Master degree Research internship in Computer Science

Semi supervised deep learning for medical image segmentation

Author :
Abdelhadi TEMMAR

Supervisor :
Prof. Desrosiers Christian
Co-supervisor : Dr. Dolz Jose
Prof. Ben Ayed Ismael

Host laboratory :
Livia (Laboratory of Vision and Artificial Intelligence), ETS, Montreal, Canada

Secretariat - tel : 01 69 15 81 58 - mail : alexandre.verrecchia@u-psud.fr

Table des matières

Abstract

This report described the work I've done during my internship at Livia at Montreal. Semantic segmentation for medical images is very important for clinicians in order to extract multiple 3D information about the object of interest. Convolutional Neural Network(CNN) became the most efficient way to do that. But using Deep Neural Network need a huge amount of labeled data with variability among them, which is very costly, especially in medical images where the segmentation has to be done by an expert. That's why we propose a new method minimizing an energy function which use as unary term, the output of a CNN and a binary term for smoothing the prediction, in order to do the training in a semi-supervised setting, which means that only a few images with ground truth segmentation done by experts are used during the training and we use data where the segmentation was not provided to add knowledge to the network. We tested it in two different datasets and obtained satisfactory results (increase up to 20%) and reduce considerably the gab with the fully supervised setting.

Key words

Deep Learning, Energy function, Graphcut, Semi-Supervised, CNN.

Résumé

Ce rapport décrit le travail que j'ai accompli pendant mon passage à Livia à Montréal. La segmentation sémantique pour les images médicales est très importante pour les cliniciens afin d'extraire de multiples informations 3D sur l'objet d'intérêt. Le réseau neuronal convolutif (CNN) est devenu le moyen le plus efficace de le faire. Mais l'utilisation de Deep Neural Network nécessite une grande quantité de données avec une variabilité parmi celles-ci, ce qui est très coûteux, en particulier dans les images médicales où la segmentation doit être effectuée par un expert. C'est pourquoi nous proposons une nouvelle méthode qui minimise une fonction énergétique utilisant comme "unary term" la sortie d'un CNN et un "binairy term" pour lisser la prédiction, afin de faire l'apprentissage dans un cadre semi-supervisé, ce qui signifie quelques images annotées par des experts sont utilisées pendant l'apprentissage et nous utilisons des données où la segmentation n'a pas été fournie pour ajouter des connaissances au réseau. Nous l'avons testé sur deux ensembles de données différents et obtenu des résultats satisfaisants (augmentation jusqu'à 20%) et réduit concidérablement l'écart avec les résultats obtenues après un entrainement entièrement supervisé.

Mot clés

Apprentissage profond, Fonction d'énergie, Graphcut, Semi-supervisé, CNN.

CHAPITRE 1

Presentation of the laboratory

I did my internship at the Livia laboratory. According to [2], the unifying theme of the LIVIA research team lies in the field of visual perception of 2D and 3D scenes, as well as elements of artificial intelligence (the use of a priori knowledge, context, intelligent control and inspection, interpretation, and other machine vision concepts). The laboratory's scientific orientation around the key foundations of « image processing, analysis, and interpretation » is illustrated by modeling, methodology and application studies. These considerations account for the laboratory's structure and its evolution. Within this context, the following six priority axes have undergone sustained development over the years : i) Machine vision, ii) automatic image processing of documents, iii) imaging (medical, satellite images, etc.), iv) biometrics, surveillance, and intrusion detection, v) learning in static and dynamic environments and vi) perception and environment. Multiple PhDs and Postdocs coming from all over the world are working in this lab on a wide variety of subjects. Among them, a team of students is working particularly on medical imaging. I was one of them.

CHAPITRE 2

Introduction

2.1 Problem

Image segmentation consists on labeling each object of interest in an image. That means assigning the same class of each pixels representing the same object of interest. With increasing use of Computed topography (CT) and Magnetic resonance (MRI) [34], it is essential to use computers and algorithms to assist radiological experts and doctors in clinical diagnosis for the sake of accurate diagnosis, treatments, and follow-up. Indeed segmentation is very useful for extracting multiple information about the region of interest. However, doing this work manually, by segmenting every slice, is time-consuming for clinicians, for whom every second is precious and very hard to reproduce due to inter observer variability.

2.2 Relative works

Therefore, a multitude of methods have been proposed for segmenting medical images in the most accurate way possible [26] using a variety of techniques such as edge detection, deformable models [24], atlas models [23] [4] and statistical models [15]. Although those methods gave satisfactory results in many applications, they have their limits. Atlas-based models work by aligning one or several anatomical templates to the target image and then transferring segmentation labels from the templates to the image. Therefore this method may not be able to capture the full variability of the object of interest and segmenting one image can take several hours. On the other hand, statistical models use a training data to learn a parametric model describing the structure to the segment. Even if this method can give good results when training with a large dataset, it is very sensitive to noise and overfits often the training data when this one is small compared to the number of parameters to learn.

Most of the successful semantic segmentation systems developed in the previous decade relied on hand-crafted features combined with flat classifiers, such as Random Forests [35], or most commonly Support Vector Machines [37]. Although those methods outperformed well in some applications [8], they were limited by the way features were extracted (i.e : defined by the user, hand-crafted) and are now outmatch by deep learning methods.

Deep learning algorithms have been applied to a wide range of different problems (computer vision, natural language processing, speech recognition) and have, most of the times, surpasses the previous state of the art algorithms. Therefore, it became very common to use Deep learning for image segmentation. Deep learning refers to neural networks with multiple layers that are able to learn different levels of abstraction. In this work, we will only focus on Convolutional Neural Networks (CNNS), the most popular neural network for dealing with image data. CNN, like every machine learning algorithm, are designed to solve an optimization problem which will be described in section 3.1. In 2015, Ronenberger [32] proposed a new architecture, called U-net, applied specifically to medical images. It was designed for fast and precise segmentation of images. The architecture will be described in section 3.5. Also, [9], [17], [10] propose to use a 3D CNN for medical-image-segmentation. Indeed, most of the time, for medical images, we have volume in space or time. Using this information can be very useful for the segmentation and demonstrated to outperformed the 2D versions. However, it has a cost, 3D CNN uses more GPU memory and takes longer to train.

With enough data, Deep learning can give very good results for image-segmentation. Through, as said before, it is very costly to obtain the ground truth segmentation for each image. That's why

some recent research papers were concentrated by finding a way to train CNN in a Weakly/Semi Supervised setting. Weakly supervised means training with only weak labels such as bounding box. On the other hand, a semi supervised setting means using labeled data (ground truth segmentation provided) and unlabelled data. In [36], the authors proposed an iterative algorithm, combining CNN and Grabcut [33] for training in a weekly supervised way (bounding box as week labels). The proposed DeepCut method is described by the authors as an iterative energy minimization similar to GrabCut [33]. Contrary to GrabCut, DeepCut does not use a GMM to parametrize the color distributions of the foreground and background but it replaces the GMM with a Neural Network model and the graph cut solver with a densely connected graph. In addition, the method proposed does not recompute completely the parameters θ at each iteration. Instead, the CNN is reinitialized with the parameters of the last iterations. Pathak et al in [28] proposed a method to optimize a CNN with only a set of linear constraints as weak labels. Linear constraints can be many things such as lower bound and upper bound on the number of pixels predicted background and foreground. The method is original but it is hard to find linear constraints that fit and represent a complete dataset. Furthermore, after testing the code, the algorithm does not seem to always respect the constraints. Papandreou et al [27] proposed a weekly and semi supervised learning developed on top of DeepLab algorithm [7] based on an Expectation-Maximization algorithm to iteratively update the targets which achieved similar results that fully supervised methods in the Pascal Dataset (real images). Graphical energy minimization methods are usually the keys for solving those kind of problems due to inherent optimality guarantees and computational efficiency. These techniques have been already employed for fully automated image segmentation [39], [38],[19], [30]. Furthermore, it has become common to use graphical energy minimization such as CRF[20] for post processing the segmentation output of a CNN.

2.3 Contribution

Similarly to DeepCut [36] we propose an iterative algorithm based on CNN and energy minimization(i.e graphcuts [6]) which will be described in 3. Unlike Deep-Cut, we do not pre-segment the images with grabcut [33], but with a CNN trained on a few samples of the dataset. Indeed, in medical images, with only bounding box, it can be very hard for GrabCut to distinguish which area we wish to segment. Indeed, MRI images are grayscale images while grabcut was firstly designed for RGB images, also, the contrast between the object of interest and the other objects can be very low. On the other hand, CNN learns from an image with ground truth segmentation and can most of the case localize precisely the center of the object we want to segment. Also, we experimented different foreground weights loss and background weight loss for the labels predicted by the CNN which will be described in 3 and results of those experiments presented in section 4.

The purpose of our work is to show that in medical images, unlabeled data are not useless and can be used with Deep Learning methods to improve the results of the segmentation. We demonstrate the robustness of this algorithm in two different datasets, one representing the spine and the other one the right ventricle.

CHAPITRE 3

Method

In this section, we present first a brief description of Convolution Neural Network (CNN). Secondly, we describe the energy function we want to minimize and which algorithm we use for optimizing this function.

3.1 Convolutional Neural Network

Convolutional Neural Networks are very similar to a regular neural network such as Multi Layer Perceptron. One of the differences between CNN and multi layer perceptron is that CNN expects an image as input or any multidimensional data which has a spatial relationship. In a regular neural network, a neuron is connected to every neuron in the previous layer, which is hardly manageable for large images such as the ones in medical field [1] (for example, an image of dimension 200 x 200 x 3 = 120 000 parameters to learn for a single neuron). In a CNN architecture, a neuron is not connected to every input neurons. As we can see in the figure 3.1, each pixel is connected to only a local area of the input image. That is because CNN takes into account particularities of images and extracts features in an elegant way using convolution operations.

In a CNN, we can have different types of layers such has convolutional layers, pooling layers or fully-Connected layers. Convolutional layers are certainly the heart of a CNN architecture. The main goal of those layers is to extract features from the input image(such as edges). Convolution layer is composed of a set of filters (with a small width and height) which preserve the spatial relationship between the pixels. Therefore, a CNN decreases drastically the number of parameters to learn. Convolution operations are applied between the input images (output of the previous layer) and filters of the next layer to detect multiple features. It is usually followed by a nonlinear activation function such as RELU [25] in order to introduce nonlinearity into the network. Pooling layers are mainly used to reduce the dimension of the input image and at the same time increase the receptive field (number of input pixels seen by one neuron). Indeed, by using pooling layer, we reduce the meaningless information and then the filter will be able to see more useful information. The training process is summarized in the figure 3.2. By passing through each layer, it can be computed an output prediction. Then we can compute the error by using a loss function such as the cross entropy loss defined in equation 3.1. The next step is to optimize the weights of the CNN to reduce this error by using the stochastic gradient descent algorithm and back-propagation algorithm[21]. Those steps are repeated until convergence of the loss in a validation set.

$$\mathcal{L}(x, y) = -\frac{1}{n}\sum_{i=1}^{n} y^{(i)} \ln a(x^{(i)}) + \left(1 - y^{(i)}\right) \ln\left(1 - a(x^{(i)})\right) \tag{3.1}$$

with $X = \left\{x^{(1)}, \ldots, x^{(n)}\right\}$ is the set of training set and $\mathrm{Y} = \left\{y^{(1)}, \ldots, y^{(n)}\right\}$ the corresponding labels (equals to 0 or 1) and n the number of images in the batch size. a(x) represent the last layer output of a CNN which we apply an activation function, softmax is the most popular, 3.2 which make sure that the probabilities values sum to one.

$$P(y_i = j|z^i) = \frac{e^{z_j^i}}{\sum_k e^{z_k^i}} \tag{3.2}$$

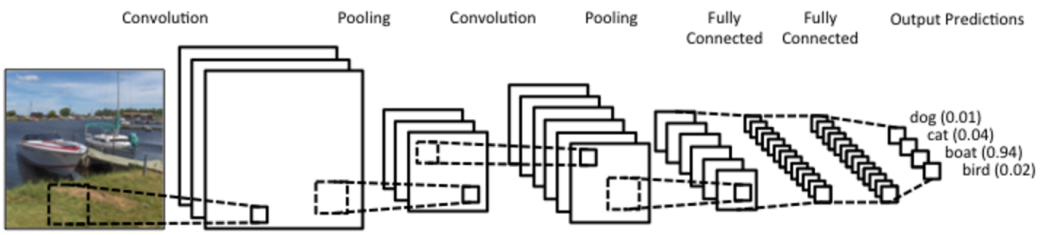

FIGURE 3.1 – Local connectivity in CNN (LeCun architecture [21])

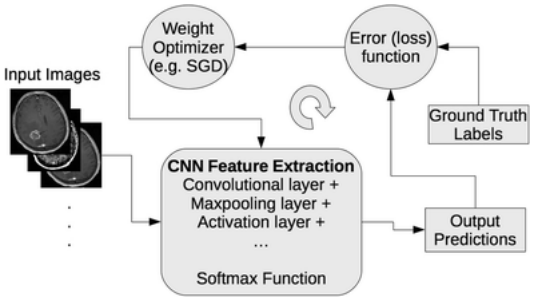

FIGURE 3.2 – A schematic representation of a convolutional neural network (CNN) training process (from [3])

$P(y_i = j|z^i)$ represents the probability that the image I is classified as j knowing the neurons values of the last fully connected layer $z^{\hat{1}}$. This description is specific for image classification tasks but only a few modifications have to be made in order to apply CNN for segmentation problems. One of the first paper talking about this topic is the one from Long.et.al [22]. The authors explain how to train a Convolutional Neural Network(CNN) end-to-end/pixel-to-pixel for semantic segmentation and therefore identify precisely each object of interest. The difference with a regular CNN is that instead of classifying images, we classify each pixel of each image. So the output of a CNN must be of the same size as the input image. The authors explained how to transform any regular CNN in a fully convolutional network by converting all the fully connected layers into convolutional layers using 1x1 filters. We describe the fully convolutional architecture in the section 3.5

3.2 Energy function to minimize

In the following sections, we consider S_L the set of fully labeled data, S_U, the set of unlabelled data used for the training and S_T the testing set. P_x is the set of pixels of the image x. θ the set of parameters of the CNN. V_i is the neighborhood of the pixel i.The method proposed consist of minimizing an energy function similar from the one used in GrabCut [33] and [31]

$$E(y_u, \theta) = \sum_{x \in S_L} \sum_{i \in P} \psi(x_i|\theta) + \sum_{u \in S_U} \sum_{i \in P_x} \psi(u_i|\theta) + \lambda * \sum_{i \in P_u, j \in V_{u_i}} \psi(u_i, u_j) \tag{3.3}$$

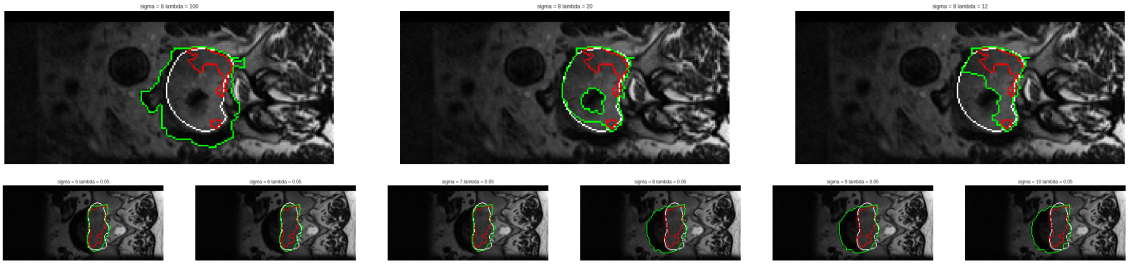

FIGURE 3.3 – Impact of the parameters λ and σ on the spine dataset. First Row : Variation of the parameter λ. Second Row : Variation of the parameter σ.

The first two terms represent the data terms or "unary potential" and the last term represents the smoothness term or "binary potential". Unlike grabcut [33] we do not use a mixture of gaussian for the unary terms but the following equations :

$$\psi(x_i) = y_{x_i} log(P(x_i|\theta)) - (1 - y_{x_i}) log(1 - P(x_i|\theta)) \tag{3.4}$$

$$\psi(u_i) = y_{u_i} log(P(u_i|\theta)) - (1 - y_{u_i}) log(1 - P(u_i|\theta)) \tag{3.5}$$

with $P(x_i|\theta)$ represents the probability map output of the CNN for the foreground label. Therefore, we penalize the pixels predicted background and promote the pixels predicted foreground.

For the binary term, formalized in equation 3.6, we want to attribute more weight to the pixels which have similar intensities than their neighbors (in the original image) (for 3D graph-cut we use 8*2 neighbors) :

$$\psi(u_i, u_j) = exp^{-\frac{(u_i - u_j)^2}{\sigma^2}} \tag{3.6}$$

The sigma parameter controls how much we want to penalize pixels with different intensities. lower this parameter is, less smooth the contours will be. The second parameter λ controls the impact of the binary term. Those two parameters are fixed and defined depending on the dataset.

With the right parameters, using the max-flow algorithm to solve the graphcut problem (more precision will be given in the next section) can give very accurate segmentation. The figure 3.3 show the impact of those parameters. We can see that smaller λ is, more the graph-cut segmentation will be different from the CNN result and consequently segment another object than the one of interest.

3.3 Optimization algorithm

The optimization process of the equation 3.3 can be seen as the algorithm 1
Step 0 : Train the parameters θ of the CNN with the set S_L until convergence.
for *number of iterations* **do**
  Step 1 : Use θ, to obtain the probability map foreground/background $P(u_i|\theta)$ for each image $u \in S_u$
  Step 2 : Use $P(u_i|\theta)$ with graphcut to segment $u \in S_u$
  Step 3 : Use this segmentation as ground truth for the unlabeled data and use them to train the CNN with the labeled data
  Step 4 : Continue to train the CNN with K iteration in order to update θ. During the training, multiply the cross entropy loss function by α_{fg} for the pixels predicted foreground and α_{bg} for the pixels predicted background for the unlabeled data
end

Algorithm 1: Algorithm minimizing the energy function 3.3

It is hard to minimize the function 3.3 by optimizing θ and Y_u at the same time, because the function is not convex. Therefore, we optimize θ and Y_u separately. The problem can be described this way. We have a restricted number of 3D volumes labeled and we want to use unlabeled data to improve the segmentation. Therefore, the energy function 3.3 is minimized by optimizing Y_u and θ. First, we train with the restricted number of patients until it converged. We use the parameters θ learned by the CNN to have $P(u_i|\theta)$ for each pixel "i" of the set of images S_L and S_U. Then we keep θ fixed and minimize 3.3 by applying the max-flow algorithm (described in section 3.4) in 3D (neighborhood 3*8) to have the new labels Y_u. Those labels are used as ground truth for each image u of S_U. The next step is to keep Y_U fixed and optimise θ by training the CNN. We do not train until convergence because we do not want to over-fit those uncertain predictions, therefore we define K, the number of iteration done by the back-propagation algorithm [21]. At each iteration, the CNN only see one image (batch size = 1). α_{fg} and α_{bg} are here for the same reason. Most of the time the prediction is incomplete, that's why two different weights are used, α_{fg} for the pixels predicted foreground at step 2 and α_{bg} for the pixels predicted background. The idea is to give lower weight for these last ones. The modified loss function of the network is described in equation 3.7 We'll see the impact of these parameters in the section 4. This step optimise θ in order to have a more accurate probability distribution $P(y_i|\theta)$. We repeat this process iteratively as described in the algorithm 1.

$$\mathcal{L}(x,y) = -\frac{1}{n}\sum_{i=1}^{n} \alpha_{fg} * y^{(i)} \ln a(x^{(i)}) + \alpha_{bg}\left(1-y^{(i)}\right)\ln\left(1-a(x^{(i)})\right) \tag{3.7}$$

3.4 Max-flow / Min cut problem

As said earlier, the step 2 consist of solving a max-flow problem [5] in order to find the new set of labels Y_u. To solve this problem, it must be represented as a graph problem. It can be defined mathematically in this way : the graph G=(V, E) with V a set of nodes and E a set of edges between the node. If G is a weighted graph, then each edge $(u, v) \in E$ is associated with a weight w(u,v). If G is a directed graph, that means that E is a set of directed edges. For segmentation problems, the graph is an s-t graph which is a weighted directed graph with two identified node, the source S, which represents the foreground, and the sink T, which represents the background (an example can be seen in figure 3.4. Each node $v \in V\backslash\{S,T\}$ represents the pixels of an image. There is an edge between each pixel and the source, the same for the sink. The weights in each edge are the values of the unary potential described by the equation 3.5. Also, the weights between

each pixel (nodes) are the values of the binary potential described in the equation 3.6. An s-t cut c(s,t) is a set of edges E_{cut} such that there is no path from the sink to the source when we remove all the edges of E_{cut}. The cost of E_{cut} is the sum of the edges weights in E_{cut}. Finding E_{cut} which has the minimum cost is equivalent to finding the maximum flow of the graph as stated in the max-flow min-cut theorem [11]. There are different algorithms to solve this problem as described by Y.Boykov et al [5].

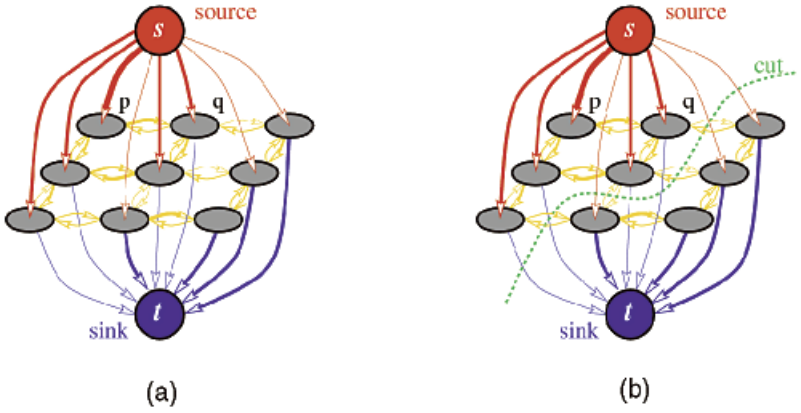

FIGURE 3.4 – Example of s-t graph from [5]

3.5 Experiments details

Spine Dataset

The dataset consists of MRI spine images of 20 different patients with one 3D volume by patients. Each volume is represented by approximately 250 slices (it depends on the subject). The figure 3.6 represent a 3D volume of the spine we want to segment. An example of a 2D slice can be seen at the bottom left of the figure. The main challenge of this dataset is the very low contrast between the bones and the tissues.

We cropped the images to remove useless information and accelerate the training process. The size of each slice became 96 for the width and 208 for the height. 5 volumes have been chosen to be part of the set S_L, 10 for S_U and 5 for the test set S_T

Right Ventricle Dataset

According to [29], the dataset was collected in 2008 at Rouen Hospital. Cardiac MR examinations were performed at 1.5T (Symphony Tim, Siemens Medical Systems, Erlangen, Germany). A dedicated eight-element phased-array cardiac coil was used. Retrospectively synchronized balanced steady-state free precession sequences were performed for cine analysis, with repeated breath-holds of 10–15 s. Since the subject could not hold the breath at exactly the same position each time, there may be a shift in the slices. This inter-slice shift was not corrected. All conventional planes (2-, 3- and 4-chamber views) were acquired and a total of 10–14 contiguous cine short axis slices were performed from the base to the apex of the ventricles. Sequence parameters were as follows : TR = 50 ms ; TE = 1.7 ms ; flip angle = 55 ; slice thickness = 7 mm ; matrix size = 256 216 ; Field of view (FOV) = 360 mm 420 mm ; 20 images per cardiac cycle.

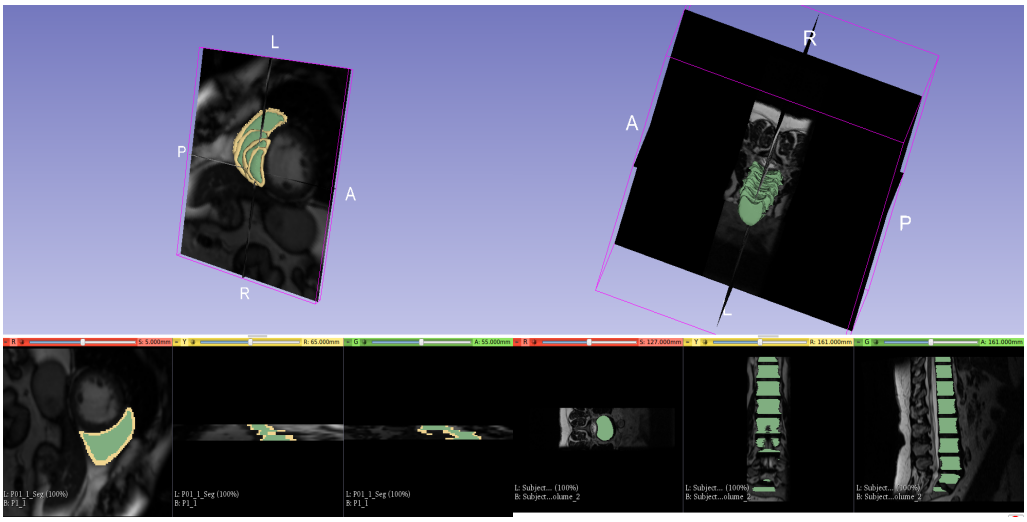

FIGURE 3.5 – Visualization of the 3D volume ground truth segmentation for one patient of the right ventricle dataset

FIGURE 3.6 – Visualization of the 3D volume ground truth segmentation for one patient of the spine dataset

This dataset is also very challenging because of the variable shape of the Right Ventricle. Also in each volume, there is a low number of slices (between 8 and 11) which makes the 3D graphcut less efficient. A 3D representation of the object is shown in the figure 3.5.

Here again, we chose the volumes of 5 different patients to be part of S_L, 6 for S_U and 5 for S_T

CNN architecture

For each experiment, we used U-net architecture, from [32]. It has been designed for fast and precise segmentation of images. The issue of most CNN architecture is that they are not designed for segmentation but classification. There is usually a lot of pooling layer which is good to increase the receptive field and then for image-classification, but most of the case we loose information and context with pooling layer, which is not appropriate for segmentation problems. The idea of U-net is not to remove pooling layers, which are important, as said earlier, to increase the receptive field and decrease the resolutions of the input image, but in order to not loose any features, the features of the same size are concatenated during the up-sampling steps, as we can see on the figure 3.7.This architecture is well known for medical image segmentation and has won multiple challenges. The one we used for our experiments follow this architecture but with fewer parameters (number of features map at each layer) for memory performance.

Also, the training is done by taking the full image as input and not patches.

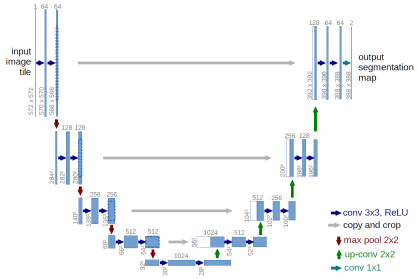

FIGURE 3.7 – U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multichannel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

Evaluation of the segmentation

To evaluate the results, we use Dice coefficient, one of the most used metrics to evaluate image-segmentation. It is the overlap size between the two segmentations divided by the total size of the two objects :

$$DSC(I_{gt}, I_{auto}) = \frac{2|I_{gt} \cap I_{auto}|}{|I_{gt} + I_{auto}|} = \frac{2TP}{2TP + FP + FN} \tag{3.8}$$

With I_{gt} = ground truth segmentation, I_{auto} = segmentation to evaluate, TP = True positive, FP = False Positive, FN = False negative.

This distance of similarity gives a result between 0 and 1 (0 when the two segmentation are disjoint and 1 when they are equals). In the following, every result is computed by using this metric. Also, the scores are computed in 3D. That means that for each volume, the dice of each slice is computed and the average is taken. Then, we compute the mean of all volume dice. It can be translated by the formula 3.9

$$D = \frac{1}{n_v} \sum_{v} \frac{1}{ns_v} \sum_{i=0}^{ns_v} Dice(v_{i_{Pred}}, v_{i_{GT}}) \tag{3.9}$$

With n_v the number of volumes, ns_v the number of slices of the volume v, v_i is the slice i of the volume v, $v_{i_{Pred}}$ is the segmentation prediction of v_i and $v_{i_{GT}}$ is the the ground truth segmentation of the slice v_i.

Data augmentation

During the training at step 0 of the algorithm 1, we applied random rotation and translation at each iteration.

Parameters

The weights of the network have been initialized with Xavier initialization [12] which automatically determines the scale of initialization based on the number of input and output neurons. Instead of using the classical stochastic gradient descent algorithm to optimize the weight of the

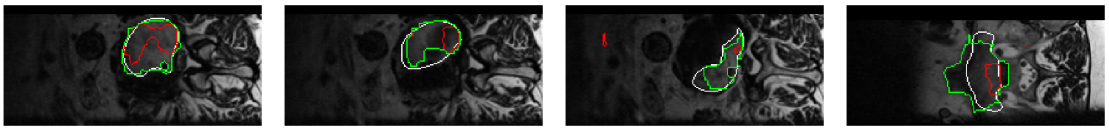

FIGURE 3.8 – Segmentation results at epoch 0 after using 3D graphCut. White : GroundTruth, Red : CNN output, Green : 3D GraphCut improvement

network, Adam optimizer[18] is used. In contrary to stochastic gradient descent, Adam does not use a fixed learning rate during the training but update a per-parameter learning rate depending on the gradient value.

Parameter	Value
Learning rate	1e-8
batch size	1
momentum	0.99
weight decay	0.0005
Graphcut λ	20
Graphcut σ	0.04

Implementations details

For defining, training, testing the Convolutional Neural Network presented in section 3.5, the Caffe framework [16] developed by Berkley AI Research was used. We used medpy library for applying the maxflow algorithm. The iterative algorithm was developed in python and is available at github link. We used the same random seed to choose the training image at each iteration for each experiment in order to compare fairly. The test dataset is never shown during the training process in order to show the robustness of the algorithm to over-fitting. All the experiments have been done in a NVIDIA 960 2 GB of memory.

Max flow / Min cut details experiments

We set a maximum weight for the pixels predicted foreground by the CNN with a probability higher than 0.95 in order to compensate the sensibility of the parameters and prevent the max flow algorithm to make everything disappear. Some results of graph-cut improvement can be seen before the iterative process in the figure 3.8

3.6 Naive learning approach

In order to compare our approach, we compare it to the naive learning approach which consist of setting the labels $Y_u \forall u \in S_u$ to $Y_{u_i} = \arg\max_y y * P(u_i|\theta) + (1-y) * P(u_i|\theta))$. With y=0 for background and y=1 for foreground. That means, the binary potential is removed from the equation 3.3. Also, we compare with the results obtained using the fully supervised training and the ones obtained at step 0 of the algorithm.

CHAPITRE 4

Results

4.1 Impact of the parameters

Firstly, we studied the impact of each parameters (figure 4.5 and figure 4.1) on the right ventricle dataset and spine dataset. As we can see, the weight loss parameters α_{fg} and α_{bg} that we attribute for the images in S_u have a real impact on the results. Two things can be noticed here, first it's important to not attribute high weight for the pixels predicted background by the graphcut optimization. Indeed the best results are obtained with $\alpha_{bg} = 0.1$. If we increase this parameter, the dice decrease significantly. It seems logical because most of the time the prediction is incomplete, which means that there is more chance that some pixel's prediction is false negative than false positive.

4.2 Comparison with the naive learning approach

Secondly, we compared our approach to the naive learning approach described in section 3.6. On the figure 4.7 we can see the comparison between applying graph-cut and without post-processing. We were expecting no improvements because no new knowledge is introduced, but it seems that the thresholding did impact a little. However, the difference is clear with grahcut as post-processing. An other interesting point can be seen by looking closer to the two last plot of this figure. Some volumes have a dice very low at the beginning of the algorithm, but with the binary potential, the improvement is clear after one algorithm iteration, which is not the case with the naive learning approach.

4.3 Comparison with the fully supervised approach

As we see on the figure 4.5 and figure 4.1, the algorithm improve significantly the dice. Despite that improvement, we do not achieve the results of the fully supervised training. However, we get closer, and using more training images labeled could definitively help to achieve the fully supervised results.

4.4 Result for the unlabelled data used for the training

The figure 4.3 show after each algorithm iteration the dice of the CNN output and graphcut output for the spine dataset. An interesting point that we can see is the convergence between the two curve at some point. Which is totally logical, because the CNN is trained for fitting the graphcut output who makes use of the CNN output and the binary potential to improve the segmentation. At some point, the graphcut reach his limitation, that means than the labels $y_u \forall u \in S_u$ stopped updating which makes the parameters θ of the CNN converge too.

4.5 Results spine dataset

For this dataset, there is one subject(patient 25) on the set where the CNN prediction at epoch 0 is very bad compared to the others, as we can see on the figure 4.2. However, after some algorithms iterations, the dice increase of 20 % for this patient. The figure 4.4 show some visual results. The contrast is very low on this subject, that's why training with only 5 subjects was not enough to

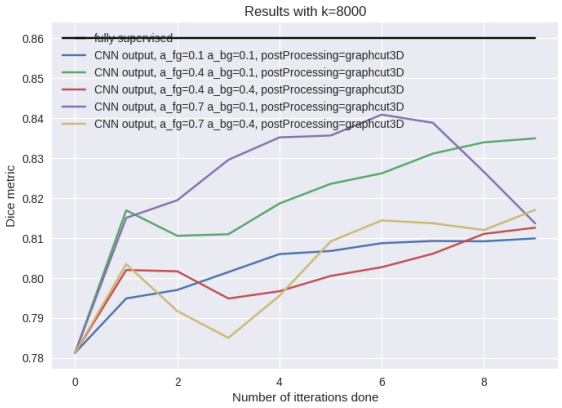

FIGURE 4.1 – Impact of α_{fg} and α_{bg} on the spine dataset

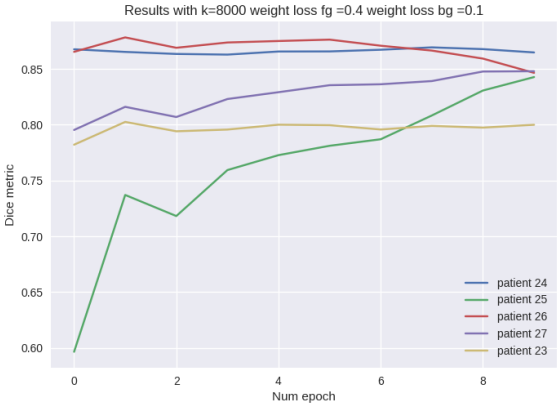

FIGURE 4.2 – Results for each patient volume on test set after each iterations (spine dataset)

adapt for this contrast but by introducing knowledge with unlabelled data, the CNN learn to localize precisely the vertebra.

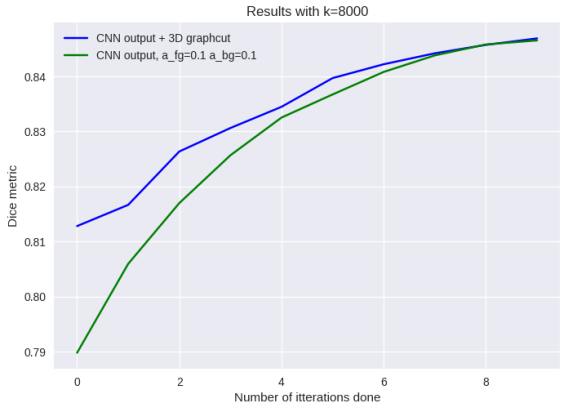

FIGURE 4.3 – Convergence between the graphcut output and the CNN output on the unlabelled data used for the training

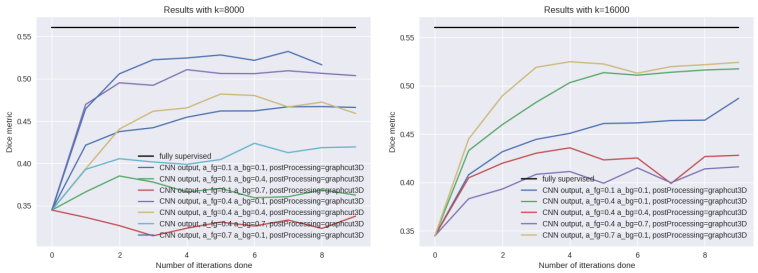

FIGURE 4.5 – Impact of α_{fg} and α_{bg} on the spine dataset

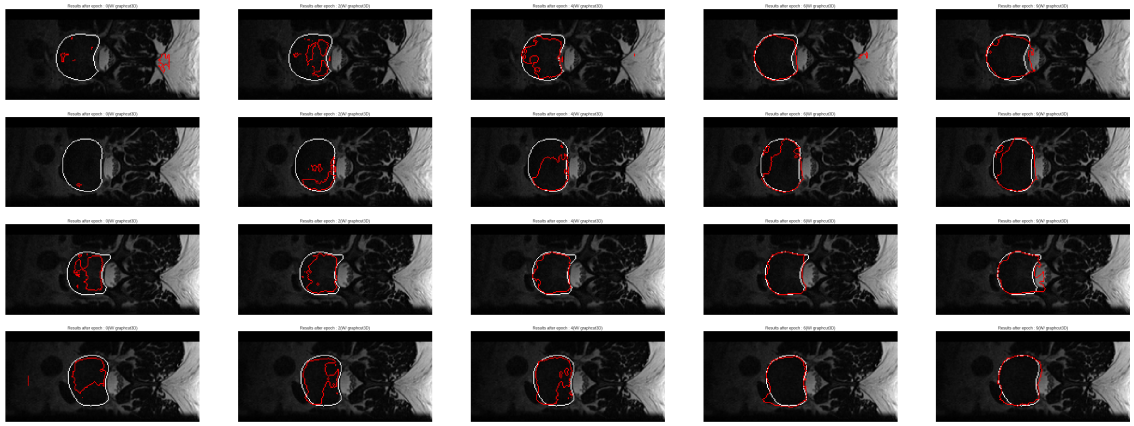

FIGURE 4.4 – Patient 25 : Results after each iteration (with 3D graph-cut improvement, K = 8000 and foreground weight loss = 0.4 and background weight loss = 0.1). The results are presented in this order : after epoch 0, 2, 4, 6 and 9. Red is the output of the CNN and white is the ground truth

4.6 Results Right Ventricle dataset

Some visual results can be seen at the figure 4.8. The first three rows are results for the test set, and the last ones for the unlabelled data used to feed the network during the iterative process. It shows than the CNN segmentation improves even for images which have never been seen by the network. That means that by optimizing θ and y_u, the energy function 3.3 is minimized indirectly for the test set. This is due to the fact that by introducing knowledge from the unlabeled dataset, the CNN learn more features and then adapt to others inputs images. For this application, we manage to increase the dice by 15%.

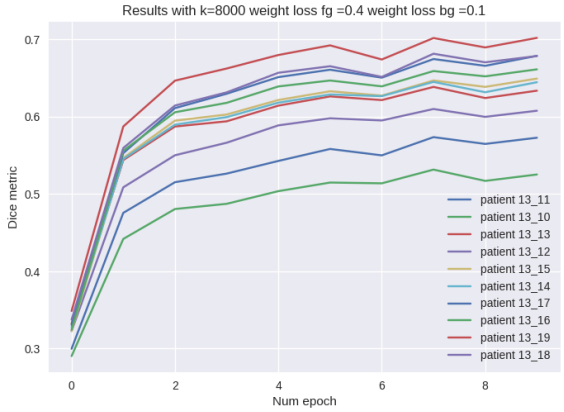

FIGURE 4.6 – Result after each iteration for some volume on the test set. With $\alpha_{fg} = 0.4, \alpha_{bg} = 0.1$

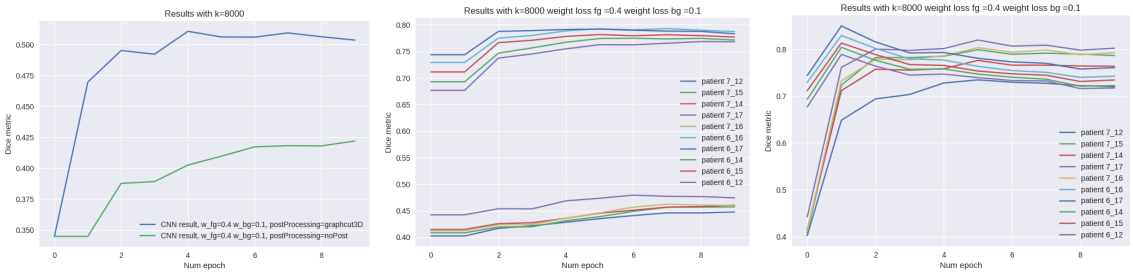

FIGURE 4.7 – Comparison with graphcut and without postprocessing for the right ventricle dataset on the test set. Middle : Result without post processing on the unlabelled dataset for some volume. Right : Result with 3D graphcut on the unlabelled dataset for some volume

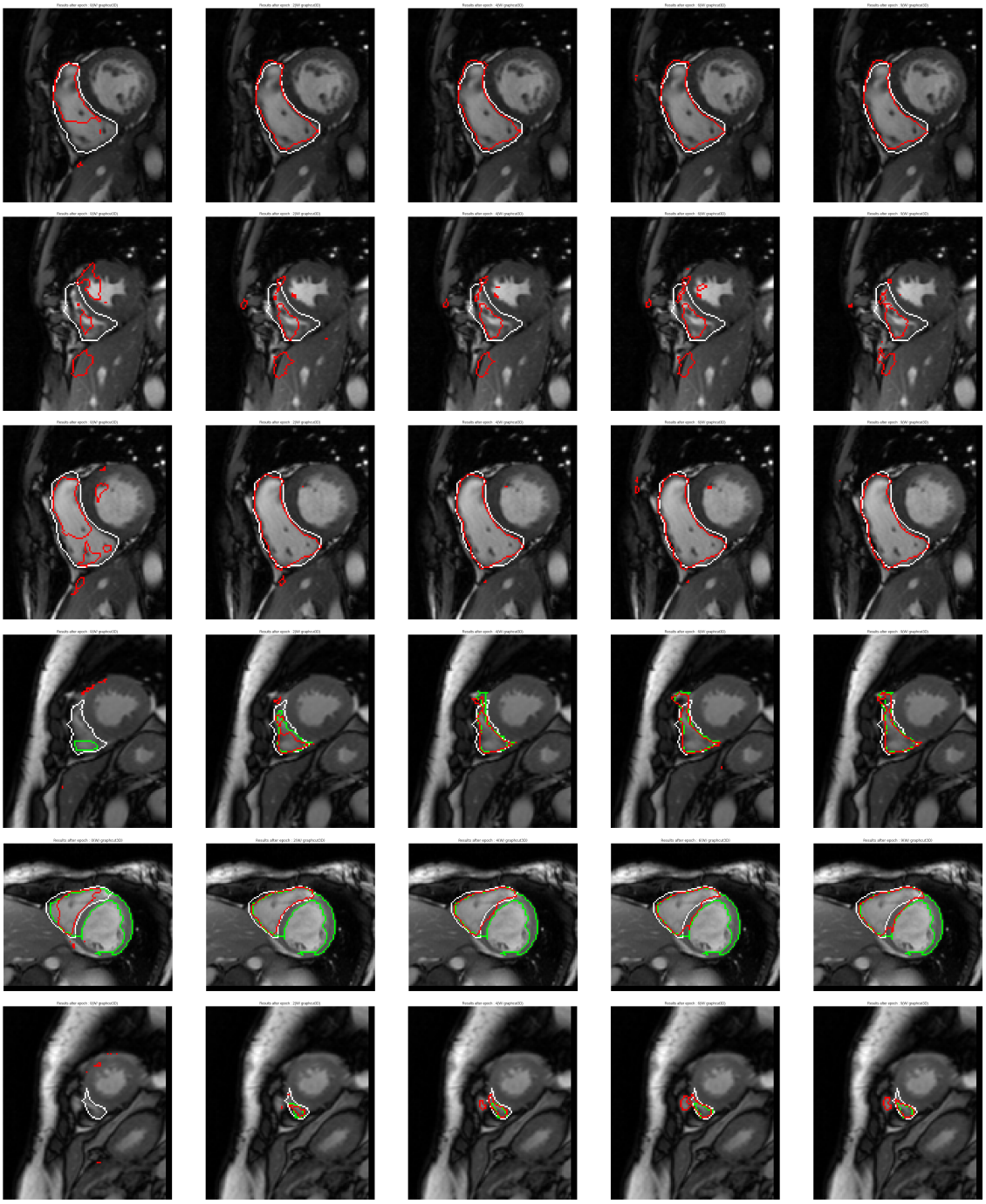

FIGURE 4.8 – Results after each iteration on the RV dataset (with 3D graphcut improvement, K = 8000 and foreground loss = 0.4 and background loss = 0.1). The results are presented in this order : after iteration 0, 2, 4, 6 and 9. Red is the result of the CNN, Green is the result after graphCut improvement and white is the ground truth. The first three rows are results on the test set, the last ones are on the validation set

4.7 Improvements and limitations

One of the limitations of this work is that the graphcut parameters must be defined manually, and can vary from an application to another. They can however be found by doing a search grid in a validation set. However, if the difference of contrast between the volumes is too high, it would be hard to find parameters which fit all the dataset, and the effectiveness of the iterative algorithm became limited.

An improvement can be to use a more advanced graphcut method with size constraint [14]. Then, the segmentation will be more accurate and it will indirectly constrain the CNN. On the other hand, more parameters have to be defined. Also, doing the training in 3D (considering all the volume) can help the Network to give a prediction consistent with the neighborhood slices.

Another way to apply the algorithm in a more efficient way would be to use Generative Adversarial Network [13] to generate new images which look real and use those images as unlabelled data for improving the segmentation.

Also, like neural networks, this algorithm has no guarantee to converge. That's why using a validation set for stopping the training would be very useful.

CHAPITRE 5

Conclusion

We propose a method allowing to use non-annotated data in order to improve the result of Convolutional Neural Networks for segmentation of medical images. The iterative algorithm able to work in two different datasets(right ventricle and spine) and is applicable to similar problems in medical images. It increased the dice up to 20 % for some subjects and reduce significantly the gap with the results obtained with the fully supervised setting. We studied the impact of adding two weights in the cross entropy loss of the CNN which shows that giving less importance to the pixels predicted background by the CNN is the best configuration. The results can be improved by using more sophisticated graphcuts methods and increasing the unlabeled data using Generative Adversarial Network.

Acknowledgement

I would like firstly to express my gratitude to my supervisors Christian DESROSIERS, Jose DOLZ, Ismail BEN AYED. Christians Desorisers for trusting me, have offered me this internship and the possibility to work on this project, always giving me accurate advice for improving my work and sharing his excellent expertise with optimization and CNN. Jose Dolz for keeping track of my progress, always having new ideas to improve the results, explaining things that I didn't know and sharing his passion for research with me. Ismael Ben Ayed for his extraordinary expertise in energy minimization and optimization problems especially applied for medical imaging.

I also would like to thank all my lab-mates(Lina, Rémi, Arlene, Fariba, Véronica, Eric, Helmie, Rute, Atafi, Houda, Jihen, Kuldeep, Mellie, Géraldo, Vin, Laura) who also became my friends. Thanks to them, coming to the lab every day was exciting.

I also would like to thank my master degree professors for teaching me all the basics about machine learning and deep learning which were essential for this work and especially Alexandre Allauzen for founding this master degree.

Bibliographie

[1] Convolutional neural network. http://cs231n.github.io/convolutional-networks/. Accessed : 2017-04-27.

[2] Livia description. https://en.etsmtl.ca/Unites-de-recherche/LIVIA/Accueil/. Accessed : 2017-08-21.

[3] Zeynettin Akkus, Alfiia Galimzianova, and Assaf Hoogi.

[4] Paul Aljabar, Rolf A Heckemann, Alexander Hammers, Joseph V Hajnal, and Daniel Rueckert. Multi-atlas based segmentation of brain images : atlas selection and its effect on accuracy. *Neuroimage*, 46(3) :726–738, 2009.

[5] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on pattern analysis and machine intelligence*, 26(9) :1124–1137, 2004.

[6] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab : Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv :1606.00915*, 2016.

[8] J Dolz, L Massoptier, and M Vermandel. Segmentation algorithms of subcortical brain structures on mri for radiotherapy and radiosurgery : a survey. *IRBM*, 36(4) :200–212, 2015.

[9] Jose Dolz, Christian Desrosiers, and Ismail Ben Ayed. 3d fully convolutional networks for subcortical segmentation in mri : A large-scale study. *NeuroImage*, 2017.

[10] Qi Dou, Lequan Yu, Hao Chen, Yueming Jin, Xin Yang, Jing Qin, and Pheng-Ann Heng. 3d deeply supervised network for automated segmentation of volumetric medical images. *Medical Image Analysis*, 41 :40 – 54, 2017. Special Issue on the 2016 Conference on Medical Image Computing and Computer Assisted Intervention (Analog to MICCAI 2015).

[11] Lester Randolph Ford Jr and Delbert Ray Fulkerson. *Flows in networks*. Princeton university press, 2015.

[12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[14] Lena Gorelick, Frank R Schmidt, and Yuri Boykov. Fast trust region for segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1714–1721, 2013.

[15] Tobias Heimann and Hans-Peter Meinzer. Statistical shape models for 3d medical image segmentation : a review. *Medical image analysis*, 13(4) :543–563, 2009.

[16] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe : Convolutional architecture for fast feature embedding. *arXiv preprint arXiv :1408.5093*, 2014.

[17] Konstantinos Kamnitsas, Christian Ledig, Virginia F.J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical Image Analysis*, 36 :61 – 78, 2017.

[18] Diederik P. Kingma and volume = abs/1412.6980 year = 2014 url = http ://arxiv.org/abs/1412.6980 timestamp = Wed, 07 Jun 2017 14 :40 :52 +0200 biburl = http ://dblp.uni-trier.de/rec/bib/journals/corr/KingmaB14 bibsource = dblp computer science bibliography, http ://dblp.org Jimmy Bal title = Adam : A Method for Stochastic Optimization, journal = CoRR.

[19] Lisa M Koch, Martin Rajchl, Tong Tong, Jonathan Passerat-Palmbach, Paul Aljabar, and Daniel Rueckert. Multi-atlas segmentation as a graph labelling problem : Application to partially annotated atlas data. In *International Conference on Information Processing in Medical Imaging*, pages 221–232. Springer, 2015.

[20] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. 2001.

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11) :2278–2324, 1998.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[23] Jyrki MP Lötjönen, Robin Wolz, Juha R Koikkalainen, Lennart Thurfjell, Gunhild Waldemar, Hilkka Soininen, Daniel Rueckert, Alzheimer's Disease Neuroimaging Initiative, et al. Fast and robust multi-atlas segmentation of brain magnetic resonance images. *Neuroimage*, 49(3) :2352–2365, 2010.

[24] Tim McInerney and Demetri Terzopoulos. Deformable models in medical image analysis : a survey. *Medical image analysis*, 1(2) :91–108, 1996.

[25] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[26] Alireza Norouzi, Mohd Shafry Mohd Rahim, Ayman Altameem, Tanzila Saba, Abdolvahab Ehsani Rad, Amjad Rehman, and Mueen Uddin. Medical image segmentation methods, algorithms, and applications. *IETE Technical Review*, 31(3) :199–213, 2014.

[27] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv :1502.02734*, 2015.

[28] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[29] Caroline Petitjean, Maria A Zuluaga, Wenjia Bai, Jean-Nicolas Dacher, Damien Grosgeorge, Jérôme Caudron, Su Ruan, Ismail Ben Ayed, M Jorge Cardoso, Hsiang-Chou Chen, et al. Right ventricle segmentation from cardiac mri : a collation study. *Medical image analysis*, 19(1) :187–202, 2015.

[30] Martin Rajchl, John SH Baxter, A Jonathan McLeod, Jing Yuan, Wu Qiu, Terry M Peters, and Ali R Khan. Hierarchical max-flow segmentation framework for multi-atlas segmentation with kohonen self-organizing map based gaussian mixture modeling. *Medical image analysis*, 27 :45–56, 2016.

[31] Martin Rajchl, Matthew CH Lee, Ozan Oktay, Konstantinos Kamnitsas, Jonathan Passerat-Palmbach, Wenjia Bai, Mellisa Damodaram, Mary A Rutherford, Joseph V Hajnal, Bernhard Kainz, et al. Deepcut : Object segmentation from bounding box annotations using convolutional neural networks. *IEEE transactions on medical imaging*, 36(2) :674–683, 2017.

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net : Convolutional Networks for Biomedical Image Segmentation*, pages 234–241. Springer International Publishing, Cham, 2015.

[33] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut : Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[34] Neeraj Sharma and Lalit M Aggarwal. Automated medical image segmentation techniques. *Journal of medical physics/Association of Medical Physicists of India*, 35(1) :3, 2010.

[35] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[36] Korsuk Sirinukunwattana, Josien P. W. Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, Bogdan J. Matuszewski, Elia Bruni, Urko Sanchez, Anton Böhm, Olaf Ronneberger, Bassem Ben Cheikh, Daniel Racoceanu, Philipp Kainz, Michael Pfeiffer, Martin Urschler, David R. J. Snead, and Nasir M. Rajpoot. Gland segmentation in colon histology images : The glas challenge contest. *CoRR*, abs/1603.00275, 2016.

[37] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[38] Robin Wolz, Rolf A Heckemann, Paul Aljabar, Joseph V Hajnal, Alexander Hammers, Jyrki Lötjönen, Daniel Rueckert, Alzheimer's Disease Neuroimaging Initiative, et al. Measurement of hippocampal atrophy using 4d graph-cut segmentation : application to adni. *NeuroImage*, 52(1) :109–118, 2010.

[39] Wei Xia, Csaba Domokos, Jian Dong, Loong-Fah Cheong, and Shuicheng Yan. Semantic segmentation without annotating segments. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2176–2183, 2013.