Deep Aesthetic Assessment of Breast Cancer Surgery Outcomes

Tiago Gonçalves(✉), Wilson Silva, and Jaime Cardoso

Faculty of Engineering, University of Porto, Porto, Portugal
{up201607753,wilson,jsc}@fe.up.pt

Abstract. Breast cancer is a highly mutable and rapidly evolving disease, with a large worldwide incidence. Even though, it is estimated that approximately 90% of the cases are treatable and curable if detected on early staging and given the best treatment. Nowadays, with the existence of breast cancer routine screening habits, better clinical treatment plans and proper management of the disease, it is possible to treat most cancers with conservative approaches, also known as breast cancer conservative treatments (BCCT). With such a treatment methodology, it is possible to focus on the aesthetic results of the surgery and the patient's Quality of Life, which may influence BCCT outcomes. In the past, this assessment would be done through subjective methods, where a panel of experts would be needed to perform the assessment; however, with the development of computer vision techniques, objective methods, such as BAT© and BCCT.core, which perform the assessment based on asymmetry measurements, have been used. On the other hand, they still require information given by the user and none of them has been considered the gold standard for this task. Recently, with the advent of deep learning techniques, algorithms capable of improving the performance of traditional methods on the detection of breast fiducial points (required for asymmetry measurements) have been proposed and showed promising results. There is still, however, a large margin for investigation on how to integrate such algorithms in a complete application, capable of performing an end-to-end classification of the BCCT outcomes. Taking this into account, this thesis shows a comparative study between deep convolutional networks for image segmentation and two different quality-driven keypoint detection architectures for the detection of the breast contour. One that uses a deep learning model that has learned to predict the quality (given by the mean squared error) of an array of keypoints, and, based on this quality, applies the backpropagation algorithm, with gradient descent, to improve them; another which uses a deep learning model which was trained with the quality as a regularization method and that used iterative refinement, in each training step, to improve the quality of the keypoints that were fed into the network. Although none of the methods surpasses the current state of the art, they present promising results for the creation of alternative methodologies to address other regression problems in which the learning of the quality metric may be easier. Following the current trend in the field of web development and with the objective of transferring BCCT.core to an online format, a prototype of a web application for the automatic keypoint detection was

J. Henriques et al. (Eds.): MEDICON 2019, IFMBE Proceedings 76, pp. 1967–1983, 2020.
https://doi.org/10.1007/978-3-030-31635-8_236

developed and is presented in this document. Currently, the user may upload an image and automatically detect and/or manipulate its keypoints. This prototype is completely scalable and can be upgraded with new functionalities according to the user's needs.

1 Introduction

1.1 Context

Breast cancer is considered a systemic disease from diagnosis, so, it is extremely important to determine cancer staging (how widespread cancer is at the time of diagnosis), a factor which will influence the treatment decision from that moment [1,2]. Surgery is usually the primary approach in terms of treatment and its main objectives are to remove cancer and to determine the stage. It is possible to specify surgery in the following types:

- **Partial (segmental) mastectomy or lumpectomy**: which is based on the removal of the cancerous tissue with a rim of healthy tissue (free margin). This approach is also known as Breast Cancer Conservative Surgery (BCS) or Breast Cancer Conservative Treatment (BCCT) [1–3].
- **Simple or total mastectomy**: which includes the removal of the entire breast, but not of the lymph nodes under the arm or muscle tissue from beneath the breast [1,2,4].
- **Modified radical mastectomy**: based on the removal of the entire breast without the chest muscle and removal of the first two stages of lymph nodes under the arm [1,2].
- **Radical mastectomy**: which consists of the extensive removal of the entire breast, lymph nodes, and chest wall muscles under the breast [1,2,5].

Patients usually undergo radiation or systemic therapies in order to increase the effectiveness of the treatments. It is important to take into account that the duration of the therapy and the type of radiation that will be used will always depend on the type of performed surgery [6].

1.2 Motivation

Despite being a highly mutable and rapidly evolving disease, it is estimated that most breast cancers (approximately 90% of the cases) are treatable and curable if detected on early staging and given the best treatment [7]. Nowadays, with implemented breast cancer routine screening habits [8,9], it is possible to do proper management of the disease, in case it appears, and to develop a better clinical plan to treat it. This means that most breast cancers can now be treated using conservative approaches, i.e., BCCT, contributing, thus, for local control of the disease, with similar survival rates to those obtained with mastectomy, although, with better cosmetic results [4,7,10]. Moreover, the definition of an objective evaluation standard for the cosmetic outcome of BCCT has become crucial, due to the fact that with the development of new oncoplastic techniques it becomes necessary to have an objective method to compare cosmetic results [7,11]. There is a need for an automatic method that is able to perform this assessment by simply receiving data (i.e., images) as input. To accomplish this, one must develop an algorithm that is able to automatically detect breast fiducial points (i.e., keypoints), which are needed to compute asymmetry measurements and to perform the assessment.

2 Literature Review

2.1 BCCT Assessment

Although BCCT has an identical meaning worldwide, it is not a standardized procedure [12], due to the fact that there are some technical variations related both to surgery and radiotherapy. Generally, the aesthetic result of BCCT is done by an observer who identifies and evaluates colour, shape, geometry, irregularity and roughness of the treated breast in comparison with the untreated breast. This relies on the assumption that better results correspond to more similar breasts; this is considered to be a much more practical approach and a method that fits the evolution and the appearance of new emerging oncoplastic techniques, where, usually, both breasts are submitted to surgery, leading, consequently, to a more demanding comparison [7,13]. Historically, it was Jay Harris, in 1979, the one who had introduced a subjective overall cosmetic score, that would become a standard in conservation breast procedures: the Harvard scale [14]. This scale classifies the overall cosmetic result in four classes:

(1) **Excellent**: if the treated breast is nearly identical to the untreated one.
(2) **Good**: if the treated breast has some differences when compared with the untreated one.
(3) **Fair**: if the treated breast is clearly different from the untreated one, but not seriously distorted.
(4) **Poor**: if the treated breast is seriously distorted.

2.2 Computer-Aided Aesthetic Classification of BCCT Outcomes

Later, software-based methods came up with the main idea of predicting the global aesthetic result since they are based on different individual characteristics which are automatically and objectively extracted from patient photographs. These approaches explore the ability that computational methods have to provide an effective, easy, fast, reliable and reproducible tool to evaluate the consequences of breast cancer patient care [7]. Regarding this topic one must talk about Breast Analysing Tool - BAT© from Fitzal *et al.* [15] and BCCT.core from Cardoso and Cardoso [16]. On behalf of BAT©, Fitzal *et al.* proposed the Breast Symmetry Index (BSI) to evaluate the cosmetic outcome of BCCT. With BAT©, they were able to measure differences between left and right breast sizes from a patient's digital picture. The sum of all differences will output the BSI score, which has the capability of measuring size differences in between breasts. Concerning BCCT.core, it is important to state that it is a computer-aided medical system that detects breast keypoints and performs automatic feature extraction from patient photographs, in order to capture some of the relevant factors for the overall cosmetic result. To do a proper classification, it is necessary to obtain a concise representation of a BCCT image, based on asymmetry, colour differences and scar visibility features. To describe breast asymmetry, several features are taken into consideration [16] and, in order to extract them, it is necessary to mark some specific keypoints in the image, using BCCT.core software: sternal notch, the scale mark, the nipples and left and right breasts contours, adjusted with an active contour based on splines with control points.

2.3 A Deep Learning Approach to Keypoint Detection

Recently, Silva *et al.* have proposed the use of a deep neural network (DNN) for the keypoints detection task. With deep learning, it is possible to follow an integrated learning approach that uses the context information [17]. Actually, regarding keypoint detection with deep learning, there are two relevant works that presented interesting ideas: Cao *al.* [18], which brought the idea of learning *part confidence maps* and *part confidence fields* as a mean to detect keypoints in the end; Belagiannis *et al.* [19] which also proposed an architecture that first learns how to regress *heatmaps* (i.e., each keypoint is modelled with a Gaussian distribution) and, after iterative tuning on the training of heatmap regression, it is able to predict keypoint localizations. Taking these ideas into account, Silva *et al.* have proposed the generation of an intermediate representation (i.e., *heatmaps*), which consists on a fuzzy localization for the keypoints that are intended to be detected. The heatmaps are obtained using the segmentation model, U-Net [20]. For the keypoint regression, the strategy starts with the multiplication of the image with the refined output of the previous module, as a way of improving the fuzzy localization of keypoints, which will then enhance the exact keypoint detection. Therefore, a regression module predicts the keypoints' coordinates.

3 Image Segmentation for Breast Contour Detection

3.1 Background

Typically, in biomedical applications, the employment of a proper segmentation method is of utmost importance, being one of the most relevant steps in the pipeline [21,22]. With the advent of novel artificial intelligence techniques, it is possible to divide image segmentation into traditional approaches, such as **thresholding** [23,24], **edge based** [25,26], **region based** [27–30], **deformable models** [30–32] or **graph cuts** [33,34], or deep learning approaches which make use of the capabilities of DNNs to succeed in these tasks. Regarding the scope of this thesis, it is relevant to mention four architectures:

- **Fully Convolutional Network** (FCN): developed by Long *et al.* and trained end-to-end and/or pixel-to-pixel on semantic segmentation [35]. All of the following architectures are based on this one.
- **U-Net**: proposed by Ronneberger *et al.* and applied in biomedical image segmentation [20].
- **Global Convolution Network** (GCN): proposed by Peng *et al.* and applied in localization and classification tasks in image semantic segmentation [36].
- **DeepLabv3+**: published by Chen *et al.* in 2018, is, in fact, an extension of DeepLabv3 [37], with the addition of an encoder-decoder module to refine segmentation results. This work is based on the use of *atrous convolution*, or dilated convolution, and is the state of the art for image segmentation [38].

3.2 Segmentation for Breast Contour Detection

The main hypothesis behind this approach is that it should be easier to detect breast contours if one is able to properly detect the breast first. This is a challenge of semantic segmentation, where one needs to properly separate both breasts (i.e., *foreground*) from the other image components (i.e., *background*). In this case, the segmentation task would work as means to an end; ideally, within this approach, the pipeline would be composed of two main processes: breast segmentation and breast contour detection, where the second is fully-dependent on the first. In this experience, the main goal was to study the capabilities of state of the art segmentation DNNs on the specific task of segmenting both breasts from the images of the dataset. It is important to take into account that one can only move to the subtask of breast contour detection if good performance is achieved in the segmentation task.

3.3 Implementation and Results

For this study, it was decided to use state of the art architectures, such as: U-Net, GCN and DeepLabv3+. The dataset (221 images) was divided into train (107 images), validation (47 images) and test (67 images) sets. All images were first resized to the dimensions of 512×384. It was used the Dice Coefficient [39] (DC) as performance metric, which is very common in image segmentation tasks; DC values near 1 mean that there is high similarity between both ground-truth and predicted masks. It can be written as

$$DC = \frac{2\sum_{i}^{N} p_i g_i}{\sum_{i}^{N} p_i^2 + \sum_{i}^{N} g_i^2} \tag{1}$$

where N represents the total number of pixels, p_i is the predicted pixel label, g_i is the ground-truth pixel label and $i \in \{0, 1\}$, meaning that this is a binary-class problem. Regarding loss function, the *Dice Coefficient Loss* (DCL) was chosen and monitored during training. It can be written as

$$DCL = 1 - DC \tag{2}$$

where DC is the Dice Coefficient. U-Net was implemented in Keras [40] and GCN was implemented in PyTorch [41]. Both were trained during 300 epochs, with Adadelta [42] as optimization function. During training phase, data augmentation techniques such as rotation, horizontal and vertical shifting, shear mapping, zoom in and zoom out or horizontal and vertical flips were employed. The model with the lower loss value on the validation set was chosen during training and saved. It was then used to perform inference and evaluation on the test set. Results on test set are shown in Table 1; each DC value on the table is actually the average DC value obtained with each model in test set. Due to the lack of computational resources it was not possible to train the TensorFlow implementation of DeepLabv3+.

Table 1. Average Dice Coefficient results on test set for U-Net and GCN models. Best result is highlighted in bold.

	Average Dice Coefficient
U-Net	0.8689
GCN	**0.8937**

3.4 Discussion and Conclusions

Although it was possible to obtain DC results near 1, for this specific task, they do not contribute as much as it was initially expected. Here, the main goal was to generate breast binary masks from a given image in order to detect breast contours, being a two-class segmentation problem, where breasts are foreground and the rest of the image is the background. Taking this into account, it is possible to explain high results on the DC score: masks are mostly composed by background pixels (imbalanced-class problem), so it is easier for the network to predict background instead of foreground due to the fact that it has naturally seen more background pixels, during training. On the other hand, it can be said that, during inference, the trained models are able to predict the most probable breast localization. The main issue is that this is still a fuzzy localization, which means that it will not help in finding breast contours. Also, when images contain small or undefined breasts, it becomes even harder to obtain reliable breast image masks, which would lead to increased difficulty in finding breast contours. For these reasons, post-processing experiences to detect contours after segmentation were not performed. On the other hand, since breast contour detection fully depends on excellent results on image segmentation, this approach, at the moment, is not a viable option.

4 Quality-Driven Keypoint Detection

4.1 State of the Art on Deep Quality-Driven Architectures

Generally, deep learning architectures for image analysis take an image as input and return a mask (image segmentation) or a value (image classification) as output; these deep models are then regularized in a supervised manner on the object of interest. On the other hand, the need for great amounts of training data or the necessity of re-training/fine-tuning such models in order to apply them in different contexts may be problematic [43]. To overcome these issues, two recent works from Fernandes *et al.* and from Rebelo *et al.* are presented and explained.

Deep Image Segmentation by Quality Inference. In their original work, Fernandes *et al.* started by defining the quality metric for the learning phase; they ended up choosing DC, which is widely used in this type of tasks as a similarity metric. Considering that modelling of utility functions is usually inspired in pairwise preferences [44] and/or cardinal/ordinal functions [45], the authors proposed a model that is capable of learning the quality (i.e., DC), given an image and mask pair; in this case, the utility function is precisely this measure of correspondence between a mask and an image. To achieve such a model, the authors proposed a deep architecture, *Gossip Network*, which receives an image and mask pair as input and has two streams that try to model the foreground and background, respectively by the input mask. Moreover, as a strategy to increase/decrease the network's confidence in the recognition of their corresponding regions, these streams communicate (i.e., to gossip) between each other. After training, the *Gossip Network* is used to predict the quality of a given image and mask pair and, based on this prediction, the mask is iteratively refined by backpropagation until it reaches a stop criteria (e.g., quality value, number of iterations).

Quality-Based Regularization for Iterative Deep Image Segmentation. Rebelo *et al.* have presented a new methodology, which uses the notion of quality as a regularization method during the training of a network for direct segmentation refinement [46]. They proposed a network with an encoder-decoder structure which takes an image and a mask as inputs, and returns quality and a refined segmentation as output, following a multi-task learning paradigm. Although this work follows a similar strategy to the one presented by Fernandes *et al.* [43] (e.g., the quality metric is the same), the main focus in this case was the study of a DNN which could receive the predicted mask from a state of the art network and iteratively improve the segmentation, while predicting the quality of the input segmentation, at the same time; this parallel quality prediction can be understood as a regularization method that will favour the learning of features connected to the degree of correction necessary to improve the input segmentations. To start the training, it is required to feed the network with an image and an initial segmentation mask (obtained from any other model); this initial mask will be refined according to an iterative method, where the network will use its own output as a new input for further improvement, leading, thus, to higher predicted quality values. In their original work, Rebelo *et al.* use U-Net [20], trained on the same dataset, to obtain the initial segmentation masks, which are then concatenated to the input image as an additional channel.

4.2 Deep Keypoint Detection by Quality Inference

The main idea behind this first approach is the application of the main concepts from [43] to keypoint detection, instead of image segmentation. To achieve such a model, one must first define a suitable quality metric. Following the work developed in [17], mean squared error (MSE) was chosen as the quality metric. It is written as:

$$MSE = \frac{1}{n}\sum_{i=0}^{n-1}(k_i - \hat{k}_i)^2 \tag{3}$$

where n is the number of samples (i.e., number of coordinates in each keypoints array), i is the index of the value, $\hat{k}$ is the predicted keypoints array and k is the ground-truth keypoints array. The main objective is to have a model that receives an image and an array with initial keypoints candidates as inputs and that outputs a quality value (i.e., MSE between the candidate keypoints array and the ground-truth keypoints array). The image is given as an input to a CNN, a latent representation is obtained and then concatenated with the candidate keypoints array. This concatenation is then fed to a Multilayer Perceptron (MLP) which will output the quality value predicted, related to the image and the candidate keypoints. The CNN Module is composed by VGG16 [47] as backbone without its dense layers, followed by four convolutional layers and three dense layers; the latent representation of the image is obtained from the last dense layer. This latent representation is then concatenated with the candidate keypoints array which is then fed to a MLP; this MLP has three dense layers. The quality value is obtained from the last dense layer. After training the model on the quality inference task, the next step would be the use of the backpropagation algorithm to improve the predicted quality, leading, thus, to coordinate values near ground-truth. In order to do this, one must first to compute the gradients of the predicted quality, $\hat{q}$, with respect to the given candidate keypoints, k', and use gradient descent (important to remind that the quality metric is given by an error metric, so, what is aimed is the minimization of this value) with a learning rate, α, as follows:

$$k' \leftarrow k' - \alpha\frac{\partial \hat{q}}{\partial k'} \tag{4}$$

After several steps, it would be expected that the real MSE between the changed candidate keypoints and the ground-truth keypoints would be lower than the real MSE between the initial candidate keypoints and the ground-truth keypoints.

4.3 Deep Iterative Refinement of Keypoint Detection

This second approach aims to apply the methodology proposed in [46] to the task of keypoint detection, through the iterative refinement of the given candidate keypoints. Similarly to the work developed by Rebelo *et al.*, the main objective here is to use an iterative approach to improve the quality (in this case, minimize its value, since it is a MSE value) of the given candidate keypoints; the quality value acts here as a form of regularization, during training. When compared to the architecture presented in Subsection 4.2, this one has some modifications. The CNN Module keeps the same structure, however, three new MLPs are added to the model in order to fulfill the main objective. The Intermediate MLP receives as input the result of the concatenation between the candidate keypoints and the Image Embedding and generates what is called an Intermediate Embedding, which has lower dimensions than the first one. This latent representation is then given to the Keypoints MLP, which has the task of predicting the keypoints coordinates, and to the Quality MLP, which has to learn how to predict the quality value, related to the image and candidate keypoints pair.

4.4 Implementation and Results

Deep Keypoint Detection by Quality Inference. This model was implemented in Keras [40]. All images were first resized to the dimensions of 512 × 384; keypoints were also resized accordingly and were normalized by the width of the image, in order to be between 0 and 1. The dataset was divided into train, validation and test sets, as previously explained. To ensure that the model was robust, data augmentation techniques were employed during training, in an online fashion, to both images and ground-truth keypoints: translations, rotations and flips were used. Since the main objective is to teach the model on quality inference, it is important to ensure that a large range of quality values is shown to the network, during training; in order to achieve that, one must be certain that to the initial candidate keypoints were applied the correct diversity of transformations, so that, the correspondent qualities cover a wide range of values. To do this, the data augmentation strategy presented by Fernandes *et al.* in [43] was adapted and used; in this case, the transformations that were applied to the keypoints were translations, rotations and horizontal flips. Once again, all candidate keypoints and their correspondent qualities are generated in an online setting to keep dynamic training. Regarding the training phase, the loss function was defined as follows:

$$\mathcal{L}_{model} = \lambda_1 \mathcal{L}_{imageembedding} + \lambda_2 \mathcal{L}_{quality} \tag{5}$$

where λ_1 and λ_2 represent non-negative weights attributed to each loss. The loss of the image embedding is computed as the MSE between predicted image embedding and the ground-truth keypoints. The intuition behind this supervised approach is to guide the CNN Module to learn a latent representation that is near the ground-truth keypoints in order to facilitate the learning of the quality by the MLP. Regarding quality prediction, the loss function is also computed as the MSE between the predicted quality value and the ground-truth quality for the candidate keypoints. The model was trained during 300 epochs, with Adadelta [42] as optimization function, and the model with lower loss on the validation set was saved. After training, several initial candidate keypoints for test set were generated, following three possible situations: candidate keypoints were the result of the prediction obtained with the model described in [17] in images of the test set; candidate keypoints were the result of random generated values; candidate keypoints were coordinates with value 0. The quality was then inferred from the initial candidate keypoints, using the trained model, and then the backpropagation algorithm proposed was applied. Regarding backpropagation, the presented results use a learning rate $\alpha = 0.001$, selected taking into account the order of magnitude of the errors. The number of backpropagation iterations was 1000 and 5000. Regarding initial candidate keypoints and adapted candidate keypoints, the average MSE, its standard deviation and the maximum and minimum errors were obtained. Tables 2, 3 and 4 show results for the three possible initial candidate keypoints status: candidate keypoints were the result of the prediction obtained with the model described in [17] in images of the test set; candidate keypoints were the result of random generated values; candidate keypoints were coordinates with value 0. **Note:** In the following Tables, GT stands for ground-truth keypoints, IC for initial candidate keypoints, AC for adapted candidate keypoints and STD for standard deviation.

Table 2. Results for the backpropagation applied to the candidate keypoints which were the result of the prediction obtained with the model described in [17] in images of the test set. Best results are highlighted in bold.

Real MSE	Mean	STD	Max	Min	Steps
GT & IC	**0.00116**	**0.00150**	0.00659	**0.00013**	**-**
GT & AC	0.00118	0.00151	**0.00651**	0.00014	**1000**
GT & AC	0.00139	0.00158	0.00656	0.00017	**5000**

Table 3. Results for the backpropagation applied to the candidate keypoints which were the result of random generated values. Best results are highlighted in bold.

Real MSE	Mean	STD	Max	Min	Steps
GT & IC	0.11286	0.01700	0.15130	0.07088	**-**
GT & AC	0.11087	0.01639	0.14704	0.07051	**1000**
GT & AC	**0.10519**	**0.01492**	**0.14063**	**0.06954**	**5000**

Table 4. Results for the backpropagation applied to the candidate keypoints which were coordinates with value 0. Best results are highlighted in bold.

Real MSE	Mean	STD	Max	Min	Steps
GT & IC	0.27012	0.03186	0.34395	0.21046	-
GT & AC	0.25837	0.03128	0.33107	0.20054	**1000**
GT & AC	**0.21929**	**0.02865**	**0.28671**	**0.16724**	**5000**

Deep Iterative Refinement of Keypoint Detection. This model was implemented in PyTorch [41]. The data preparation process was the same as previously explained. Regarding training phase, the model was trained during 300 epochs, with Adadelta [42] as optimization function, and the model with lower loss on the validation set was saved. For this approach, a different strategy for the loss function was designed:

$$\mathcal{L}_{model} = \lambda_1 \mathcal{L}_{quality} + \lambda_2 \mathcal{L}_{refinedkeypoints} \tag{6}$$

where λ_1 and λ_2 represent non-negative weights attributed to each loss. The loss of the quality (regularization term) is computed the same way as presented above. The loss of the refined keypoints (main goal) is calculated as the MSE between the predicted refined keypoints and the ground-truth keypoints. Tables 5, 6 and 7 show results for the three possible initial candidate keypoints status: candidate keypoints were the result of the prediction obtained with the model described in [17] in images of the test set; candidate keypoints were the result of random generated values; candidate keypoints were coordinates with value 0.

Table 5. Results for the iterative refinement model applied to the candidate keypoints which were the result of the prediction obtained with the model described in [17] in images of the test set.

Real MSE	Mean	STD	Max	Min
GT & IC	**0.00116**	**0.00150**	**0.00659**	**0.00013**
GT & AC	0.00218	0.00164	0.00750	0.00028

Table 6. Results for the iterative refinement model applied to the candidate keypoints which were the result of random generated values.

Real MSE	Mean	STD	Max	Min
GT & IC	0.11286	0.01700	0.15130	0.07088
GT & AC	**0.00233**	**0.00150**	**0.00633**	**0.00033**

Table 7. Results for the iterative refinement model applied to the candidate keypoints which were coordinates with value 0.

Real MSE	Mean	STD	Max	Min
GT & IC	0.27012	0.03186	0.34395	0.21046
GT & AC	**0.00345**	**0.00294**	**0.01293**	**0.00045**

4.5 Discussion and Conclusions

Both proposed models were functional, however, they were not able to improve the state of the art results, i.e., the ones published by Silva *et al.* in [17]. Regarding the first approach, explained in Subsection 4.1, there are some considerations that have to be taken into account: inference of quality has always an associated error amplitude, which means, that the inference will always have some deviations from the real quality value; this will have implications when performing backpropagation in the candidate keypoints, due to the fact, that, in each step, the inferred quality value will always diminish, but that does not necessarily mean that the real quality value has diminished. Also, although results are presented for different values of backpropagation steps, an optimal value still needs to be found; this is crucial, since, following the main conclusions from Fernandes *et al.* in [43], too much backpropagation steps may lead to a deterioration of the adapted candidate keypoints. On behalf of the second approach, the refined keypoints predictions have better performances, but still need further improvements, regarding the fact that they did not improve the results obtained with the model proposed by Silva *et al.* in [17]. It is important to notice that, even with data augmentation, the dataset used to train the models may not be large enough to address these specific tasks; the applied data augmentation parameters may also not be enough, and, it is possible that the models need an even higher variety of examples to properly learn. It is also important to mention that this was the first work which used a quality-driven approach for a regression task; this means that there is space for improvements. Also, it is still not clear that this approach will lead to success, considering this kind of tasks. For all these reasons, these models still need further research and development in order to be considered viable options to solve this thesis' main problem.

5 A Web Application for Automatic Keypoint Detection

5.1 Motivation

Currently, in order to use BCCT.core [16], users must download, install and set-up the software on their computers; these three steps may be considered a burden for some of the users and they also imply that the software will use their computational resources to properly function. On the other hand, fundamental services such as software maintenance (updates and/or upgrades) or support could be difficult in an offline setting. To overcome these issues, and taking into account that most of the healthcare institutions have an Internet connection [48], it was decided that the best option would be to move BCCT.core into a web-based application. This way, users would only need to deal with a simple website registration in order to use all the functionalities of the software; the rest would be handled in the background, on the server where the application is hosted. Moreover, the field of web development is in a trending evolution, being able to keep up with most of the market's new requirements [49]. Use-cases and the proposed architecture for such an application are further described.

5.2 Use-Cases and Proposed Architecture

The starting point for the establishment of use-cases for this web-based application for automatic keypoint detection was BCCT.core [16]; the main functionalities will not require major improvements, however, the transition from offline to online setup has a big impact on the new practicalities of the application, and this should be taken into account on the redesign of the software. The identified use-cases, from the user side, were:

(1) The user should log in to enter the application: with the application in an online setting, it is crucial that measures that take into account the privacy of the clinical data are implemented, and, as such, it is believed that a login system for user authentication inside the application is of utmost importance.
(2) The user may upload medical photos: to perform the keypoint detection, the application should have an intuitive interface so users can upload the photos that are going to be subjected to evaluation.
(3) The user may consult and change the detection records: it is important that the user has access to the evaluated photos and to the predicted keypoints and he/she should be able to manually correct the model's predictions.
(4) The user may erase all the records from the system: once again, since one is dealing with sensitive clinical data, it is important that the users have the possibility to reset their records.

From the server side, the system should have a database. This database will contain:

(1) User's information: name, e-mail and password.
(2) Uploaded Photo's information: location, predicted keypoints and the associated user.

Regarding the type of relationship, it can be seen that it is a one-to-many relationship, i.e., an user can upload many photos. Another fundamental requirement is a back-end capable of interacting with deep learning models created in Python; the server should have enough space to host such models.

5.3 Implementation and Prototype

The web application prototype was developed with Django, a free and open source high-level Python web framework that promotes fast development, security and scalability. The main reason behind this choice is indeed related with the fact that the deep learning models are written in Python, and since Django is also powered by the same programming language, this application-models interaction would be simpler. Moreover, Django permits the creation and manipulation of databases through Python, simplifying, once again, all the development that requires interaction between the front-end and the back-end. With everything on the back-end being handled by Django, one has to deal with the front-end. The front-end of an application is directly related to its usability and user experience. In terms of development, it was used HTML and CSS, which are the standard code languages for web development. One of the main objectives is to have an interface that allows users to perform their tasks (i.e., keypoint detection and/or aesthetic assessment) as simple as possible; to do that, it was decided to incorporate all the functionalities as buttons, where the user clicks and sees the results, without the need to check what is happening in the background. An administration interface was also developed, since, in reality, such a system requires a professional who can properly manage, update and maintain it, while giving some kind of support to the users.

5.4 Discussion and Conclusions

It was possible to achieve a fully working prototype, capable of fulfilling all the previously stated use-cases and representing, thus, a reliable proof of concept that can act as a starting point for the final deployment of a final version of the web application for automatic keypoint detection. Regarding the fact that the system is in development mode, one of the next steps would be, for example, changing the database framework (currently the system uses SQLite) to a scalable one (for example, PostgreSQL), which is supported by Django; important to remind, that the method of interacting with the database stays almost untouched. Once again, Django handles all the background work.

6 Future Work

Regarding future work recommendations, it is proposed to:

- Use DeepLabV3+ [38] for breast image segmentation;
- Explore recurrent neural networks [50] (RNN) or graph neural networks [51] (GNN) for keypoint prediction;
- Test different metrics for quality evaluation;
- Acquire more data to improve training performance;
- Deploy the Web application for healthcare professionals.

References

1. Ely, S., Vioral, A.N.: Breast cancer overview. Plast. Surg. Nurs. **27**, 128–133 (2007)
2. Street, W.: Breast Cancer Facts & Figures 2017–2018, p. 44 (2017)
3. Grisotti, A.: Immediate Reconstruction After Partial Mastectomy, p. 12 (1994)
4. Fisher, B., Anderson, S., Bryant, J., Margolese, R.G., Deutsch, M., Fisher, E.R., Jeong, J.-H., Wolmark, N.: Twenty-year follow-up of a randomized trial comparing total mastectomy, lumpectomy, and lumpectomy plus irradiation for the treatment of invasive breast cancer. New Engl. J. Med. **347**, 1233–1241 (2002)
5. Fisher, B., Montague, E., Redmond, C., Barton, B., Borland, D., Fisher, E.R., Deutsch, M., Schwarz, G., Margolese, R., Donegan, W., Volk, H., Konvolinka, C., Gardner, B., Cohn, I., Lesnick, G., Cruz, A.B., Lawrence, W., Nealon, T., Butcher, H., Lawton, R., Investigators, O.N.: Comparison of radical mastectomy with alternative treatments for primary breast cancer: a first report of results from a prospective randomized clinical trial. Cancer **39**, 2827–2839 (1977)
6. E. B. C. T. C. G. (EBCTCG): Effect of radiotherapy after breast-conserving surgery on 10-year recurrence and 15-year breast cancer death: meta-analysis of individual patient data for 10 801 women in 17 randomised trials. The Lancet **378**, 1707–1716 (2011)
7. Oliveira, H.P., Cardoso, J.S., Magalhaes, A., Cardoso, M.J.: Methods for the aesthetic evaluation of breast cancer conservation treatment: a technological review. Curr. Med. Imaging Rev. **9**, 32–46 (2013)
8. Grady, K.E., Lemkau, J.P., McVay, J.M., Reisine, S.T.: The importance of physician encouragement in breast cancer screening of older women. Prev. Med. **21**, 766–780 (1992)
9. Smith, R.A., Haynes, S.: Barriers to screening for breast cancer, p. 11 (1992)
10. Veronesi, U., Cascinelli, N., Mariani, L., Greco, M., Saccozzi, R., Luini, A., Aguilar, M., Marubini, E.: Twenty-year follow-up of a randomized study comparing breast-conserving surgery with radical mastectomy for early breast cancer. New Engl. J. Med. **347**, 1227–1232 (2002)
11. Cardoso, M.J., Cardoso, J.S., Vrieling, C., Macmillan, D., Rainsbury, D., Heil, J., Hau, E., Keshtgar, M.: Recommendations for the aesthetic evaluation of breast cancer conservative treatment. Breast Cancer Res. Treat. **135**, 629–637 (2012)
12. Christiaens, M., van der Schueren, E., Vantongelen, K.: More detailed documentation of operative procedures in breast conserving treatment: what good will it do us? Eur. J. Surg. Oncol. (EJSO) **22**, 326–330 (1996)
13. Asgeirsson, K., Rasheed, T., McCulley, S., Macmillan, R.: Oncological and cosmetic outcomes of oncoplastic breast conserving surgery. Eur. J. Surg. Oncol. (EJSO) **31**, 817–823 (2005)
14. Harris, J.R., Levene, M.B., Svensson, G., Hellman, S.: Analysis of cosmetic results following primary radiation therapy for stages I and II carcinoma of the breast. Int. J. Radiat. Oncol. Biol. Phys. **5**, 257–261 (1979)
15. Fitzal, F., Krois, W., Trischler, H., Wutzel, L., Riedl, O., Kühbelböck, U., Wintersteiner, B., Cardoso, M., Dubsky, P., Gnant, M., Jakesz, R., Wild, T.: The use of a breast symmetry index for objective evaluation of breast cosmesis. The Breast **16**, 429–435 (2007)
16. Cardoso, J.S., Cardoso, M.J.: Towards an intelligent medical system for the aesthetic evaluation of breast cancer conservative treatment. Artif. Intell. Med. **40**, 115–126 (2007)

17. Silva, W., Castro, E., Cardoso, M.J., Fitzal, F., Cardoso, J.S.: Deep keypoint detection for the aesthetic evaluation of breast cancer surgery outcomes. In: Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI 2019) (2019)
18. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, arXiv:1611.08050 [cs], November 2016
19. Belagiannis, V., Zisserman, A.: Recurrent Human Pose Estimation, arXiv:1605.02914 [cs], May 2016
20. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv:1505.04597 [cs], May 2015
21. Fasihi, M.S., Mikhael, W.B.: Overview of current biomedical image segmentation methods. In: 2016 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, pp. 803–808. IEEE, December 2016
22. Sharma, N., Ray, A., Shukla, K., Sharma, S., Pradhan, S., Srivastva, A., Aggarwal, L.: Automated medical image segmentation techniques. J. Med. Phys. **35**(1), 3 (2010)
23. Saha, P., Udupa, J.: Optimum image thresholding via class uncertainty and region homogeneity. IEEE Trans. Pattern Anal. Mach. Intell. **23**, 689–706 (2001)
24. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**, 62–66 (1979)
25. Kekre, D.H.B., Gharge, S.M.: Image Segmentation using Extended Edge Operator for Mammographic Images, vol. 02, no. 04, p. 6 (2010)
26. Zanaty, E.A.: Improved region growing method for magnetic resonance images (MRIs) segmentation. Am. J. Remote Sens. **1**(2), 53 (2013)
27. Shan, J., Cheng, H., Wang, Y.: A completely automatic segmentation method for breast ultrasound images using region growing. In: Proceedings of the 11th Joint Conference on Information Sciences (JCIS). The Harbin Institue of Technology, Shenzhen, China. Atlantis Press (2008)
28. Tamilselvi, P.R., Thangaraj, D.P.: Segmentation of Calculi from Ultrasound Kidney Images by Region Indicator with Contour Segmentation Method, p. 10 (2011)
29. Day, E., Betler, J., Parda, D., Reitz, B., Kirichenko, A., Mohammadi, S., Miften, M.: A region growing method for tumor volume segmentation on PET images for rectal and anal cancer patients: a region growing method for tumor segmentation. Med. Phys. **36**, 4349–4358 (2009)
30. Davis, J.B., Reiner, B., Huser, M., Burger, C., Székely, G., Ciernik, I.F.: Assessment of 18f PET signals for automatic target volume definition in radiotherapy treatment planning. Radiother. Oncol. **80**, 43–50 (2006)
31. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. Int. J. Comput. Vis. **1**, 321–331 (1988)
32. Xu, C., Prince, J.: Snakes, shapes, and gradient vector flow. IEEE Trans. Image Process. **7**, 359–369 (1998)
33. Thongnuch, V., Uyyanonvara, B.: Automatic Optic Disk Detection From Low Contrast Retinal Images of ROP Infant Using GVF Snake, p. 13 (2007)
34. Boykov, Y., Funka-Lea, G.: Graph cuts and efficient N-D image segmentation. Int. J. Comput. Vis. **70**, 109–131 (2006)
35. Shelhamer, E., Long, J., Darrell, T.: Fully Convolutional Networks for Semantic Segmentation, arXiv:1605.06211 [cs], May 2016
36. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large Kernel Matters – Improve Semantic Segmentation by Global Convolutional Network, arXiv:1703.02719 [cs], March 2017

37. Chen, L.-C., Papandreou, G., Schroff, F., Adam, H.: Rethinking Atrous Convolution for Semantic Image Segmentation, arXiv:1706.05587 [cs], June 2017
38. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation, arXiv:1802.02611 [cs], February 2018
39. Milletari, F., Navab, N., Ahmadi, S.-A.: V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation, arXiv:1606.04797 [cs], June 2016
40. Chollet, F., et al.: Keras (2015)
41. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch (2017)
42. Zeiler, M.D.: ADADELTA: An Adaptive Learning Rate Method, arXiv:1212.5701 [cs], December 2012
43. Fernandes, K., Cruz, R., Cardoso, J.S.: Deep image segmentation by quality inference. In: 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, pp. 1–8. IEEE, July 2018
44. Fernandes, K., Cardoso, J.S., Palacios, H.: Learning and ensembling lexicographic preference trees with multiple kernels. In: 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, pp. 2140–2147. IEEE, July 2016
45. Ellsberg, D.: Classic and current notions of “measurable utility”. Econ. J. **64**, 528 (1954)
46. Rebelo, J., Fernandes, K., Cardoso, J.S.: Quality-based Regularization for Iterative Deep Image Segmentation, p. 4 (2019)
47. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv:1409.1556 [cs], September 2014
48. Séror, A.C.: Internet infrastructures and health care systems: a qualitative comparative analysis on networks and markets in the British national health service and kaiser permanente. J. Med, Internet Res. **4**, e21 (2002)
49. Ricca, F., Tonella, P.: Analysis and testing of web applications. In: Proceedings of the 23rd International Conference on Software Engineering, ICSE 2001, Toronto, Ont., Canada, pp. 25–34. IEEE Computer Society (2001)
50. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W.: CNN-RNN: a unified framework for multi-label image classification. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 2285–2294. IEEE, June 2016
51. Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., Pascanu, R.: Relational inductive biases, deep learning, and graph networks, arXiv:1806.01261 [cs, stat], June 2018