the pixel-wise counts as described before and the right part of the matrix shows the normalized percentage outcomes. It's clear that confusion matrix gives us a more precise view of the segmentation performance.

Clinical metrics, correlation coefficient (corr) and mean absolute error (MAE) for ED, ES volume and ejection fraction percentage, are calculated indirectly by the predicted pixels of the Left Ventricle and the given information of ED, ES volume and ejection fraction true values, by CAMUS training dataset. More clearly, Camus training dataset, for every patient, gives in a configuration file, addition information. This information includes Sex, Age, Image Quality, LV ED Volume (in ml), and LV ES Volume (in ml), calculated by three expert cardiologist [23] and Simpson's rule [11], the simplest and more accurate math formula that produces the Volume of LV, given the 2D dimensions extracted from area notations. LV Ejection Fraction is already calculated by the formula:

$$LV_{EF} = \frac{LV_{EDV} - LV_{ESV}}{LV_{EDV}} \tag{2.6}$$

Given this ground truth LV Volume information in validation set, we use the calculated number of pixels of LV area, after the prediction, and the *rule of three*

model_augmented_256_w1511_dice_extralayer_lr1e3_BS32_bn_NF16_fold8													
CONFUSION MATRIX							NORMALIZED CONFUSION MATRIX						
		Predicted label ED							Predicted label ED				
		BGR	LV	MYO	ATR	sum of pixels			BGR	LV	MYO	ATR	sum
True label	BGR	5.513.735	365	24.812	17.926	5.556.838	True label	BGR	0,99224	0,00007	0,00447	0,00323	1,00000
	LV	256	353.042	25.483	1.200	379.981		LV	0,00067	0,92910	0,06706	0,00316	1,00000
	MYO	33.728	16.262	372.175	99	422.264		MYO	0,07987	0,03851	0,88138	0,00023	1,00000
	ATR	9.175	3.701	201	181.440	194.517		ATR	0,04717	0,01903	0,00103	0,93277	1,00000
		Predicted label ES				6.553.600			Predicted label ES				
		BGR	LV	MYO	ATR	sum of pixels			BGR	LV	MYO	ATR	sum
True label	BGR	5.471.422	540	32.531	20.764	5.525.257	True label	BGR	0,99026	0,00010	0,00589	0,00376	1,00000
	LV	161	363.846	26.516	1.029	391.552		LV	0,00041	0,92924	0,06772	0,00263	1,00000
	MYO	34.228	20.224	388.657	49	443.158		MYO	0,07724	0,04564	0,87702	0,00011	1,00000
	ES	10.581	4.687	169	178.196	193.633		ATR	0,05464	0,02421	0,00087	0,92028	1,00000

Figure 2.10: Confusion Matrix that corresponds to the 16th calculated weights file of the Table 2.2 obtained by validation set (100 images of 256×256 dimension, 6.553.600 total pixels).

to calculate the LV Volume in prediction images (both for ED and ES). We simply assume that number of pixels of LV area correlates with the corresponding Volume (in ml) according to Simpson rule. So, the rule of three gives us the formula to calculate the LV ED Volume (in ml) in image prediction:

$$LV_{EDV_{pred}} = \frac{LV_{EDV_{gt}} * LV_{EDV_{pred-pixels}}}{LV_{EDV_{gt-pixels}}} \tag{2.7}$$

where $LV_{EDV_{gt}}$, is the given ground truth LV ED Volume (in ml), $LV_{EDV_{pred-pixels}}$ is the number of predicted pixels with LV ED label (LV ED predicted area) and $LV_{EDV_{gt-pixels}}$ is the number of ground truth pixels with LV ED label (LV ED ground truth area).

Accordingly, the formula to calculate the LV ES Volume (in ml) in image prediction:

$$LV_{ESV_{pred}} = \frac{LV_{ESV_{gt}} * LV_{ESV_{pred-pixels}}}{LV_{ESV_{gt-pixels}}} \tag{2.8}$$

where $LV_{ESV_{gt}}$, is the given ground truth LV ES Volume (in ml), $LV_{ESV_{pred-pixels}}$ is the number of predicted pixels with LV ES label (LV ES predicted area) and $LV_{ESV_{gt-pixels}}$ is the number of ground truth pixels with LV ES label (LV ES ground truth area).

So, we can finally calculate the percentage Ejection Fraction for the image prediction with the general formula 2.6.

The above formulas give us the capability to calculate the *Pearson Correlation coefficient (CORRe)* (Columns 13-15, Table 2.2), between the given ground truth LV Volumes (ED and ES) and more important the CORRe between the real LV ejection fraction and the predicted, for the whole validation set (50 (2CH images) + 50 (4CH images)).

Accordingly, we have the capability to measure the *MAE* (Last 3 columns, Table 2.2), between the given ground truth LV Volumes (ED and ES) and more important the *MAE* between the real LV ejection fraction and the predicted, for the whole validation set.

2.6.1 Ensemble prediction

After the calculation of *CORRe* and *MAE*, the next step to our ensemble method is to choose the five best performance calculated weight files. The first criterion is the highest CORRe of LV_{ef} and the second criterion is the lowest MAE of LV_{ef} performed to the validation fold (see underline numbers in Table 2.2).

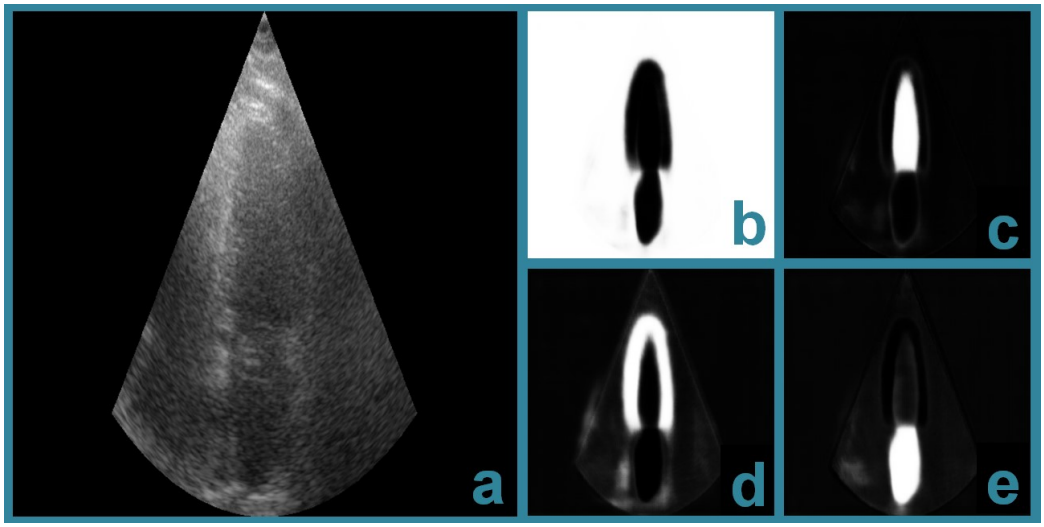


Figure 2.11: Visualization of calculated class probabilities for the 27th test image. (a) The input test image, (b) the probabilities of Background class, (c) the probabilities of Left Ventricle class, (d) the probabilities of Myocardium class, (e) the probabilities of Left Atrium.

Every calculated weight file is used for computing pixel-wise probabilities for the four classes. This is actually the output of our model (2.8) where the final blog consists 4 slices, one for each class, with the calculated probabilities for every pixel in 2-D.

Prediction process for each image starts with the calculation of these probabilities (2.11) and ends with labeling the maximum probability per pixel (argmax function).

Instead of computing the labels from one prediction, we use the five best calculated weights (bold notation in Table 2.2), to produce five different probabilities per class. This gives us the ability to generalize the final probability prediction per pixel by taking the mean of probabilities per class. Mean probability per class

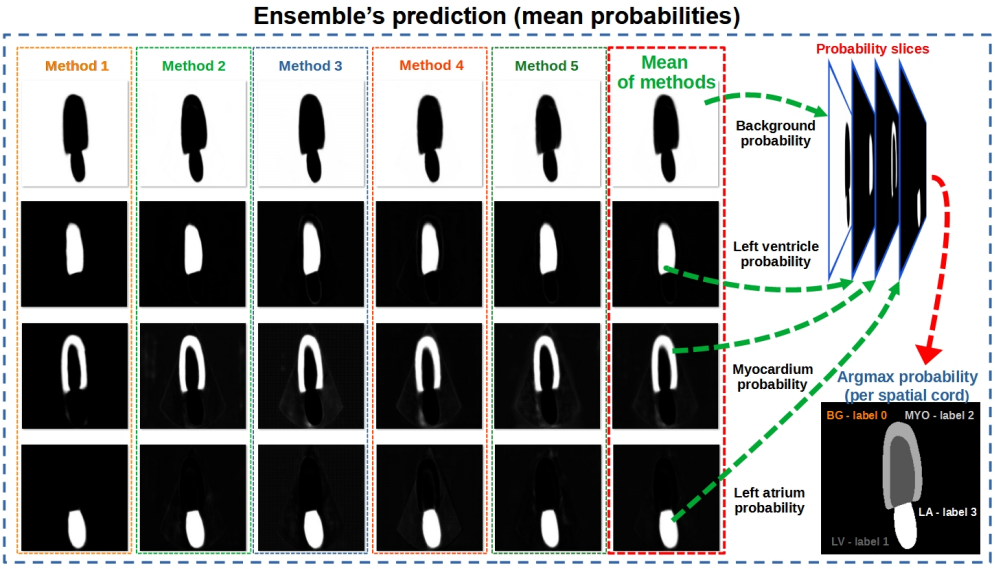


Figure 2.12: Proposed Ensemble's prediction method: Five best performance calculated weights, give five different, but close, probabilities per class. We take the mean probability per class to predict with argmax the labels.

gives us the final segmentation map by labeling the maximum probability per pixel (Figure 2.12)

2.6.2 Method issues

Before going to Post-processing section we can determine some issues that came up from Table 2.2 and Confusion Matrix paradigm (2.10).

The first issue is the very high number of training parameters produced by adding one more layer in U-Net model or by increasing the initial number of filters in entry level of the model. Although this seems to give better performance in validation clinical metrics (e.g. lines 9, 15, 17, 18 of Table 2.2, with 35M training parameters), we have to think if this leads to a inconvenient heavy model. Line 16 of Table 2.2 shows that we can have similar good results with considerably less parameters to learn (8M) by fine-tuning the rest of the parameters. This seems to be also the conclusion in Leclerc et. al [23] paper, where the U-Net 1 with minimum training parameters has same or better performance than the U-Net 2 with higher number of parameters.

The second issue is the importance of the Batch Normalization (BN). It seems that in most cases leads to a better performance. This is also described in Xiao Yun Zhou et. al paper [39], where after an extended experimentation, the conclusion is that BN improves the accuracy of training U-Net for biomedical semantic

segmentation. Batch Normalization also reduces the need of Dropout Layers (See also: [17]) and that's why we didn't perform any Dropout layer in our model implementation.

The third issue has to do with the penalties we applied in miss-classification combined with the Dice Loss Function. It's clear that seems to work and is important to implement also in case of Categorical Cross Entropy loss function to see the deference.

The forth issue regards the capability to include a different Optimizer as training parameter in our parametric model. It seems that there is a discussion in academic community that in some cases, SGD optimizer with momentum can give a big improvement over Adam in segmentation problems [37].

2.7 Post-processing

Prediction for validation set already discussed in previous sections. It produces 100 square segmented images and all metrics of table 2.2 is based on this prediction. There is no need for post processing. Also, prediction on test set also produces 100 square segmented images with the same dimension as the feeding on the network, for example 256×256. But in case of prediction on test set, we have to go back in the initial dimension of the images in order to have the segmentation map in correct dimension. This is because, evaluation and clinical metrics on test set (see Chapter 3) are calculated by CAMUS online evaluation platform [23] and presupposes the initial dimension and format of the images.

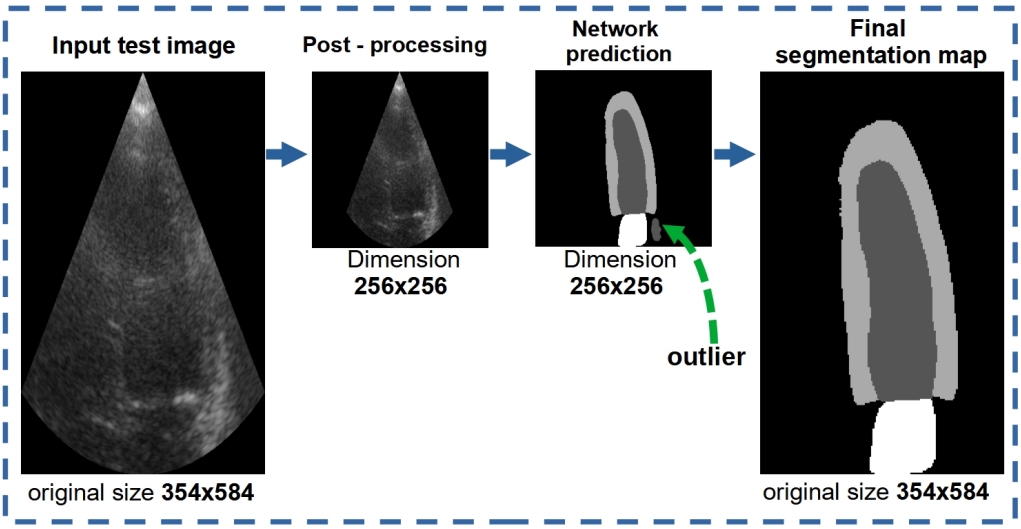


Figure 2.13: Post - processing procedure (the last two stages). Erase any small connecting components (prediction outlier) and resize to original image size, removing the extra zero padding areas added during pre - processing.

Consequently the first post-processing step is to reconstruct the product of segmentation map (a square 4 label image) to the initial dimension of the original test image. This is done by reversing the procedure of "making square and padding" we saw in preprocessing section (2.1), applied to the predicted label images and using the programming tools MedPy [25] and Pillow [7]. MedPy help us to read the initial configuration file that includes the original image dimension, and Pillow tools are used for cropping the extra zero padding and resizing to the original dimension. NumPy [14], the fundamental package for scientific computing with Python also used.

We have also add a check if prediction produces small connecting components that are performed as outliers in our segmentation process (Figure 2.13). In this case, we carefully erase them from the final prediction map, using the morphology tools of SkImage Python library [35]. We saw that these outliers where more possible to produced in case we didn't use the penalties we discussed in Loss function section and only in very poor image quality.

Chapter 3

Results and Discussion

We have already discussed the evaluation metrics, after training, in validation set (Table 2.2), where we have available the ground truth images and also the correct LV_{edv}, LV_{esv} and LV_{ef}. This availability gave us the opportunity to focus on which parameters give the best metrics of the LV_{ef} and the MAE.

Choosing the five best calculated weight files, the next step is to make the predictions, with the same ensemble method, to test set given by CAMUS evaluation challenge. Online evaluation platform in CAMUS testing phase, provides 50 patients, also two and four chamber for both ED and ES. This time there are not available the ground truth images and the correct metrics for LV_{edv}, LV_{esv} and LV_{ef}.

We have to participate to CAMUS challenge to get access to the involved test dataset and also to submit our segmentation results through a dedicated online evaluation platform by uploading the corresponding files, where:

- For each patient, we have to upload four files: two corresponding to the two-chamber (2CH) view acquisition for both End Diastole (ED) and End Systolic (ES) and two corresponding to the four-chamber (4CH) view acquisition for both End Diastole (ED) and End Systolic (ES).
- Each file has to be named according to a special naming convention
- All results should be saved into a raw/mhd image format. Each segmented image should involve discrete values using the convention: 0 for background, 1 for left ventricle, 2 for myocardium and 3 for left atrium
- Each segmented image should be expressed in the corresponding input US image space.
- In order to optimize the computation of the different error measures, we are invited to respect the same properties as the ones of the reference segmented images whose information are provided by a special protocol for LV Endo, LV Epi and LA segmentation.

After the test prediction and uploading the correct files, CAMUS evaluation platform provides a full functional mechanism to measure both geometrical and clinical metrics.

3.1 Geometrical Metrics

It's important to measure the degree of accuracy of the left ventricular endocardium and epicardium as the left atrium. This will be done through global and local measures of similarity with the reference contours. A set of geometrical metrics are computed per structure:

- For left ventricle endocardium (LV endo), (a) the average Dice value for the left ventricle cavity at ED and at ES, (b) the average mean absolute distance for the endocardial contour of the left ventricle at ED and at ES, (c) the average Hausdorff distance for the endocardial contour of the left ventricle at ED and at ES.
- For left ventricle endocardium (LV epi), (a) the average Dice value for the area composed by the myocardium and the left ventricle cavity at ED and at ES, (b) the average mean absolute distance for the epicardial contour of the left ventricle at ED and at ES, (c) the average Hausdorff distance for the epicardial contour of the right ventricle at ED and at ES.
- For left atrium (LA), (a) The average Dice value for the left atrium region at ED and at ES, (b) the average mean absolute distance for the left atrium contours at ED and at ES, (c) the average Hausdorff distance for the left atrium contours at ED and at ES.

Table 3.1, shows the above metrics for LV endo and LV epi for the proposed method, measured by CAMUS evaluation platform after our last submission, compared with the three evaluated methods of the current leader board.

Table 3.1: Segmentation Accuracy (LV_{endo} and LV_{epi}) of the 3 evaluated methods and the proposed method on the test set

Methods	LV_{endo}						LV_{epi}					
	ED			ES			ES			ES		
	DSI	MAD	HD	DSI	MAD	HD	DSI	MAD	HD	DSI	MAD	HD
U-Net-1	0.936	1.7	5.3	0.912	1.7	5.5	0.956	1.7	**5.2**	0.946	1.9	5.7
U-Net-2	0.922	1.6	5.7	0.899	1.7	5.3	0.932	2.0	6.4	0.923	2.1	6.4
ACNNs	0.936	1.7	5.6	0.913	1.7	5.6	0.953	1.9	5.9	0.945	2.0	5.9
Proposed	**0.941**	**1.5**	**4.9**	**0.922**	**1.5**	**4.9**	**0.956**	**1.7**	5.3	**0.950**	**1.8**	**5.3**

We have to notice that except from the best Dice metric for LV_{endo} end LV_{epi}, our proposed method has the lower average Hausdorff distance, a crucial metric

in case of segmentation. The Hausdorff distance [16] calculates the maximum distance between the contours of the predicted and the ground truth image. A low Hausdorff distance value represents a good segmentation result. The unit of the Hausdorff distance is millimeter (mm), which is calculated from information in the echo images. We believe that we can improve further more the geometrical metrics after the implementation of the tasks we describe in Future work section (4.2).

3.2 Clinical Metrics

Maybe the most important is the degree of accuracy of the derived clinical indices because the are ones that are the most widely used in cardiac clinical practice, specially for the Left ventricular cavity. For this purpose, CAMUS online platform provides the mechanism to measure:

- The Correlation coefficient computed from the set of End Diastolic Volumes (EDV), the set of End Systolic Volumes (ESV), the set of Ejection Fraction (EF) measurements
- The Bias computed from the set of EDV, the set of ESV, the set of EF measurements
- The Limits of agreement (LOA = 1.96 times the standard deviation) computed from the set of EDV, the set of ESV, the set of EF measurements

Table 3.2, shows the correlation, bias and standard deviation (calculated by LOA) for LV_{edv}, LV_{esv} and LV_{ef} for the proposed method, measured by CAMUS evaluation platform after our last submission, compared with the three evaluated methods of the current leader board. Clinical metrics indices were computed with the Simpson's rule [11] from the segmentation results of each algorithm.

Table 3.2: Clinical Metrics of the 3 evaluated methods and the proposed method on the test set

Methods	LV_{ef}			LV_{ed}			LV_{es}		
	corr	bias	std	corr	bias	std	corr	bias	std
U-Net-1	0.845	**0.1**	**7.3**	0.926	7.2	15.6	0.960	4.4	10.2
U-Net-2	0.792	2.6	8.5	0.963	**-2.4**	**11.1**	0.972	-3.0	**7.6**
ACNNs	0.807	0.3	8.3	0.928	2.8	15.5	0.954	2.0	10.1
Proposed	**0.871**	1.6	8.0	**0.973**	3.0	11.4	**0.974**	**0.4**	9.4

We have to notice that although our proposed method has the best performance in LV_{ed}, LV_{es} and total LV_{ef}, we still have high values for Bias and std. This fact

shows that meybe we have the opportunity to further improve the clinical metrics after the implementation of the tasks we describe in Method issues subsection (2.6.2), specially the LV_{ef}.

In conclusion, it is really encourage that in both evaluation cases (geometrical and clinical metrics) measured by the independent CAMUS platform tools (based on C++ ITK libraries), our proposed method gives the best performance compared to the other three evaluated methods, specially in case of clinical metrics that is most important.

Christoforos Sfakianakis

View leaderboard

Grown Ultrasound Data for Unet (GUDU) in Echocardiography Segmentation (ver. 3)

Submitted **July 13, 2021 at 15:32:39** by **Christoforos Sfakianakis** with approach **default**

View All Structures scores | View left atrium scores | View epicardium scores | View endocardium scores

LEFT VENTRICLE: ENDOCARDIUM

	ED dice	ES dice	ED hausdorff	ES hausdorff	ED mad	ES mad	EF (%)	Volume ED (ml)	Volume ES (ml)
Mean	0.941	0.922	4.9	4.9	1.5	1.5			
Correlation coefficient							0.871	0.973	0.974
BIAS							1.6	3.0	0.4
LOA							[-11.5 ; 19.8]	[-16.0 ; 28.7]	[-14.5 ; 22.5]

LEFT VENTRICLE: EPICARDIUM

	ED dice	ES dice	ED hausdorff	ES hausdorff	ED mad	ES mad
Mean	0.956	0.950	5.3	5.3	1.7	1.8

LEFT ATRIUM

	ED dice	ES dice	ED hausdorff	ES hausdorff	ED mad	ES mad
Mean	0.885	0.915	5.9	5.5	2.4	2.1

LEADERBOARD

User	Mean DICE ED	Mean DICE ES	Mean Hausdorff ED	Mean Hausdorff ES	Mean MAD ED	Mean MAD ES	EF correlation	EF bias	EF standard deviation(std)	Volume ED correlation	Volume ED bias	Volume ED std	Volume ES correlation	Volume ES bias	Volume ES std
Sarah Leclerc															
U-net-1(instance)															
endo	0.936	0.912	5.3	5.5	1.7	1.7	0.845	0.1	7.3	0.926	7.2	15.6	0.960	4.4	10.2
epi	0.958	0.946	5.2	5.7	1.7	1.9									
la	0.889	0.918	5.7	5.3	2.2	2.0									
U-net-2(instance)															
endo	0.922	0.899	5.7	5.3	1.6	1.7	0.792	2.6	8.5	0.963	-2.4	11.1	0.972	-3.0	7.6
epi	0.932	0.923	6.4	6.4	2.0	2.1									
la	0.848	0.888	6.9	6.2	2.6	2.1									
ACNNs(instance)															
endo	0.936	0.913	5.6	5.6	1.7	1.7	0.807	0.3	8.3	0.928	2.8	15.5	0.954	2.0	10.1
epi	0.953	0.945	5.9	5.9	1.9	2.0									
la	0.881	0.911	6.0	5.8	2.3	2.2									

Figure 3.1: Camus online evaluation platform results. Screenshot after login from http://camus.creatis.insa-lyon.fr/challenge/#challenges. Red numbers are our computed results, black numbers are the current leader-board results.

Chapter 4

Conclusion

4.1 Summary

In this work, we presented a novel ensemble of deep learning models for the problem of cardiac ultrasound (echocardiography) image segmentation. The probability predictions by several training instances are combined based on ensemble mean method. Five best performance calculated weights, was used to find the mean probability per area of interest (Background, Left Ventricle, Myocardium and Left Atrium) and produce the corresponding labels. Pearson Correlation coefficient (CORRe) and mean absolute error (MAE) between the real LV ejection fraction and the predicted, was used as a metric criteria.

In order to improve the learning capability and the generality of our U-net based Neural Network, we designed and implement a specialized data augmentation, adapted to the echocadiographic image data. Three novel augmentation techniques implemented, two virtual probe orientations and a virtual low-contrast scanner setting, based on the fact that human hand is involved in echocardiography procedure and is highly subjective the final ultrasound image result.

Our results on the testing dataset of CAMUS online competition shows that the proposed data augmentation method and ensemble mean prediction method achieves an overall improvement compared to several benchmark techniques.

4.2 Future work

We concede that the proposed approach has several advantages, that mainly has to do with the simplicity of the methods pipeline. Future plans include to experiment with the interesting state-of-the-art Generative Adversarial Networks (GANs) introduced by [13] to produce augmented samples which are not manually designed but learned by a neural network. We also plan to test our approach on other datasets to determine, if the adaptation could be applied in a wide range of similar data from different clinical centers and different ultrasound scanners. We think that a good start is to evaluate our calculated weights with ultrasound

images extracted from EchoNet-Dynamic dataset [28], a publicly available dataset of de-identified echocardiogram videos or maybe to produce a mixed new training dataset taking images from both datasets and re-train our proposed model.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Yasser Ali, Farrokh Janabi-Sharifi, and Soosan Beheshti. Echocardiographic image segmentation using deep res-u network. *Biomedical Signal Processing and Control*, 64:102248, 2021.

[3] Alyaa Amer, Xujiong Ye, Massoud Zolgharni, and Faraz Janan. Resdunet: Residual dilated unet for left ventricle segmentation from echocardiographic images. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 2019–2022. IEEE, 2020.

[4] Anil Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109, 1943.

[5] Chen Chen, Chen Qin, Huaqi Qiu, Giacomo Tarroni, Jinming Duan, Wenjia Bai, and Daniel Rueckert. Deep learning for cardiac image segmentation: a review. *Frontiers in Cardiovascular Medicine*, 7:25, 2020.

[6] Francois Chollet. Keras, 2015. `https://keras.io/`. Last accessed 16 February 2022.

[7] Alex Clark. Pillow (pil fork) documentation. *Readthedocs. Https://Buildmedia. Readthedocs. Org/Media/Pdf/Pillow/Latest/Pillow. Pdf*, 2015.

[8] Antonio Criminisi, Ian Reid, and Andrew Zisserman. A plane measuring device. *Image and Vision Computing*, 17(8):625–634, 1999.

[9] Truong Dang, Tien Thanh Nguyen, Carlos Francisco Moreno-García, Eyad Elyan, and John McCall. Weighted ensemble of deep learning models based on comprehensive learning particle swarm optimization for medical image segmentation. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 744–751. IEEE, 2021.

[10] Vilson Soares de Siqueira, Moisés Marcos Borges, Rogério Gomes Furtado, Colandy Nunes Dourado, and Ronaldo Martins da Costa. Artificial intelligence applied to support medical decisions for the automatic analysis of echocardiogram images: A systematic review. *Artificial intelligence in medicine*, 120:102165, 2021.

[11] ED Folland, AF Parisi, PF Moynihan, D Ray Jones, Ch L Feldman, and DE Tow. Assessment of left ventricular ejection fraction and volumes by real-time, two-dimensional echocardiography. a comparison of cineangiographic and radionuclide techniques. *Circulation*, 60(4):760–766, 1979.

[12] Paul Gavrikov. visualkeras. `https://github.com/paulgavrikov/visualkeras`, 2020.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[14] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[15] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[16] Daniel P Huttenlocher, Gregory A. Klanderman, and William J Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, 15(9):850–863, 1993.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Kenya Kusunose, Akihiro Haga, Takashi Abe, and Masataka Sata. Utilization of artificial intelligence in echocardiography. *Circulation Journal*, pages CJ–19, 2019.

[20] Roberto M Lang, Luigi P Badano, Victor Mor-Avi, Jonathan Afilalo, Anderson Armstrong, Laura Ernande, Frank A Flachskampf, Elyse Foster, Steven A Goldstein, Tatiana Kuznetsova, et al. Recommendations for cardiac chamber quantification by echocardiography in adults: an update from the american society of echocardiography and the european association of cardiovascular imaging. *European Heart Journal-Cardiovascular Imaging*, 16(3):233–271, 2015.

[21] Sarah Leclerc, Erik Smistad, Thomas Grenier, Carole Lartizien, Andreas Ostvik, Frédéric Cervenansky, Florian Espinosa, Torvald Espeland, Erik Andreas Rye Berg, Pierre-Marc Jodoin, et al. Ru-net: A refining segmentation network for 2d echocardiography. In *2019 IEEE International Ultrasonics Symposium (IUS)*, pages 1160–1163. IEEE, 2019.

[22] Sarah Leclerc, Erik Smistad, Andreas Østvik, Frédéric Cervenansky, Florian Espinosa, Torvald Espeland, Erik Andreas Rye Berg, Mourad Belhamissi, Sardor Israilov, Thomas Grenier, et al. Lu-net: a multistage attention network to improve the robustness of segmentation of left ventricular structures in 2-d echocardiography. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 67(12):2519–2530, 2020.

[23] Sarah Leclerc, Erik Smistad, Joao Pedrosa, Andreas Østvik, Frederic Cervenansky, Florian Espinosa, Torvald Espeland, Erik Andreas Rye Berg, Pierre-Marc Jodoin, Thomas Grenier, et al. Deep learning for segmentation using an open large-scale dataset in 2d echocardiography. *IEEE transactions on medical imaging*, 38(9):2198–2210, 2019.

[24] Fei Liu, Kun Wang, Dan Liu, Xin Yang, and Jie Tian. Deep pyramid local attention neural network for cardiac structure segmentation in two-dimensional echocardiography. *Medical Image Analysis*, 67:101873, 2021.

[25] Oskar Maier. Medpy, 2013. `https://pypi.python.org/pypi/MedPy`. Last accessed 16 February 2022.

[26] J Alison Noble and Djamal Boukerroui. Ultrasound image segmentation: a survey. *IEEE Transactions on medical imaging*, 25(8):987–1010, 2006.

[27] Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitsas, Mattias Heinrich, Wenjia Bai, Jose Caballero, Stuart A Cook, Antonio De Marvao, Timothy Dawes, Declan P O'Regan, et al. Anatomically constrained neural networks (acnns): application to cardiac image enhancement and segmentation. *IEEE transactions on medical imaging*, 37(2):384–395, 2017.

[28] David Ouyang, Bryan He, Amirata Ghorbani, Neal Yuan, Joseph Ebinger, Curtis P Langlotz, Paul A Heidenreich, Robert A Harrington, David H Liang, Euan A Ashley, et al. Video-based ai for beat-to-beat assessment of cardiac function. *Nature*, 580(7802):252–256, 2020.

[29] Nathan Painchaud, Nicolas Duchateau, Olivier Bernard, and Pierre-Marc Jodoin. Echocardiography segmentation with enforced temporal consistency. *arXiv preprint arXiv:2112.02102*, 2021.

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[31] John Greenlees Semple and Geoffrey Thomas Kneebone. *Algebraic projective geometry*. Clarendon Press, 1979.

[32] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

[33] Georgios Simantiris and Georgios Tziritas. Cardiac mri segmentation with a dilated cnn incorporating domain-specific constraints. *IEEE Journal of Selected Topics in Signal Processing*, 14(6):1235–1243, 2020.

[34] Erik Smistad, Ivar Mjåland Salte, Andreas Østvik, Sarah Leclerc, Olivier Bernard, and Lasse Lovstakken. Segmentation of apical long axis, four-and two-chamber views using deep neural networks. In *2019 IEEE International Ultrasonics Symposium (IUS)*, pages 8–11. IEEE, 2019.

[35] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.

[36] Ke Wang, Bin Fang, Jiye Qian, Su Yang, Xin Zhou, and Jie Zhou. Perspective transformation data augmentation for object detection. *IEEE Access*, 8:4935–4943, 2019.

[37] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. *arXiv preprint arXiv:1705.08292*, 2017.

[38] Ziv Yaniv, Bradley C Lowekamp, Hans J Johnson, and Richard Beare. Simpleitk image-analysis notebooks: a collaborative environment for education and reproducible research. *Journal of digital imaging*, 31(3):290–303, 2018.

[39] Xiao-Yun Zhou and Guang-Zhong Yang. Normalization in training u-net for 2-d biomedical semantic segmentation. *IEEE Robotics and Automation Letters*, 4(2):1792–1799, 2019.

[40] Vasily Zyuzin, Andrey Mukhtarov, Denis Neustroev, and Tatiana Chumarnaya. Segmentation of 2d echocardiography images using residual blocks in u-net architectures. In *2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*, pages 499–502. IEEE, 2020.