Insbria University

Insubria University
Department of Theoretical and Applied Sciences (DiSTA)
PhD Thesis in Computer Science
XXXIII Cycle of Study

UNIVERSITAS STUDIORUM · INSUBRIAE

Investigating the role of machine learning and deep learning techniques in medical image segmentation

Advisor: Prof. Elisabetta Binaghi
Scientific Supervisor: Prof. Elisabetta Binaghi, and Dr. Marco Marcon
External Reviewers: Dr. Valentina Pedoia, and Dr. Alberto Signoroni

Candidate
Gloria Gonella

December 2020

In memory of my Father

Contents

List of Figures

List of Tables

Acknowledgments

I want to express my gratitude to Prof. Elisabetta Binaghi for having entrusted me with such a delicate job. It was a unique experience that made me grow a lot, above all, thanks to his teachings that led me to the results obtained. It was a great honour to collaborate with the Department of Health Physics and Gynecology to create a system capable of supporting radiologists and finding their respective cancer pathologies.

I thank my scientific supervisor Dr. Marco Marcon, and Dr. Marco Paracchini of the Politecnico di Milano for the support given.

I also want to thank my thesis reviewers, Dr. Valentina Pedoia, and Dr. Alberto Signoroni for their instructive input and for giving me precious and priceless comments.

Thanks to my friends Dr. Albero Vergani and Dr. Christian Rondanini. I was lucky enough to share this great adventure called Insubria with you, and I know that, at the end of this journey, the laughter, the hours spent studying are something, I will remember for the next few times.

Thanks to my family, your sacrifices and support have been vital to me this year.

Thanks to my husband, Samuele, you are the love of my life. Since I met you my life has changed and, even if I don't know what the future holds, I will do everything to be by your side, to support you as you have done with me in these difficult months. I firmly believe that together we can achieve such beautiful goals. I love you.

Thanks to all the friends near and far who help me and who brought me here. I firmly believe that all of you have played a significant role in the events of my life and all the difficulties we have faced together seem very few and insignificant compared to our great friendship.

Ringraziamenti

Vorrei esprimere la mia gratitudine alla Prof.ssa Elisabetta Binaghi per avermi affidato un lavoro così delicato. E' stata una preziosa esperienza che mi ha fatto crescere molto, soprattutto grazie ai suoi insegnamenti che mi hanno portato ai risultati ottenuti. È stato un grande onore collaborare con il reparto di Fisica Sanitaria e di Ginecologia per creare un sistema capace di supportare i radiologi a trovare le rispettive patologie tumorali.

Ringrazio il mio supervisore Dr. Marco Marcon, e Dr. Marco Paracchini del Politecnico di Milano per il supporto dato.

Voglio anche ringraziare i miei revisori di tesi, Dr. Valentina Pedoia, e Dr. Alberto Signoroni per il loro contributo istruttivo e per avermi fornito commenti preziosi e inestimabili.

Grazie ai miei amici del team Dr. Albero Vergani, Dr. Christian Rondanini. Ho avuto la fortuna di condividere con voi questa grande avventura chiamata Insubria e so che, alla fine di questo viaggio, le risate, le ore passate a studiare sono qualcosa che ricorderò per i prossimi tempi.

Grazie alla mia famiglia, i vostri sacrifici e il vostro sostegno sono stati vitali per me in quest'anno.

Grazie a mio marito, Samuele, sei l'amore della mia vita. Da quando ti ho incontrato la mia vita è cambiata e, anche se non so cosa ci riserva il futuro, posso solo giurare che farò di tutto per stare al tuo fianco, per sostenerti come hai fatto con me in questi mesi difficili. Credo fortemente che insieme possiamo realizzare traguardi così meravigliosi. Ti amo.

Grazie a tutti gli amici vicini e lontani che mi aiutano e che mi hanno portato qui. Credo fortemente che tutti voi abbiate giocato un ruolo molto importante negli eventi della mia vita e tutte le difficoltà che abbiamo affrontato insieme sembrano pochissime e insignificanti rispetto alla nostra grande amicizia.

1 Introduction

This work originates from the growing interest of the medical imaging community in the application of machine learning techniques and in particular, from deep learning to improve the accuracy of cancer screening. For this reason, the Thesis is structured into two different tasks. In the first part, magnetic resonance images were analysed in order to support clinical experts in the treatment of patients with brain tumour metastases. This task is the outcome of a close collaboration between the Research Center for Image Analysis and Medical Informatics of the Insubria University and the Operative Unit of Neurosurgery, Neuroradiology and Health Physics of the University Hospital “ASST dei Sette Laghi”, Varese. The project aim is to investigate new methodologies using whose, develop an integrated framework able to enhance, and in some cases make possible, the use of the information contained in the Magnetic Resonance Images, in order to support clinical experts in the treatment of patients with brain Metastases tumour. The goal was to provide tools for semiautomatic segmentation of spaces where metastases are present, to accelerate the segmentation phases defined slice by slice of the area occupied by the tumour. The second task of the thesis was developed in collaboration between the Research Center for Image Analysis and Medical Informatics of the Insubria University and the Politecnico of Milano and the MultiMedica Group with the Breast Centre of the Hospital “San Giuseppe”, Milano. This part of the Thesis analyses mammograms in search of calcifications or masses present in the patient analysed. As a final goal to integrate the system developed in clinical practice, for both fields studied, all the Medical Imaging and Pattern Recognition algorithmic solutions studied for this PhD

Thesis have been integrated into a software package. A MATLAB® prototype was developed with an integrated analysis environment that offers, in addition to all the features described explicitly in this thesis, a series of tools necessary to manage the visualisation and information deriving from the Dicom format of the exam analysed.

1.1 The role of MRI segmentation in Metastasis Tumors Treatment

Brain metastases (BMs) are one of the most common neurological neoplasms, and their incidence is increasing with the availability of advanced imaging techniques such as Magnetic Resonance Imaging (MRI) [12, 32]. By visual inspection of MR scans, physicians can accurately examine and identify pathological tissues thanks to the high spatial resolution and contrast and the enhanced signal differentiation. In the clinical practice, MRI has been confirmed as a significant approach supporting diagnosis, surgical planning, follow up and therapy. In order to exploit this potential, intelligent techniques should complement image acquisition and visualisation tools, addressing relevant issues such as cancer detection. Extensive research has already been developed to introduce computer-aided detection and segmentation methods in neuro-oncology clinical studies. Available methods include image-based methods and machine learning-based methods [31, 49]. The image-based methods use image data and image processing techniques to detect and delineate lesions. Exemplary techniques include tissue classification based on Raman spectroscopy [38], a colour-coded map from quantitative optical coherence tomography (OCT) for differentiating cancer from non-cancer in human brain tissues [42], watershed segmentation algorithms [9], active contour algorithms [37], and region growing segmentation algorithms [67]. Supervised machine learning (ML) approaches have been successfully applied to circumvent the problem of explicitly and analytically describing the specific segmentation procedure and related parameters, lying to a learning stage the charge of inducing the classifier from supervised data available. The proposed techniques make use of a single image or multi-spectral pattern and are interactive or fully automatic [31, 6, 2, 58, 8, 63, 7, 27]. Among the most promising methods, we found the support vector machine (SVM) [73, 19], discriminative models based on Random Forest and logistic regression [71, 76]. Cai et al. [13] and Verma et al. [74] proposed classification methods based on SVM to classify brain neoplasms and their sub-components using multidimensional patterns obtained by a high number of MRI modalities. Ruan et al. [62] proposed SVM to segment lesions using a lower number of modalities. Bauer et al. [5] adopt a hybrid method, based on SVM and hierarchical regularisation, to segment tumour and healthy tissues, including sub-compartments. Random Forest-based methods are proposed by Zikic et al. [77] to identify brain tumour sub-compartments from multi-modal images and by Geremia et al. [26] that generate synthetic tumour images to train a discriminative regression forest algorithm using different groups of features.

Although the problem of segmenting all or part of the brain in MRI imagery does continue to be investigated attempting to satisfy the high accuracy demand in diversified clinical and neuroimaging application, only a few studies have applied such ML approaches on BM detection and segmentation [14]. BM requires specific approaches given their small size and multiplicity and the stringent requirements of RT clinical practice in which usually segmentation procedures are inserted. In this application context, MRI T1c images are usually the only modality considered and fast segmentation is required for a rapid clinical workflow.

In their review, Perez et al. [59] describe methods that use hand-crafted templates or blob based complex procedures. These works report performances strongly affected by feature extraction methods, which becomes more and more complicated to improve results. Machine learning algorithms such as SVM and Random Forest are proposed in combination with spatial regularisation procedures and Gaussian processes to refine patch-based segmentation [75]. Despite the achievements obtained, automated segmentation of brain metastases and lesions, in general, remains an unsolved problem due to normal anatomical variations in brain morphology, variations in acquisition parameters of MRI scanners, the heterogeneous appearance of pathology.

Recent BM segmentation studies propose the use of deep learning methods using different convolution neural network (CNN) architectures and different MRI sequences as input data. Losch et al. have been among the first that investigated the use of deep networks to detect and segment BM [51]. Their work compares several types of spatial inputs and network topology to find performances comparable to conventional state-of-the-art models. The importance of database quality has also experimented. Growick et al. propose the use of a CNN based on the GoogLeNet architecture for automatic BM detection and segmentation. The retrospective study focusses on 156 patients with brain metastases from several primary cancers and makes use of multi-sequence MR images, including pre- and post-gadolinium T1-weighted and FLAIR scans. Results obtained are good, but several limitations because of limited sample size and false-positive results near vascular structures, are highlighted.

The use of multi-sequence MRI limits the applicability in the clinical domain. Charron et al. [14] studied the influence of MRI modalities together with the impact of the number of segmented classes, both on the detection and segmentation of brain metastases in MRI imagery. To do this, they used a modified version of DeepMedic neural network proposed by Kamnitsas et al. [39] and data augmentation strategies. The combined use of different MRI modalities outperformed the performances of the network when using single modalities. In single modality, best performances are obtained using volumetric T1c scans. Liu et al. investigate the use of deep learning convolutional neural network (CNN) algorithm trained on both gliomas and BM [48] on T1c.

In radiation therapy (RT), the workflow for BM treatment requires fast and accurate detection and delineation of tumour volume of small size on MRI scans. Contrast-enhanced T1 (T1c) magnetic resonance imaging (MRI) is generally the only imaging modality adopted. Segmentation accomplished through a complete manual tracing is still the standard routine although it has shown to be a time-consuming and labour-intensive process affected by high intra- and inter-observer variability. In this context, ML techniques would be very helpful, actively supporting human experts in tracing the boundaries of the pathological tissues with varying degrees of automation and improving the efficiency of the overall radiotherapy clinical workflow.

Promising results obtained by recent studies in automated BM segmentation [49, 14, 59, 33] lead us to further investigate ML techniques derived from both conventional feature-based learning and deep learning approaches and measure and compare their capability to compute spatially accurate and stable results. Indeed, both approaches have their advantages and disadvantages.

In the design of an automated segmentation procedure, the feature extraction phase plays a major role. Deep learning methods offer the advantage of automatically learn hierarchical image representations which often outperform the most effective features extracted by well-assessed procedures. However, good performances are obtained provided as large as sufficient amount of precisely annotated training sets are available. This requirement is especially critical in medical image analysis, in particular in BM segmentation where annotated consistent training data are difficult to collect and at best of our knowledge, no public datasets of diagnosed patients with metastatic brain tumours are available. Conventional ML models have shown excellent performances in MRI brain tumours segmentation studies even when using a much smaller training set. However, rarely these methods alone offer the opportunity to cope with the complete segmentation task and are usually complemented with pre- and post-processing procedure to overcome limitations of hand-craft features and improve spatial consistency in classification results.

The remainder of this work is organised as follows in chapter 3. Section 3.1 introduces the conceptual segmentation frameworks based on SVM and V-Net model, respectively. Section 3.2 describes the experimental evaluation and comparison of their performances using in-house collected datasets of different size. Section 3.3 illustrates the main features of the software package and the use of the system in clinical studies. Section 3.4 reports the discussion and conclusions.

1.2 The role of Mammography segmentation in Breast Cancer Treatment

Breast cancer is one of the major causes of death among women [11]. Mammography has a central role in screening and diagnosis, allowing early detection of the pathology and reduction of fatal cases [45]. During a mammographic examination, the angles of image acquisition are 2 for each breast, bilateral craniocaudal (CC) and mediolateral oblique (MLO). The visual inspection of these images shall enable radiologists to examine and identify tissues carefully, looking for malignant lesions in the form of clusters of microcalcifications and masses. Unfortunately, the evaluation of mammograms is a complicated and time-consuming process that inevitably affected by high intra- and inter-observer variability [3].

In this context, automated methods of image analysis represent a valuable solution by supporting human operators with varying degrees of automation in detecting important clinical signs [45].

Over the last three decades, many solutions for Computer-aided detection (CAD) of breast cancers have been proposed [22, 44]. Several techniques have been investigated in studies distinguished by the type of image analysis and classification techniques adopted. Despite the sizable achievement obtained, accurate detection of breast cancer tissues in mammograms remains very difficult mainly due to their fuzzy, heterogeneous nature, low distinguishability from the healthy backgrounds and low contrast of the mammographic imagery [45].

Novel approaches are continuously investigated to provide robust solutions and fulfil hard accuracy and reproducibility requirements. Early solutions of CAD systems principally based on image analysis and conventional pattern recognition techniques [24] had not led to an improvement in diagnostic accuracy when inserted in clinical practice. The utilization of conventional "feature-based" Machine Learning (ML) procedures have ensured a significant improvement in both screening and diagnoses. However, these methods can only work well if features extracted from image data are meaningful to discriminate pathological and healthy classes. Several studies highlighted limitations of CAD system based on conventional ML techniques identifying the feature extraction phase the bottleneck of the overall design process [25].

Promising results obtained by recent studies in automated biomedical image segmentation using Deep Learning (DL) techniques and in particular, the Convolutional Neural Networks (CNN) [25, 1] have opened up a new opportunity to address the issue of early detection of breast cancer with an acceptable level of accuracy and reproducibility. Deep learning methods offer the advantage of automatically learn hierarchical image representations which often outperform the most effective handcrafted features. Although results obtained by recent studies are promising, several aspects in the application of CNN for breast cancer detection must be furtherly investigated to evaluate how these novel techniques can be inserted in clinical practice. These aspects include the need for large annotated training sets. In medical image analysis, training data are challenging to collect, and the annotation is easily affected by inconsistencies that influence the learning process and may cause low accuracies in classification.

The remainder of this work is organised as follows in Chapter 4. Section 4.1 introduces the types of mammography data-sets available online. Section 4.2 is dedicated to the description of the pre-processing procedures. Section 4.3 illustrates the proposed methods for this work with the respective network architectures and related training procedure. Section 4.4 describes the experimental evaluation and comparison of their performance using two data-sets CIBIS-FSM and INbreast-FFDM of different sizes. Section 4.5 illustrates the main features of the software package and the use of the system in clinical studies. Section 4.6 reports the discussion and conclusions.

1.3 List of abbreviations

AI Artificial Intelligence
ANN Artificial Neural Network
AUC Area Under Curve
BM Brain Metastases
CAD Computer-Aided Detection
CNN Convolutional Neural Network
DICOM Digital Imaging and Communications in Medicine
DL Deep Learning
DSC Dice
FFDM Full Field Digital Mammography
GUI Graphical User Interface
ML Machine Learning
MO Morphological Operators
MRI Magnetic Resonance Imaging
P Precision
R Recall
ROC Receiving Operating Characteristic
ROI Region Of Interest
SFM Screen Film Mammography
SVM Support Vector Machine
VOI Volume of interest

1.4 Tesis Organization

In the introduction, I define the purpose of this thesis and review the literature related to the two tasks developed in this thesis. Chapter 2 describes some basic concepts useful for understanding the proposed approaches, used in Chapter 3 for task 1 and Chapter 4 for task 2. In particular, in Section 2.1, the algorithm concerning the classification of Brain Metastasis inherent to the task 1 is reported, starting from the works that use traditional classifiers SVM (Sec. 2.1.1) to the use CNNs where the basic concepts are explained in Section 2.1.3. Also, in task 2, in addition to the CNNs, the pre-processing operator was used to analyze the image, the Hough transformation present in Section 2.3.1. These notions are useful for understanding our proposed approaches and the advantages of using this type of architecture to perform classification tasks. The datasets used are described in Section 3.2.1 and 4.1 of the respective tasks, and the process of evaluating in Section 2.2 for the respective tasks used to interpret the results of the experiments of Sections 3.2 and 4.4. General conclusions and future works are presented in Chapter 5. Finally, in chapter 6, the code used in chapter 3 for the study of metastases with V-Net is reported.

2 Background

In this thesis, the images used derive from the MRI scanner (task one) and the Mammography (task two); these images have been used in the field of supervision learning algorithms since the single- voxel or pixel has been labelled with the pathological or non-pathological class by the experts. A large amount of data available also allows us to use deep learning for the artificial vision, which deals with the theory behind artificial systems that extract information from images. For task one, the SVM (Sec.2.1.1) classifier was used in conjunction with morphological post-processing operators (Sec.2.3.2) and a particular Convolutional neural network (Sec.2.1.3) called V-Net. For activity two, on the other hand, two different types of convolutional networks were used after the application of the Hough transform (Sec.2.3.1) to determine only the breast-target analysed by CNN. The models and techniques used to classify the two tasks are analysed in the next section and the relative evaluation metrics (Sec.2.2).

Some basic concepts useful for understanding the proposed approaches, used in Chapter 3 for task 1 and Chapter 4 for task 2, are described in the following.

2.1 AI Algorithms

Artificial intelligence (AI) is evolving into a significant focus in medicine by addressing problems of inconsistency and inter- and intra-observer variability during the acquisition and interpretation of images. AI includes any technique that allows computers to imitate human intelligence includes machine learning and deep learning. Many terms are used to describe different subfields and methods within AI (Fig. 2.1).

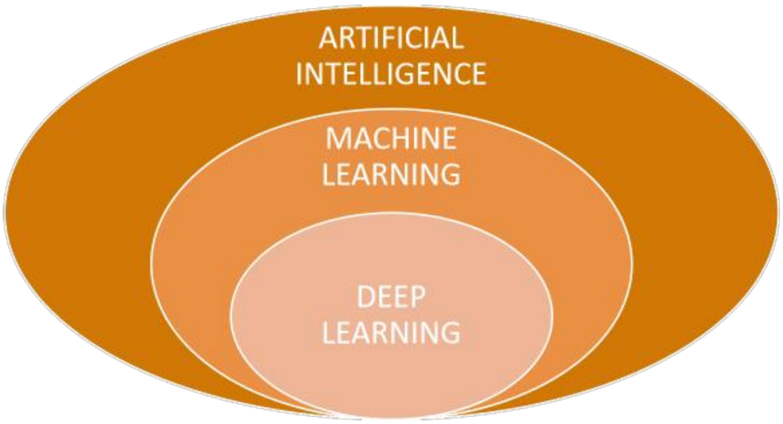


Figure 2.1: Schematic showing the hierarchy among AI, ML, and DL. Machine Learning and Deep Learning are sub-fields within Artificial Intelligence.

Machine learning is a subset of artificial intelligence that consists of a statistical method that allows machines to improve at the level of experience, including deep learning. ML is a system that is in an evolving environment that should have the ability to learn. If the system can learn and adapt to this change, the system designer is not required to foresee and provide solutions for all possible situations. Machine learning is used in many fields, in our study in the field of medicine learning programs are used for medical diagnosis. To use machine learning algorithms, the first thing to do is to find the best criterion for separating the data classes of the training set, thus selecting features to use the model. Deep learning is the subset of machine learning, is composed of algorithms that enable the software to learn from the examples to perform tasks, such as speech or image recognition, thanks to the exposure of multi-layered neural networks with large amounts of data (Fig. 2.1).

Supervised learning is a machine learning technique that aims to train a system to allow it to solve tasks independently based on a series of ideal examples, consisting of pairs of desired inputs and outputs, which are initially provided. That said, every supervised learning algorithm needs training dataset, S, which consists of N data belonging to C different classes.

$$S = \{x_n, y_n\} \mid n = 1, \dots, N;\ x_n \in R^D y_n \in C \tag{2.1}$$

Where x_n is a D-dimensional vector (or pattern) whose components are called attributes, y_n indicates the class to which the data belongs, and C is the set of possible classes. The mapping function $f : y_n = f(x)$, is not known, and a learning algorithm aims to define it. A supervised learning algorithm, therefore, looks for the function f

which assigns its class to each datum. Each statistical learning process is divided into two phases:

1. Learning phase, where the algorithm analyzes the training data and recognizes the similarities in the data to build a model that approximates f

2. Classification phase, where the model generated during training is tested on a different dataset to verify its performance

The classification activity has the purpose of organizing the entities of a given domain so that they can be exposed using criteria that enjoy certain rationality.

We can think of these machines as an alternative technique to the classical training techniques of neural networks. We could use single-layer neural networks, but in this case, we could work only in the case of linearly separable data. In the case of using multilayer neural networks, we could represent non-linear functions, useful for solving cases with non-separable data, but it must be considered that these networks present a series of problems that complications train. These difficulties are due to the high number of dimensions of the weight space and to the fact that the most common techniques have been obtained the network weights starting from the solution of a non-convex and unconstrained optimization problem which, consequently, has a number indeterminate of local minima. The technique used to train an SVM solves both questions: we have an efficient algorithm capable of representing nonlinear functions. The required parameters are also obtained from the solution of a convex quadratic programming problem with equality constraints that provides for a single global minimum. In the next section, we are seeing in detail the AI algorithms used for the processed task.

2.1.1 Support Vector Machine

The support vector algorithm has been developed [10, 18] approximately 20 years ago. Support vector machines (SVMs) are a generalization of methods based on a linear decision boundary identification for classification. SVMs produce nonlinear boundaries by constructing a linear boundary in a large, transformed version of the feature space using an appropriate kernel function [23]. This technique is based on the concept of choosing centres points critical for the classification, called support vectors and is well suited for the non-separable case, in which classes are overlapped. Indeed, a margin surrounding the boundary line is identified, and its width is inversely proportional to the value of a cost parameter C that has to be optimized. Given a set of training samples N called training sets, each of which is labelled with the class to belong to among the possible classes, an SVM classifier creates a model that will assign a classification label to the new input data. An SVM model is a representation of the training data as points in space and dimensions, mapped in such a way that the examples belonging to different

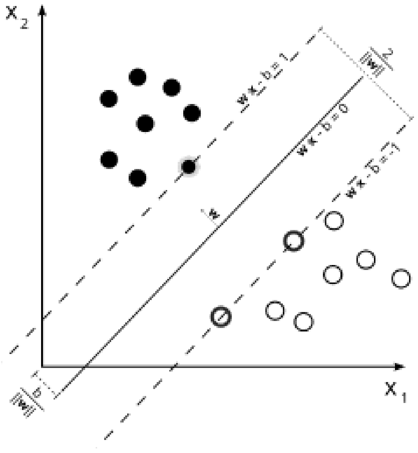


Figure 2.2: Example of classification of two classes with hyperplane representation of the SVM algorithm.

classes are separated from an area as large as possible. The new samples to be classified are then mapped in space, and the prediction of the belonging class is made based on the side on which this sample falls. The classifier calculates the distances from the support vectors (hyperplanes, in the case of N-features dimensions) identified during training. It should be noted that the SVM classifier is a binary classifier: given a set of classes, this is reduced from time to time to two classes, the class considered against the conjugated class (in our case, for example, tumour and not-tumour). The distances will be calculated as many times as the value of N-classes, and the class in which the new champion is most distant in the direction of the class itself will be provided. In detail, suppose you have a set of N separable data with which you want to train the Support Vector Machine, and you want to perform a binary classification, that is, with only two possible classes. The training data is described as:

$$S = \{x_n, y_n\} \mid n = 1, \dots ,N; \; x_n \in R^D y_n \in C \tag{2.2}$$

where y_n is a label that can assume one of the two values provided for identifying the class it belongs to, while x_n a vector containing the attributes that characterize the data. If the data, for example, is an image formed by 1000 pixels, the vector x_n will contain 1000 elements, each indicating the numerical representation of the colour of the single pixel. In the case of Support Vector Machine, the data with which you train can be positive or negative (respectively the value of y_n will be 1 or -1). The purpose of the SVM is to find a separation hyperplane that divides these two classes in the best possible way. The points x found on the hyperplane satisfy, as already seen, the equation:

$$x_n \cdot \omega + b \geq +1 \quad for \quad y_n = \; +1 \tag{2.3}$$

$$x_n \cdot \omega + b \leq +1 \quad for \quad y_n = \ -1 \tag{2.4}$$

Being in the binary case and assigning the labels as illustrated above, it is possible to combine these two conditions into a single inequality:

$$y_n \left(x_n \cdot \omega + b\right) - 1 \geq 0 \quad \forall n \tag{2.5}$$

Only the valid points are considered since both in the case of positive data and in the case of negative data, the product is positive. Considering the points against equality alone (Eq.2.3). These points are positioned on the hyperplane $H_1 : x_n \ \cdot \ \omega \ + \ b \ = +1$ while, similarly, the points that operate the equality (Eq.2.4) are placed on the hyperplane H_2 : $x_n \ \cdot \ \omega \ + \ b \ = \ -1$. The other points will be found in the space external to that delimited by the planes H_1 and H_2 going to verify the strict inequality (Eq.2.5). On the other hand, no point can be found in the space inside that bounded by the marginal planes. Considering the definitions given in the previous paragraph, the distance between the separating hyperplane and the hyperplane H_1 is the same as that between the separating hyperplane and the hyperplane H_2 and is $d_{\,+} = \ d_{\,+} = \ 1 \ / \ \omega$. That said, the width of the margin is simply the sum of these two distances, which is $2 \ / \ ||\omega||$. The hyperplanes H_1 and H_2 are parallel, both having the same vector ω as an orthogonal vector, we set ourselves the goal of identifying the two hyperplanes that give the largest margin, minimizing $||\omega||^2$ under the condition (Eq.2.5). As seen previously, the training points that find equality (Eq.2.5) are those that are precisely above one of the hyperplanes that determine the margin. These points are of fundamental importance, as their removal modifies the solution found. These points are called support vectors, and in the figures, they are represented by a circle (Fig.2.2).

2.1.2 Artificial Neural Network

Before analyzing the structure of Convolutional Neural Networks, it is necessary to see the definition of classical neural networks, models with a high number of parameters inspired by the architecture of the human brain and promoted as universal approximators, i.e. systems which, if fed by a sufficient quantity of data are able to learn any predictive relationship. Invented in the 1980s, neural networks enjoyed a relatively short period of popularity, as they were soon shelved by more recent inventions such as boosting and support vector machines. Thanks to the considerable improvement of computational resources in 2010 they returned to use, so the use of deep learning algorithms opened the way to new perspectives for solving complex tasks such as the classification of images and videos.

The inspiration for the neural network comes from brain function and its ability to learn. Therefore, taking into consideration the neuron as the essential elements of the nervous system, which can be considered the primary computational unit at the basis of our intelligence, making it as a mathematical model it was possible to build artificial networks. The artificial neural network model is composed of many neurons connected and joined in layered groups where each neuron can learn a primary function [40]. There are several parameters that the network learns during the training phase to carry out the final classification task.

Artificial Neural Networks Artificial is a supervised machine learning model where each neural unit is connected to many others, and the connection can be of a reinforcing or inhibitory type towards the activation of the units to which it is connected. Each neuron contains a function used to combine the values of all its inputs and a function, called the trigger function, which returns the neuron's output. The general form of the overall function contained in a neuron is represented by the following general equation:

$$y = f\left(\sum_i w_i x_i \ + \ b\right) \tag{2.6}$$

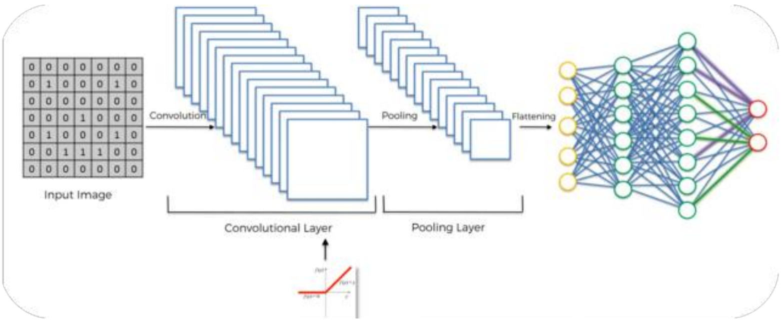
In this Eq.2.6, w_i are the weights assigned to each input being combined and b a term, called bias, which is added later. The set of weights and bias represent the information that the neuron learns during the training phase and that it conserves subsequently. The f function represents the activation function, which usually consists of a threshold or regulation function that causes only signals with values compatible with the threshold or limit imposed to propagate to the next neuron or neurons. Typically, the activation function is a non-linear function, and in the details, it is a step function, a sigmoid or a logistic function. Therefore, the training phase in supervised learning of a network consists of estimating the weights w_i contained in each neuron which minimizes the error between the output values expected from the training data and the values predicted by the network. The function that calculates this error is commonly called loss function, can be of different types. Weight estimation can be obtained with known mathematical optimization techniques. The technique is typically used is that of gradient descent using backpropagation [61]. It is a cyclical technique in two phases: the propagation and updating of the weights. In the first stage (forwarding propagation), the inputs traverse the entire network from the input level to the output level. After propagation, the produced outputs are recovered, and through the loss function, the prediction error concerning the expected outputs is calculated. This error is used for the calculation of the gradient of the loss function, which is then propagated backwards in the network until each neuron obtains its gradient value [70]. At this point, the weights update phase begins. In this phase, the calculated values of the gradient are given as input to the gradient descent algorithm, which uses them to update the weights of each neuron, to minimize the value of the loss function. Specifically, to find the minimum of the loss function, the gradient descent updates the network weights using steps proportional to the value that the gradient assumes at a moment of the computation. Now the network output is directed in the direction of the gradient, with the effect of reducing the value of the loss function in each cycle. The constant of proportionality used for updating the weights is called “learning rate”.

In real use, this term is usually variable and generally has a decreasing trend with the continuation of training. In this way, the network can take more significant steps in the direction of the gradient at the beginning of the training, when it is generally far from the minimum value, and reduce the number of updates when it begins to approach it, so to facilitate convergence and avoid unwanted divergences. A possible adverse scenario deriving from the use of this technique arises when the search for the minimum remains blocked at a local minimum, which prevents the network from achieving the expected result. There are two methods of execution for the management of training cycles, the stochastic one and the group one. In stochastic learning, each forward propagation step is immediately followed by a weights update step. In group learning, the propagation step is carried out for each example in the training set and only then are the weights updated using the gradient accumulated between all the propagations. Stochastic learning generally introduces a certain amount of “noise” into the training process, since at each step, it uses the gradient calculated at a single point in the data space; the risk of getting stuck in a local minimum but also the increased risk of divergence or non-convergence. Training in the group, has the advantage of leading directly to the result, since the amplification of the weights considers the entire training set. Unfortunately, its implementation is infeasible due to significant limitations regarding the performance and memory resources required. A compromise typically adopted in modern applications represents a middle ground between the two approaches and consists in the use of so-called “mini-batches”, or small sets containing examples randomly selected from the available data. One of the most important and interesting aspects of training is the ability of the neurons of the intermediate layers to organize themselves independently so that each of them learns to recognize different features in the input space. After the training phase, when a new input is fed to the network, the neurons of the hidden levels will be able to respond with an active state if the given input contains a particular pattern that resembles a feature that each neuron has learned to recognize.

2.1.3 Convolutional Neural Networks - CNN

The CNN model is part of the deep learning, the architectures that are used for Deep Learning refer to neural networks composed of different layers, each of which is made up of a certain number of nodes, the neurons, which perform non-linear transformations on the data coming from the previous layer. Deep Learning refers to a class of Machine Learning techniques for classifying data and extracting features [35]. These techniques allow them to learn a representation of data on different levels of abstraction and are applied to computational models with a layered structure. The algorithms used to make it possible to extract the complex internal structure of a large set of data and indicate to the layered model how to modify its internal parameters. These parameters, in each layer, are updated to create its own representation of the data starting from the representation present at the previous level. Deep Learning methods are also called representation-learning methods at multiple levels of representation, each of which is obtained through the composition of simple but non-linear modules. Each module, starting from a set of raw data, transforms the representation from one level to another with an ever-higher degree of abstraction. The composition of a certain number of transformations allows them to learn very complex functions. As part of the classification, the higher levels of representation allow you to amplify some important aspects of the input and eliminate irrelevant variations. An image, for example, takes the form of an array containing the pixel values and the features learned in the first level of representation, typically, indicate the presence or absence of edges in particular positions and with different orientations. The second layer, on the other hand, searches for patterns in the image by identifying specific arrangements of the edges regardless of small variations and their positions. The third level assembles the patterns producing many shapes, and the subsequent layers identify the objects in the image as a combination of these shapes. The key aspect of Deep Learning lies in the fact that these levels of features are not studied and designed by man but are learned from the input data using a general-purpose learning procedure. Since the early 2000s, Deep Learning has been applied with great success in various fields for the detection, segmentation and recognition of objects and regions present in images. Among these, we are starting to see exciting results in the classification of images representing road signs [16] and as in our study for the use of segmentation of biological images [55]. One of the sectors in which Deep Learning techniques have led to the achievement of more extraordinary results is that of image processing, especially with the use of convolutional neural networks, whose structure and functioning is described in detail below. We see in detail from what type of deep neural network, the structure of the CNN is composed together with an example.

The convolutive neural networks are multi-level neural networks (Fig.2.3), similar to the Multilayer Perceptron, however with a very particular structure possessing at least one layer composed of similar repeated structures, or replicas of artificial neurons that share among their weights: these layers are listed convolutive (convolutional layer). Each neuron receives some inputs, runs a product, and optionally follows it with a non-linearity. The entire network still expresses a single function that can differentiate the score: from the raw image pixels on the one hand to the class votes on the other. Moreover, they still have a loss function (for example SoftMax) on the last level (completely connected).



0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Figure 2.3: CNN example for binary classification.

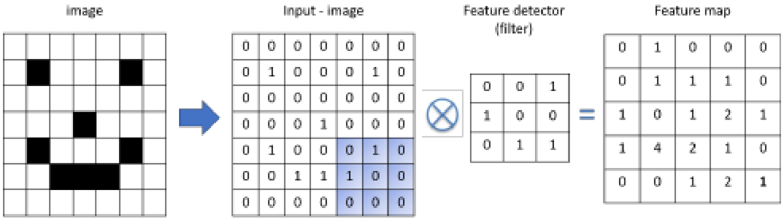
CNN in Fig.2.3 is thus composed of the following blocks:

1. **Convolution**

   Given an image x with k channels and a kernel w, the convolution operator creates a new image y in the following way:

$$y_{i'j'k'} = \sum_{ijk} w_{ijkk'} \cdot x_{i+i',j+j',k} \tag{2.7}$$

   where k' corresponds to the index of filters/kernels in the convolution. The image is converted into a matrix containing only 0/1 if it is an image in black and white, otherwise if in greyscale the px will have a value between 0 to 255 or in the case of RGB colour images will be created 3 matrices one for red, one for green and one for blue with pixel values between 0 and 255, an example below (Fig.2.4).



Input - image

0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Feature detector (filter)

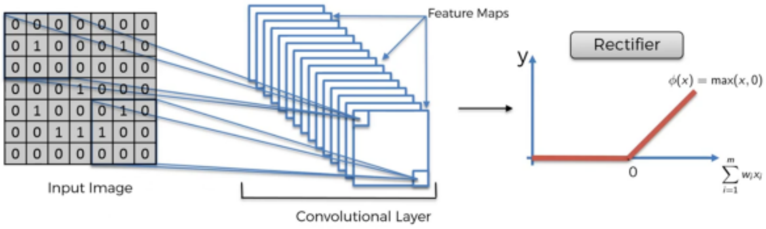
0	0	1
1	0	0
0	1	1

Feature map

0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

Figure 2.4: Example to the Feature map.

   Multiply the coloured matrix with the filter or kernel 3x3 and add as many "1" have in common. So now we have the feature map, which is our small starting image. This is our featured detector step, surely we lose information with this feature detector, because we have fewer values in the resulting matrix, but the aim is to detect the most essential features (as the human eye filters and removes useless features for the classification is not that we look at the single-pixel but the whole image).



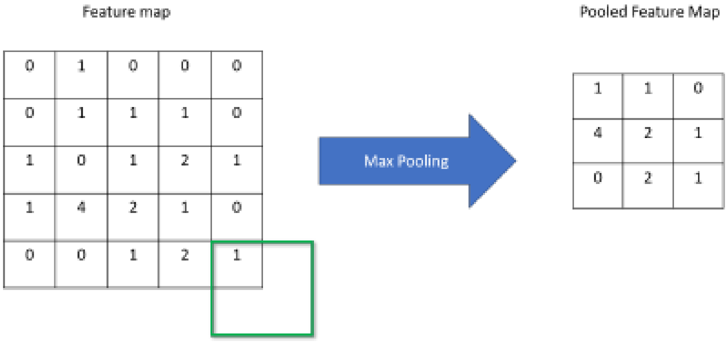
0	0	0	0	0	0	0
0	1	0	0	0	1	0
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	1	0	0	0	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Figure 2.5: Convolutional Layer.

We use many filters (featured detectors), and so we have the feature maps to keep different info on the image. Filters are learned during the training phase. The same filters in every position. Before moving to block 2 we apply the rectifier because we want to increase the non-linearity in our image because the images are not linear (objects and backgrounds), the rectifier eliminates all values under 0 and transforms the image again, there are also other applicable functions like the sigmoid or the leakly relu (Fig.2.5). The most used non-linear activation function is the Rectified Linear Unit (ReLU), and it is defined as follow:

$$y_{ijk} = \max \ \{0 \ , \quad x_{ijk}\} \tag{2.8}$$

2. **Max Pooling**



0	1	0	0	0
0	1	1	1	0
1	0	1	2	1
1	4	2	1	0
0	0	1	2	1

1	1	0
4	2	1
0	2	1

Figure 2.6: Max pooling.

The second block, Max Pooling, is shown in the Fig.2.6 where the maximum value is taken in the green box. The most common way to perform pooling is to apply a max function called max-pooling to the output of each filter. Max pooling is defined as follow:

$$y_{ijk} = max\{y_{i^{'}j^{'}k} : i \leq i^{'} < i + p, j \leq j^{'} < j + p\} \tag{2.9}$$

Because the larger numbers on the feature map represent where the similarity of the characteristics actually exists. We are eliminating 75% of the information, but if we analyse the same image with this system but rotate between them, we have the same maximum pool, since the value “4” would be close in another cell of the matrix; therefore, we preserve the characteristics on the same images even if distorted/rotated (Fig.2.7).

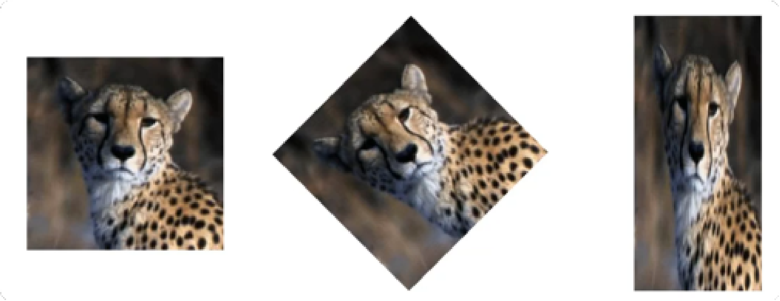
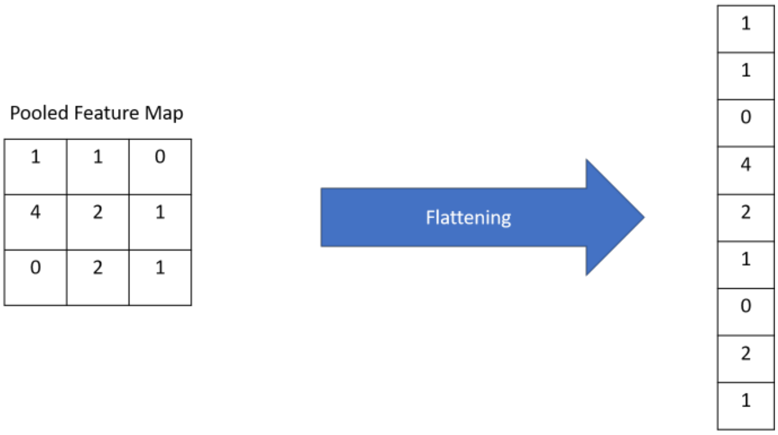


Figure 2.7: Example to images even if rotated or distorted.

By removing the information in this step, we avoid over-adaptation, reduce noise.

3. **Flattening**

   The third block, Flatting, is shown in the Fig.2.8 where Pooled Feature Map(i,j) can be flatten to 1D array b(k).



Pooled Feature Map

1	1	0
4	2	1
0	2	1

1
1
0
4
2
1
0
2
1

Figure 2.8: Example to Flattening.

We now have as many vectors as the images created with filters as shown in the Fig.2.9.

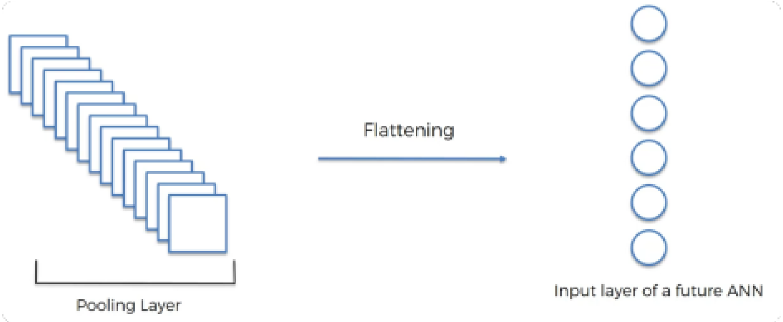


Figure 2.9: Flattening.

4. **Full Connection**

   We now have our classic artificial neural network (Fig.2.10), introduced in subsection 2.1.2, connected in a cascade to block 3 where each neuron is connected. Fully connected layer, they are all connected. The purpose of the ANN is to combine our features into attributes that even better predict the things we are trying to classify.

   One technique for reducing overfitting in the fully connected layer is dropout [65]. Dropout is a powerful regularization method, which has shown benefits for large neural networks. The simple key concept of dropout is to reduce co-adaptation between units. This objective is obtained randomly dropping units and their connections during the training phase.

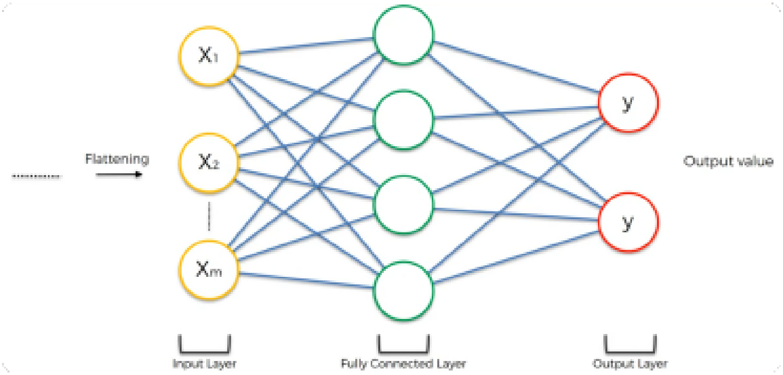


Figure 2.10: Artificial neural network.

The V-Net describe in section 3.1.2 is represented by a first part called compression which is composed of the blocks 1,2,3 just seen, and connected to a decompression part which reconstruct the labelled image.

2.2 Accuracy evaluation

In this section, is shown which metrics have been used to evaluate the result of the experiments conducted. It was considering the classification problems with two class labels, positive and negative $\{1, 0\}$. Some classification models produce continuous output (e.g. an estimate of the probability of belonging to a class of instances) to which different thresholds can be applied to predict class membership. Other models produce a discrete class label indicating only the expected class of the instance. To distinguish between the actual class and the expected class we use the labels $\{1^*, 0^*\}$ for the class predictions produced by a model. Given a classifier and an instance, there are four possible outcomes. If the instance is positive and classified as positive, it is considered as true positive; if it is classified as negative, it is counted as a false negative. If the instance is negative and classified as negative, it is considered as true negative; if it is classified as positive, it is counted as a false positive. Given a classifier and an instance set (the test set), it is possible to build a matrix two by two that represents the dispositions of the set of instances. This matrix forms the basis for many common metrics and is called the confusion matrix (Tab.2.1). The numbers along the major diagonal represent the correct decisions made, while the numbers on this diagonal represent the errors between the various classes.

		True class	
		1	0
Hypothesized class	1*	True Positive	False Positive
	0*	False Negative	True Negative

Table 2.1: Confusion matrix.

These metrics are calculated for single classes to evaluation the classification:

- Precision answers to the question, “What proportion of identifications was actually correct?”:
$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{2.10}$$

- Recall answers to, “What proportion of identifications was identified correctly?”:
$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{2.11}$$

- The F1 score or Dice index, is the harmonic average of the precision and recall:
$$F1 = 2\ \cdot\ \frac{precision\ \cdot\ recall}{precision + recall} \tag{2.12}$$

If F1 score is high, the model has good results in precision and recall; on the contrary, if it is low, it indicates that the model has problems with false positives or false negatives. F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

- False positive rate answers to, "What proportion of identifications was a false alarm?":

$$False\ positive\ rate \ = \ \frac{false\ positives}{true\ negative + false\ positives} \tag{2.13}$$

- Overall accuracy calculates how many cases have been correctly classified:

$$accuracy = \frac{true\ positives + true\ negative}{true\ positives + false\ positives + true\ negative + false\ negative} \tag{2.14}$$

- AUC answers to, "How much model is capable of distinguishing between classes?" Finally, AUC is a common method to calculate the area under the ROC curve in a graph [21]. ROC graphs are two-dimensional graphs in which true positive rate (or recall) is plotted on the y-axis and false positive rate is plotted on the x-axis (Fig. 2.11). An ROC graph depicts relative tradeoffs between benefits (true positives) and costs (false positives).

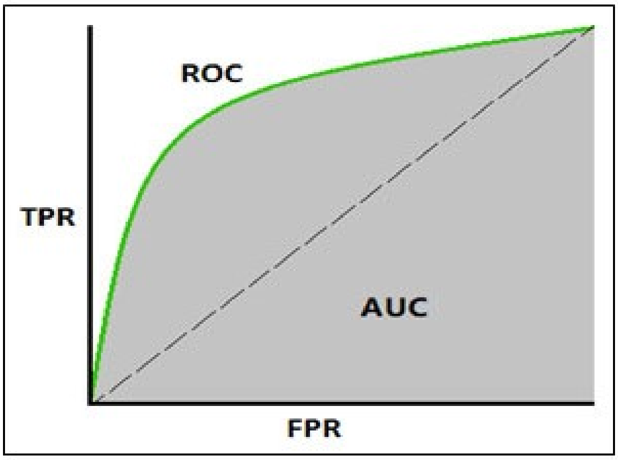


Figure 2.11: Roc curve and AUC.

2.3 Image processing techniques

In the following section, two techniques for image processing, the Hough Transform and the application of morphological operators are described. The first technique answers the fundamental problem in the field of image processing is the recognition of objects, the identification of geometric shapes presents in digital images. In the most straightforward cases, the problem boils down to figures containing a certain number of discrete points that stand on a white background. In general, the identification of shapes is a difficult problem because objects undergo translations, vary in colour, undergo changes in scale, rotations and images can be affected by noise. The problem can be solved for any desired degree of accuracy by testing the straight lines passing through each pair of points. According to this procedure, if n points are available, the number of operations required is proportional to n^2, and for n large the computational complexity could become unmanageable for practical applications. While the second technique, concerning morphological operators, is used to thin or thicken the linear elements of the object, thanks to the application of filters to make the opening with a disc structuring element that smoothes the corners from the inside and the closure with a disc that smoothes the edges from the outside.

2.3.1 The Hough Transform

Rosenfeld described an ingenious method, due to Hough, which consists in replacing the original problem of finding aligned points with a mathematically similar problem which includes in identifying straight lines. This method contains mapping each point in the image to a space in the parameter space. The parameter space is defined by a parametric representation described the straight lines in the image plane. Intuitively a model (template) is used, which scrolls over the image and compares it with the image below where an error function is minimized:

$$E(y,x) = \sum_{i,j} \left(I\left(y+i,x+j\right) - T\left(i,j\right)\right)^2 \tag{2.15}$$

$$E(y,x) = \sum_{i,j} |(I(y \ +i, x+ \ j) - T(i, \ j))| \tag{2.16}$$

In general, it is a difficult problem because objects undergo translations, changes in colour, experience changes in scale and rotations.

Hough's transformation is a technique that allows you to identify straight lines, curves or predefined shapes present within an image starting from their point projection in a parameter space that is defined as “Hough space” . Hough introduced this technique in 1962, and then taken up and modified by Duda and Hart, who improved the method and abroad the determination to other geometric shapes. Among the advantages present, we must mention its strength and stability against images with the presence of noise. Furthermore, it is an extended technique used in the field of digital image processing, and also the use in the medical field for the detection of forms in medical images through its generalized way for circumferences and ellipses.

The Hough Line Transform

Classical equation of the line is $y = ax + b$, for each point in the Image Space (IS) corresponds to a hypersurface (generalized surface) in the Parameter Space (PS) (Fig.2.12).

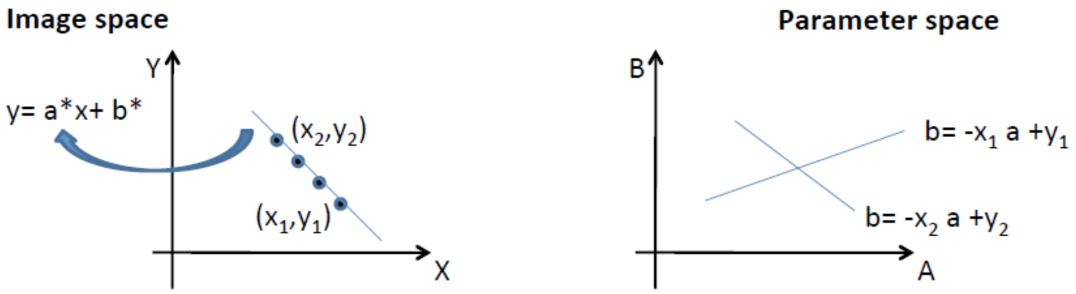


Figure 2.12: Image space and corresponding parameter space.

Fixed a point (x_i, y_i) in the image space the equation $b = y_i - ax_i$ describes the curve (which still remains a straight line). For example: for the point (x_1, y_1) along the line $y_1 = ax_1 + b$, in the parameter space is $b = y_1 – ax_1$.

The n points in the IS belonging to the same curve generate n surfaces that intersect at the same point in PS (Fig.2.13).

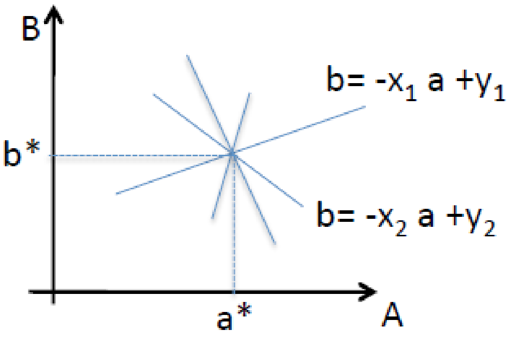


Figure 2.13: Parameter space.

So, in the PS an intersection of many surfaces is the indication of the presence of a particular instance of the analytical curve sought. In general, it will take several points at least equal to the number of parameters to identify a curve. The transform, therefore, sends to convert a curve search problem into the simpler one of intersection search.

Phases of the algorithm:

1. The discretisation of the parameter space (Fig.2.14) in NxM cells:

$$a_i = a_0 + i\delta a \quad where \quad i = 1, \ldots, N \tag{2.17}$$

$$b_j = b_0 + i\delta b \quad where \quad i = 1, \ldots, M \tag{2.18}$$

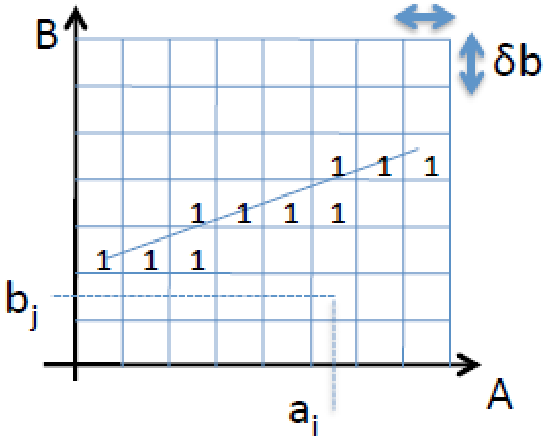


Figure 2.14: Discretisation of the parameter space.

2. Definition of the accumulation matrix: $H = H_{MxN} = (H_{ji}) \quad with \quad H_{ji} = 0 \forall i, j$.

3. Projection of the point (x_P, y_P) in the straight-line $b = -x_P a + y_P$ and look for the cells crossed by the line.

4. Update the entries of the instruction accumulator matrix to the cells to cross from the line where:

$$H_{ji} = H_{ji} + 1 \tag{2.19}$$

   if the cell is crossed by the line.

5. Repeat for each point of the image, so as to calculate the accumulation matrix (Fig.2.15).

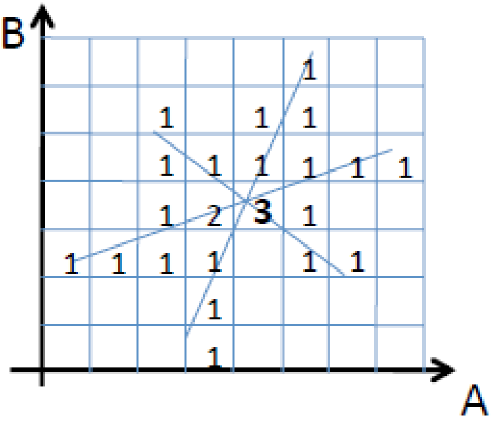


Figure 2.15: Accumulation matrix in parameter space.

6. The local maxima in the accumulator matrix identify the intersection points of the projected straight lines.

7. The local maxima in the accumulator matrix identify the collinear points in the input image.

8. If j^* and i^* is the pair of indices corresponding to the global maximum in the accumulator matrix, then:

$$y = a_{i^*}x + b_{j^*} \tag{2.20}$$

it is the straight line in the image space that we are looking for.

2.3.2 Morphological Operations Opening and Closing

In the field of image processing, the term mathematical morphology indicates the study of the geometric structure of images and morphological operators carry out elaborations on the shape of an object [20]. The fundamental morphological operations are dilation and erosion; the first consists of adding pixels to the outlines of an object, while the second is removing them. In both cases, the number of pixels added or removed depends on the organized shape on which it operates and on the size of the structuring element, i.e. a smaller reference shape on which you interact, used to make a comparison with the forms and structures of the image. The erosion of a binary image f by a structuring element s (denoted $f \ominus s$) produces a new binary image $g = f \ominus s$ with ones in all locations (x, y) of a structuring element's origin at which that structuring element s fits the input image f, i.e. $g(x, y) = 1$ is s fits f and 0 otherwise, repeating for all pixel coordinates (x, y).

The dilation of an image f by a structuring element s (denoted $f \oplus\ s$) produces a new binary image $g = f \oplus s$with ones in all locations (x,y) of a structuring element's origin at which that structuring element s hits the input image f, i.e. $g(x, y) = 1$ if s hits f and 0 otherwise, repeating for all pixel coordinates (x, y). Dilation has the opposite effect to erosion; it adds a layer of pixels to both the inner and outer boundaries of regions. The expansion and erosion processes are operations at the basis of the two operators that have been taken into consideration in the experimental tests, that is, the morphological operators of opening and closing. The morphological opening operators are characterized by erosion followed by dilatation, and the effect they have is that of preserving as much as possible regions of a shape similar to the structural element and with distinct differences. The opening of an image f by a structuring element s (denoted by $f \circ s$) is an erosion followed by a dilation:

$$f \circ s = (f \ominus s) \oplus\ s \tag{2.21}$$

On the contrary, the closing morphological operators are characterized by an expansion followed by erosion and have the task of closing any holes inside the segmentation. The closing of an image f by a structuring element s (denoted by $f \cdot s$) is a dilation followed by an erosion:

$$f \cdot s = (f \oplus\ s) \ominus s \tag{2.22}$$

For all the experiments used, a "disk" spherical shape is used as a structuring element, since it is the figure with the most characteristics in common with the form of a tumour.

3
MRI segmentation in Metastasis Tumors Treatment

The solutions investigated in this task are an extension of those presented in a previous work [28] from which we inherit the idea of using segmentation procedures based on SVM. In the present work, experiments have been extended by using an enlarged dataset and by developing and evaluating a second segmentation procedure based on deep learning approach [29].

The first objective of this study was to investigate whether BM segmentation may be approached successfully by two supervised ML classifiers belonging to feature-based and deep learning approaches, respectively. SVM and V-Net Convolutional Neural Network model [53] are selected from the literature as representative of the two approaches. In the experiments, we consider several configurations of the two methods to segment brain metastases on contrast-enhanced T1-weighted magnetic resonance images. Performances were evaluated and compared under critical conditions imposed by the clinical radiotherapy domain, using in-house dataset and public dataset created for the Multimodal Brain Tumour Image Segmentation (BraTS) challenge. Our results showed that the feature-based and the deep network approaches are promising for the segmentation of MRI brain metastases achieving both an acceptable level of performance. Experimental results also highlight different behaviour between the two methods. SVM improves performance with a smaller training set, but it is unable to manage a high level of heterogeneity in the data and requires post-processing refinement stages. The V-Net model shows good performances when trained on multiple heterogeneous cases but requires data augmentations and transfer learning procedures to optimise its behaviour.

A second objective of the present work was the design of a software package implementing an integrated set of procedures for active support in BM segmentation within the radiotherapy workflow. It was designed to be user-friendly, with a right trade-off between automation and interactivity.

3.1 Interactive BM segmentation based on supervised learning models

In line with the above considerations, the present work is aimed to design an interactive system for active support in the delineation of BM in clinical RT studies. It is designed to operate in a clinical setting to reduce the workload of health-care professionals but leaving them full control of the process. It is then conceived semi-automatic but requiring limited user interaction in an attempt to facilitate the insertion in current clinical practice.

The system implements a segmentation procedure hierarchically structured in three phases:

- Volume-of-interest (VoI) specification.
- Automated segmentation by supervised learning models.
- Segmentation refinement.

As regards the second phase, we investigated whether automated segmentation may be approached successfully by conventional, feature-based machine learning and deep learning models exploiting mutual advantages, limitations and synergies among them. We mainly studied the potential of the two methods in optimising the balance between accuracy and demand for training data. The following subsections describe the three phases of the proposed segmentation procedure.

3.1.1 VoI specification

Conceptually BM segmentation task includes both detection and delineation. Automated procedures cope with one or both of the two sub-tasks usually. In our context, the proposed interactive segmentation procedure limits the role of the automated solutions to the delineation task, lying to users the manual detection of BM in the T1c imagery. The advantage of this strategy is a simplification of the automated segmentation task for the learning model selected. In this preliminary step, a user specifies a volume of interest (VOI) by drawing a rectangular region on one slice of the input volume and selecting first and last slices in such a way that the entire pathological area is bounded within the specified parallelepiped (see Fig. 3.1, 3.2).

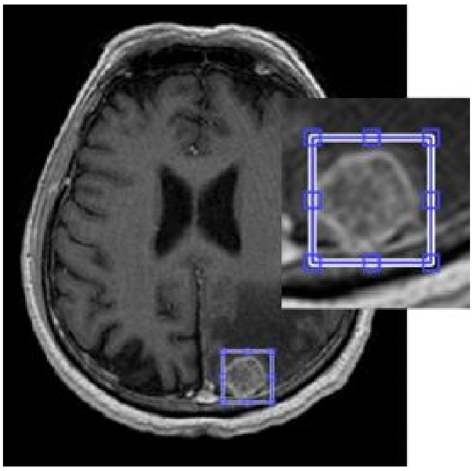


Figure 3.1: Show external source slices of T1c MR scans with the corresponding VoI slice perimeter bounding BM areas.

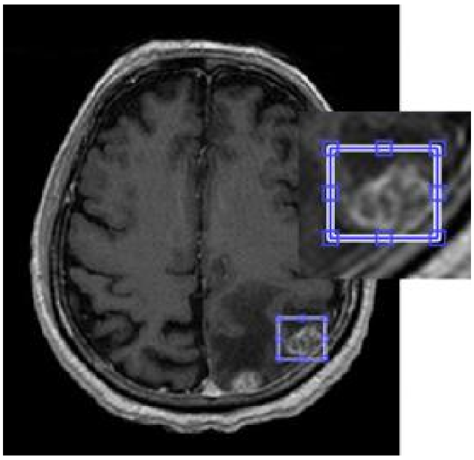


Figure 3.2: Show medium slices of T1c MR scans with the corresponding VoI slice perimeter bounding BM areas.

3.1.2 Supervised classification of pathological and healthy tissues

In the second stage of the segmentation procedure, a supervised machine learning model is applied to the selected sub-image. The supervised learning task is aimed to perform a hard binary categorisation, labelling voxels within the selected volume as Brain Metastasis (BM) and Healthy tissue (H). During the training phase, the classifier learns an approximation for the true input-output relationship based on a given training set of examples constituted by N input-output pairs $\{(x_i, y_i)\}$, i = 1, ..., N where x_i $y_i \in \{BM, H\}$ is a supervised label denoting the membership in the metastasis or healthy class.
We conceived and conducted two sets of experiments. The first used a conventional supervised classifier built on the top of a hand-crafted feature extraction procedure. In our previous works, we deal with MRI brain tumour segmentation using several methods selected from state-of-the-art classifiers in the field of MRI segmentation. In particular, we investigated the use of Fuzzy connectedness and Graph Cut for glial tumour segmentation [58] and SVM for meningioma and edema segmentation [8]. Fuzzy Connectedness and Graph Cut methods are interactive, asking experts to provide accurate initialisation information for each image processed. Results obtained by these methods were accurate but strongly influenced by the prior knowledge provided by the users or by ancillary methods. In the RT domain, where a large number of images are needed to be handled, they can be laborious and time-consuming. We have shown that trained SVM allows complete delineation of meningioma and edema tissues and accurate volume estimation by processing both volumetric and non-volumetric imagery in a few minutes, without

requiring manual selection of example voxels or seeds. Performances obtained were good, confirming the results obtained in other studies [6]. Approaches based on Random Forest show similar characteristic and can also be suitable in the RT domain. Bauer et al. in their review [6] investigate the behaviour of SVM and Random Forest segmentation. The two methods showed comparable performances. It is worth to note, however, that evaluations were performed on different datasets and different evaluation metrics, making difficult the comparison. Statnikov et al. [66] found that, both on average and in the majority of data used in the study, Random Forests exhibit larger classification error than SVM when processing microarray datasets. In the present work, the choice fell on SVM, not because the best that can be made but as it has been extensively used in MRI image segmentation and can be considered representative of the conventional learning approaches.
The second experiment was conducted using the V-Net convolutional neural network model [53] selected among the deep learning models oriented to MRI volumetric images [46] and properly adapted for our application context. To make the work self-contained, we briefly outline the basic concepts of both learning models investigated.

BM delineation using SVM

SVM is a classification algorithm based on kernel methods [73, 64] mapping the input patters into a high dimensional feature space. Classes which are non-linearly separable in the original space can be linearly separated in the higher dimensional feature space.

Let $\{(x_i, y_i)\}$ be a supervised training set of elements for a two-class classification problem, with $x_i \in X \subseteq R^n \quad and \quad y_i \in \{-1, 1\}$. Considering the case of linearly separable data, the solution to the classification problem consists of the construction of the decision function:

$$f\left(x\right) = sgn\left(g\left(x\right)\right) with \tag{3.1}$$

$$g\left(x\right) = w^t x + b \tag{3.2}$$

that can correctly classify an input pattern x not necessarily belonging to the training set.

SVM classifier defines the hyperplane that causes the largest separation between the decision function values for the “borderline” examples from the two classes. This hyperplane can be found by minimising the cost function, as follow:

$$J\left(W\right) = \frac{1}{2}||W||^2 \quad subject \quad to \tag{3.3}$$

$$W^T X_i + b \geq +1 \quad for \quad y_i = +1 \quad or \tag{3.4}$$

$$W^T X_i + b \leq -1 \quad for \quad y_i = -1 \tag{3.5}$$

The extension to the nonlinear classification is based on the function $g' = W^T \varphi\left(X\right) + b$ in which the non-linear operator $\varphi\left(.\right)$ hyperparameters.

In this case, the SVM cost function to be minimized is

$$J(W, \xi) = \frac{1}{2}||W||^2 + C\sum_{i=1}^{l} \xi_i \tag{3.6}$$

$$y_i \left(w^t \varphi\left(X_i\right) + b\right) \geq +1 - \xi_i \quad with \quad \xi_i \geq 0,\ \mathrm{i} = 1,\ 2,\ \ldots \mathrm{l} \tag{3.7}$$

Suykens [19] proposed a new formulation of SVM by adding an LS term in the original formulation of the cost function. This modification significantly reduces computational complexity.

The trained SVM classifier receives in input multidimensional patterns, in the form of vectors of measured features and assigns labels to corresponding T1c MR elements. Multidimensional input patterns are composed of T1c voxel intensities and corresponding textural and contextual features extracted from the MR scan. The literature proposes different sets of features related to the supervised classification of MRI data. Features are selected in function of the MRI channels used, and the classifiers adopted [31, 6]. Based on our previous works, in addition to image intensities from the T1c MR scan, we consider features describing neighbour relationships and texture [72]. The feature set adopted is described in the next section. As detailed in Section 3.2, several strategies were conceived to conduct the learning stage. Results obtained were analysed, and the resulting optimal configuration has been used in the comparison analysis.

BM delineation using V-Net

The V-Net model is a Fully Convolutional Neural Network proposed by Milletari et al. [53] for volumetric medical image segmentation. The name V-Net comes from the fact that the network can be drawn with the symmetric shape like the letter V. Salient aspects of the model proposed are the use of volumetric convolution to overcome the slice-by-slice processing of input image volumes and the use of a novel objective function based on Dice coefficient maximisation, optimised during training. Both these aspects are significant for our segmentation task, which is based on volumetric data characterised by a strong imbalance between the number of voxels belonging to the pathological area and background.

In our experiment, we adopt the V-Net configuration proposed for Lung Tumor Segmentation in [57] (see Fig. 3.3). The inputs of the network are T1c MRI subvolumes by dimension 64x64x64. The V-Net has an increasing number of convolutional filters (16, 32, 64, 128, 256) with a total of 116 layers and 128 connections. The output layer is a custom Dice loss layer [69].

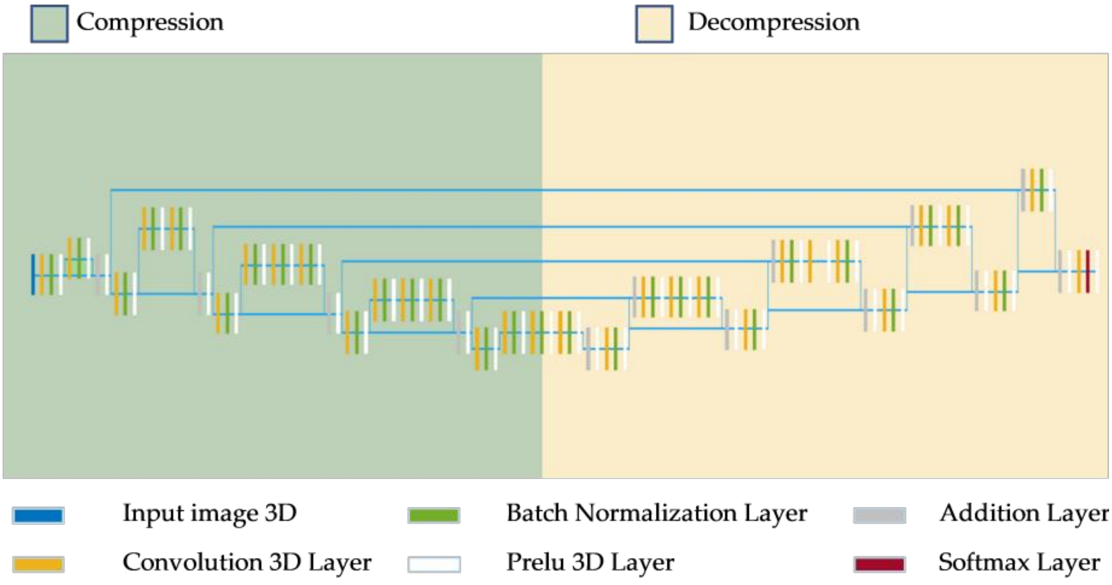


Figure 3.3: Screenshots of the system interfaces documenting the initial input/output phases.

Figure 3.3 shows a schematic representation of V-Net architecture adopted, which is composed of two parts implementing compression and decompression paths. In the left part, we found a deep residual learning strategy: all the outputs of the convolutional layers, after non-linearities processing, are added to the output of the last convolutional layer [34].

In each stage, the compression part of the network computes features two times higher in number than the one of the previous layers. In the decompression part, the network extracts features and expands the spatial support of the lower resolution feature maps. The last convolutional layer implements two outputs of the same size of the input image volume. A voxel-wise soft-max operation is applied, converting the two outputs as probabilistic segmentation of background and BM regions.

Results obtained in several segmentation studies show how necessary it is to use large-scale datasets for effective application of deep learning methods [2, 41]. Whereas in our context, as in many other biomedical domains, manually segmented reference volumes are not easy to obtain, our strategy includes data augmentation and transfer learning tasks to compensate for the limited dataset available and reaching the double goal of increasing robustness and generalisation capability of the network respectively, as described in 3.2.

3.1.3 Segmentation Refinement

In our study, we conceived automated segmentation as an intermediate task within a typical workflow in radiotherapy planning. The active decision support of the machine learning procedure reduces user-interaction that is limited to two phases: a preliminary "detection" phase with which to specify the VoI, as illustrated in Sec.3.1.1, and a post-processing phase aimed to refine the identified segments. Refinement is made necessary by commission and omission errors that are most likely to occur when lesions are of small size, and the tumour area presents inhomogeneity due to necrosis and active parts. Our strategy provides both for computer-aided manual editing as illustrated in Sec.3.3 and an automated procedure based on the use of Morphological Operators to refine the segmented masks in an attempt to reduce omission and commission errors and making the segmented tumour area more compact. For each selected slice, Opening and Closing Operators with spherical shape apply consequently. The Opening Operator removes from the binary input image all the connected components that have a lower number of pixels than a set value and outputs a new binary image. The Closing Operator closes holes present in the image and returns the closed binary image. We performed three different tests, using for all a disk-shaped structuring element, aimed to tune parameters values of the Morphological Operators and decide the order of application. In the first test, only the opening morphological operator (Open) was applied by varying the radius; in the second test, only the morphological closing operator (Close) was applied, varying the radius; in the third test, we applied both operators in a different sequence. The best result was obtained by applying first the opening morphological operator with a radius of 5 and then the closing operator with a radius equal to 10.

Figure 3.4 and Figure 3.5 illustrate the refinement of automated segmentation results performed manually by the user and by the Morphological Operators, respectively.

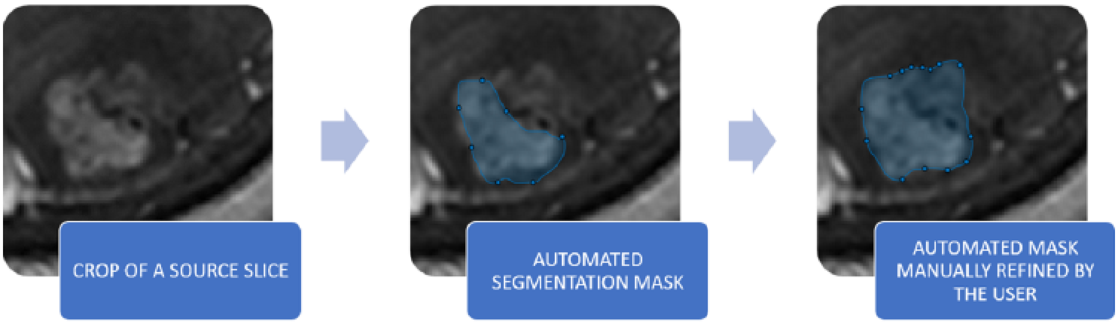

CROP OF A SOURCE SLICE
AUTOMATED SEGMENTATION MASK
AUTOMATED MASK MANUALLY REFINED BY THE USER

Figure 3.4: From left to right: crop of a source T1c MRI slice, superimposition of the automated segmentation mask, superimposition of the mask manually refined by the user.

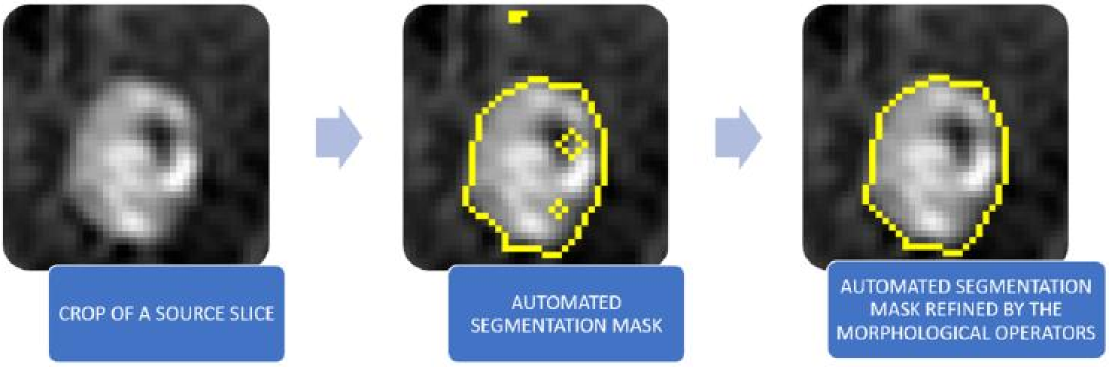

CROP OF A SOURCE SLICE
AUTOMATED SEGMENTATION MASK
AUTOMATED SEGMENTATION MASK REFINED BY THE MORPHOLOGICAL OPERATORS

Figure 3.5: From left to right: crop of a source T1c MRI slice, superimposition of the automated segmentation mask, superimposition of the mask automatically refined by Morphological Operators.

3.2 Experiments

We conceived and conducted several experiments to systematically investigate the behaviour and quantitatively assess performances of the selected machine learning models for automated support in the BM segmentation within the RT workflow (see Fig. 3.6). The experiments used two data sets: one in-house collected and one public, created for the Multimodal Brain Tumour Image Segmentation (BraTS) challenge for the detection of gliomas [52]. In-house data are used to train and test SVM classifier and to fine-tune and test V-Net network pre-trained using data from BraTS.

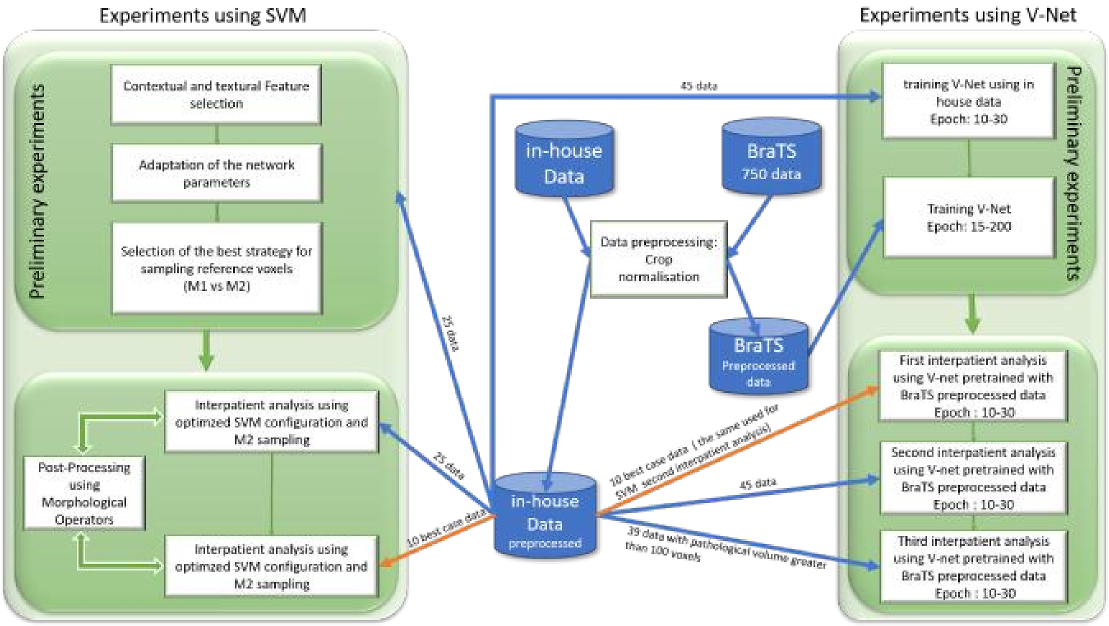

Experiments using SVM
Experiments using V-Net
Preliminary experiments
Contextual and textural Feature selection
Adaptation of the network parameters
Selection of the best strategy for sampling reference voxels (M1 vs M2)
Interpatient analysis using optimzed SVM configuration and M2 sampling
Post-Processing using Morphological Operators
Interpatient analysis using optimzed SVM configuration and M2 sampling
45 data
in-house Data
BraTS 750 data
Data preprocessing: Crop normalisation
BraTS Preprocessed data
25 data
25 data
10 best case data
in-house Data preprocessed
10 best case data (the same used for SVM second interpatient analysis)
45 data
39 data with pathological volume greater than 100 voxels
training V-Net using in house data Epoch: 10-30
Training V-Net Epoch: 15-200
Preliminary experiments
First interpatient analysis using V-net pretrained with BraTS preprocessed data Epoch : 10-30
Second interpatient analysis using V-net pretrained with BraTS preprocessed data Epoch : 10-30
Third interpatient analysis using V-net pretrained with BraTS preprocessed data Epoch : 10-30

Figure 3.6: Flowchart illustrating the experimental design.

3.2.1 In-house data acquisition

The acquired dataset is composed of 45 T1c volumetric MR scans. Volumes are acquired using a 3D sequence characterised by 0,9 mm isotropic voxels, the pixel spacing of 0,47 mm and the slice thickness of 2,67 mm. The tumour cases considered are heterogeneous in terms of shape, position and intensity level (see Fig.3.7).

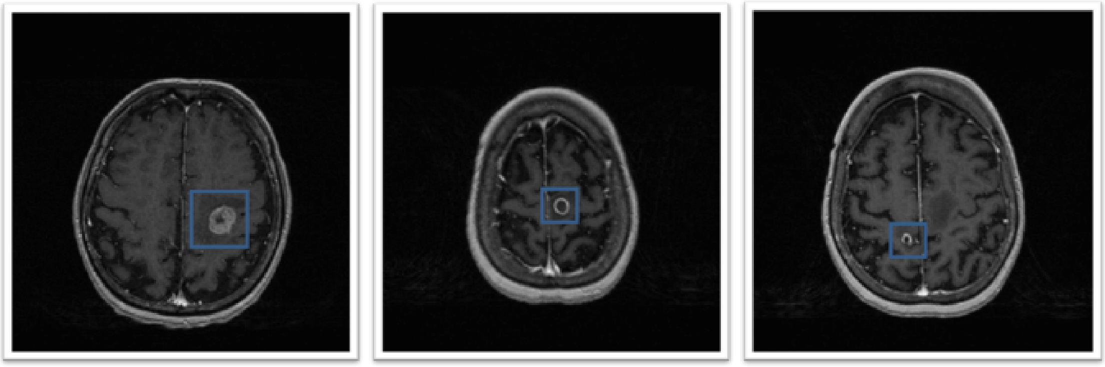


Figure 3.7: Show three source slices of T1c MR Volumes with the delimitation of the lesion area.

3.2.2 Evaluation Metrics

We assessed the accuracy of segmentation results by comparing the spatial distribution of the masks obtained by the automated segmentation with that of the masks obtained through a manual segmentation of the T1c images performed by radiologists. Metrics adopted for the accuracy is described below, according to Bouix et al. [69]. The minimal problem of assessing the agreement between two binary maps B1 and B2, representing reference and segmented data respectively, is obtained in terms of number of voxels at which both B1 and B2 score "1" (True Positive Tp) or "0" (True Negative Tn), the number of voxels at which B1 scores "0" and B2 scores "1" (False Positive Fp) and vice-versa (False Negative Fn). Several similarity indexes could be defined. We use Dice (DSC) [34], Precision and (P) and Recall (R) indexes (Olson and David, 2008) defined as follows:

$$DSC = \frac{2Tp}{(2Tp + Fn + Fp)} \tag{3.8}$$

$$P = \frac{Tp}{(Tp + Fp)} \tag{3.9}$$

$$R = \frac{Tp}{(Tp + Fn)} \tag{3.10}$$

The DSC index has been used broadly in the field of segmentation as a measure of spatial overlap, P and R indexes allow to measure under- and over-estimations [69].

3.2.3 Experiments with SVM

Performances of the SVM-based segmentation were evaluated by generating several segmentation procedures obtained by different SVM configurations and using different learning strategies. We used 25 reference data corresponding to those acquired and available in our clinical domain at the start of the study. We developed two preliminary tasks. First, we defined the optimal feature set that composes patterns in input to the SVM. For this task, the SVM has been configured as a soft-margin leas square (LS) model with a linear kernel. Contextual and textural features have been analysed systematically in order to determine the combination that is most appropriate for the current classification task. In particular, several configurations of the segmentation procedure have been experimented initially providing in input only intensity values of central voxel and neighbour voxels. Different neighbourhoods have been considered including incrementally neighbours along with voxel faces, corners and edges up to a maximum of 26 voxels. In a second step, an enlarged feature set has been considered adding textural features to the best neighbourhood configuration.

The following set of features has been finally selected (see Fig. 3.8):

- intensities from T1c scan
- first-order texture features: mean (μ), variance (σ), skewness (γ), kurtosis (β) and entropy (H)
- intensities in 26 neighbourhood voxels

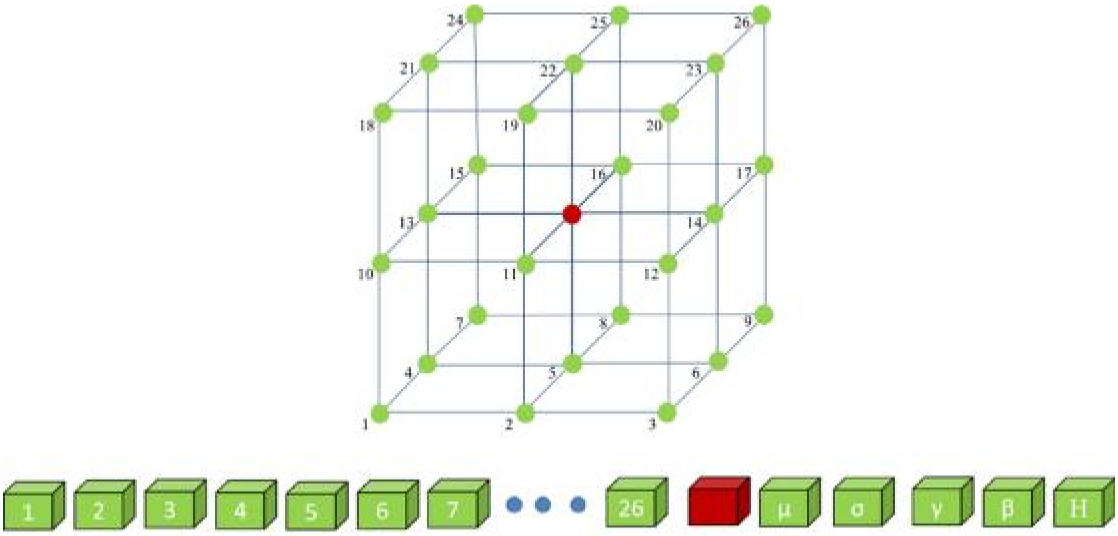


Figure 3.8: Input pattern vector composed of contextual and textural features.

Feature values have been normalised to have zero mean and unit variance.

The second step was to determine the value of the internal parameters of the model. Different types of kernels were tested, such as linear, quadratic, cubic, fine-medium-coarse gaussian. Given the results obtained, we confirmed the configuration of SVM as a soft-margin LS model with a linear kernel. With this optimised configuration, we conducted the first set of experiments by varying the modality for selecting training samples from the VoIs extracted from the available in-house dataset. In the first modality (M1) data are extracted from the reference masks selecting elements within the overall VoI under study. In the second modality (M2), the random selection was limited within a region built around the contour of the tumour reference masks. After trial and error procedures, the edge width was tuned to 8 pixels. These experiments have been conducted in intra-patient modality selecting training and test sets from the reference masks of the same VoI and built by randomly selecting elements in the proportion of 70% and 30% respectively. An equal number of elements labelled BM and H was extracted. In the M1 modality, before the random extraction of BM elements, contour elements of BM regions were reinserted. Table 3.1 shows the numerical results obtained in terms of mean values of DSC, P and R indexes over 25 cases under study. SVM, trained according to M2 strategy, slightly prevails with a mean DSC value equal to 0.878. P and R values highlight a significant reduction of omission and commission errors. These results can be explained by the fact that metastases have little extensions and a high level of heterogeneity occurs in the internal part of the pathology due to the presence of necrosis and/or active parts (see Fig.3.7). This high level of heterogeneity is difficult to learn during training and, as seen in our experimental context, SVM shows better behaviour when trained on relatively homogeneous regions.

Table 3.1: Dice (DSC), Precision (P), Recall (R) values obtained using M1 and M2 strategy and tested on the overall cases under study.

		M1	M2
DSC	Mean	0.808	0.878
	Var	0.008	0.003
	Min	0.549	0.757
	Max	0.908	0.963
P	Mean	0.824	0.884
	Var	0.006	0.003
	Min	0.648	0.749
	Max	0.927	0.963
R	Mean	0.796	0.873
	Var	0.012	0.003
	Min	0.476	0.764
	Max	0.923	0.963

Using the configuration described above and with the M2 voxel extraction strategy, we quantified segmentation performances by conducting a second set of experiments aimed to investigate the ability of the SVM to generalise under increasing levels of heterogeneity seen in training.

First, 25 SVM models are trained from one case and tested on all the cases under study. Table 3.2 shows the results obtained by the best configuration. We compute accuracy values obtained with and without the use of Morphological Operators to isolate their contribution within the overall segmentation procedure. Secondly, in order to measure the ability of the SVM to segment under an increased level of heterogeneity, we select the 10 cases where the SVM gave the best results in intra-patient analysis. The Series of 6 cases extracted from the 10 cases, previously selected are considered for training, for a total of 120 learning stages. Table 3.3 shows accuracy values obtained by the best SVM, with and without the use of Morphological Operators, tested on the overall 25 cases.

Table 3.2: Dice (DSC), Precision (P), Recall (R) values obtained by the segmentation procedure trained on one case according to M2 strategy, with and without the use of Morphological Operators (MO) and tested on the overall 25 cases under study.

		SVM	SVM+MO
DSC	Mean	0.701	0.693
	Var	0.011	0.035
	Min	0.462	0
	Max	0.844	0.897
P	Mean	0.747	0.696
	Var	0.026	0.047
	Min	0.437	0
	Max	0.997	0.997
R	Mean	0.737	0.769
	Var	0.035	0.057
	Min	0.41	0
	Max	0.983	0.990

Table 3.3: Dice (DSC), Precision (P), Recall (R) values obtained by the segmentation procedure with data from a set of VoIs belonging to different scans with and without the use of Morphological Operators (MO) and tested on the overall cases under study.

		SVM	SVM+MO
DSC	Mean	0.653	0.66
	Var	0.008	0.028
	Min	0.39	0
	Max	0.77	0.82
P	Mean	0.681	0.641
	Var	0.017	0.025
	Min	0.278	0
	Max	0.968	0.881
R	Mean	0.71	0.762
	Var	0.026	0.035
	Min	0.482	0
	Max	0.955	0.976

a	b
c	d

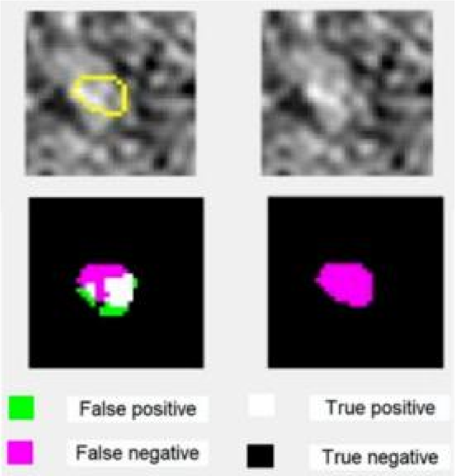


Figure 3.9: First row, from left to right: (a) crop of a source slice (Slice 1) of T1c MR Volume with superimposed the contour of metastasis reference mask (dimension: 83 elements), (b) slice of the corresponding VoI. Second row from left to right: (c) segmentation mask produced by SVM, (d) refinement by the Morphological Operators.

Looking at values in Table 3.2 and Table 3.3 in more detail, we notice that the application of Morphological Operators is not always advantageous. However, when studying individual cases, we have noticed that under-estimation and over-estimation errors occur systematically when the pathology occupies a very small volume (under the 100 elements), and it is inserted in a highly heterogeneous context. Figure 3.9 illustrates an example with a slice (Slice 1) including with a metastasis remarkable small. The refinement accomplished by the Morphological Operators deletes all the true positive elements identified by the SVM classifier. On the contrary, the segmentation masks of the larger pathological area in the slice (Slice 2) shown in Figure 3.10, indicate that the segmentation strategy benefits from the allied use of SVM and Morphological Operators. Table 3.4 lists the numerical results of the cases illustrated in Figure 3.9 and 3.10.

Table 3.4: Dice (DSC), Precision (P), Recall (R) values obtained by the segmentation procedure when processing slices in Figure 3.9 and 3.10.

		DSC	P	R
Slice 1	SVM	0.556	0.656	0.482
	SVM+MO	0	0	0
Slice 2	SVM	0.885	0.898	0.873
	SVM+MO	0.940	0.926	0.953

Automatic segmentations were evaluated qualitatively through visual inspection.

a	b
c	d

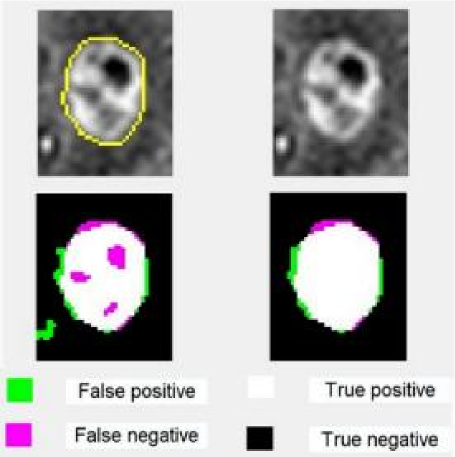


Figure 3.10: First row, from left to right: (a) crop of a source slice (Slice 2) of T1c MR Volume with superimposed the contour of metastasis reference mask (dimension: 644 elements), (b) slice of the corresponding VoI. Second row from left to right: (c) segmentation mask produced by SVM, (d) refinement by the Morphological Operators.

The complete strategy, including the combined use of SVM and Morphological Operators, have been judged satisfactory. The limitations of the segmentation procedure, inherent to specific cases, as illustrated above, are considered acceptable and manageable with interactive phases devoted to manual refinements of the automated results.

3.2.4 Experiments with V-Net

In these experiments, we consider an enlarged in-house data set composed of 45 collected MRI scans made available for this second phase of the study. These data are used to fine-tune the V-Net model after a pre-training procedure based on the use of BraTS data, including 750 4D annotated volumes, each representing a stack of 3D images. Each 4D volume has size 240x240x155x4, where the first three dimensions correspond to height, width, and depth of a 3D volumetric image. The fourth dimension corresponds to different scan modalities. For all the experiments developed, we pre-processed both in-house and public BraTS data before their use in training, according to the following strategy. To reduce the amount of data to be processed, we cropped original volumes to sub-regions containing a significant amount of brain and tumour tissues. Then, we normalise each data independently by subtracting the mean and dividing by the standard deviation of the cropped brain region; then, we removed the outliers and, finally, we applied a data resizing in the interval [0,1]. Data are augmented by randomly rotating and reflecting image patches to make training more robust. Finally, we randomly extract patches of 64x64x64 voxels sizes from the cropped data.

Initially, we conduct experiments aimed to assess the utility of transfer learning.

V-Net was trained directly using the preprocessed in-house collected data. We considered several learning strategies for training V-Net by varying the number of epochs with a range from 10 to 30 and splitting the data set into 82% for training, 6% for validation and 12% for test. We obtained poor, unacceptable results with accuracy close to zero.

We proceeded then, by pre-training V-Net using pre-processed T1c scans of BraTS data. During the experiment, the number of epochs varied from 15 to 200 and the preprocessed dataset was split again into 82% for training, 6% for validation and 12% for test. The DSC value obtained is equal to 0,53. Performances obtained would not be acceptable for the final segmentation task, but should be considered acceptable for pre-training.

The V-Net architecture, pre-trained on glioma segmentation task, was refined using in-house collected data with different learning strategies. Several configurations of V-Net-based procedure were considered by varying epoch.

We conceived three experiments addressing the following main questions:

- how did the V-Net compare with SVM when trained on the same data set
- how did performances could be optimised increasing the number of training data
- how did dimension of lesions influence accuracy of segmentation results.

To address the first question, we train and test V-Net with the same strategy used in the second interpatient analysis conducted with SVM (see section 3.2.3). Results obtained are shown in Table 3.5. Comparing these results with those obtained by the SVM, shown in Table 3.3, we found that SVM segmentation prevails.

Table 3.5: Dice (DSC), Precision (P), Recall (R) values obtained by V-Net segmentation trained according to the second inter-patient analysis conducted for SVM.

	P	R	D
Mean	0,63	0,46	0,51
Var	0,17	0,11	0,13
Min	0,56	0,39	0,46
Max	0,68	0,55	0,59

To address the second question, we used an enlarged dataset composed of 45 T1c volumes, of which 17 for test and the remaining 28 used in 4-fold cross-validation. Results obtained were still poor with accuracy values under the value of 0,50 for all the indexes used. Moving deeper into the experiment conducted and analysing case-by-case performances obtained, we found that lower accuracy values occur when the pathology occupies a very small volume (under the 100 elements), and it has a highly heterogeneous context. V-Net has difficulties in identifying a very small lesion, as illustrated in figure 3.11, where the segmented mask has a spatial distribution quite different from the reference mask.

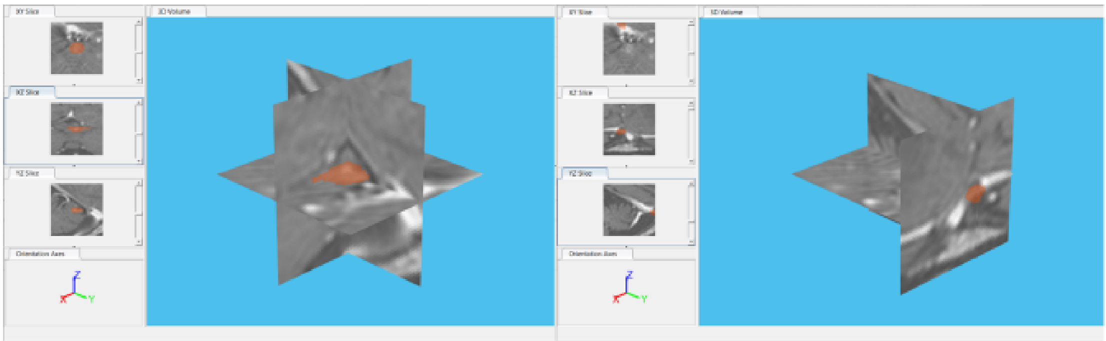


Figure 3.11: 3D visualisation of reference mask (left) and V-Net mask superimposed on a very small lesion in MRI scan under study.

We conducted a third experiment by using a reduced set of data obtained by the initial in-house data set, by eliminating cases in which pathology occupies volumes under 100 voxels. The final dataset is composed of 39 T1c volumes of which 12 for test and the remaining 27 used in 3-fold cross-validation. Results obtained are shown in Table 6.

Table 3.6: Dice (DSC), Precision (P), Recall (R) values obtained by V-Net segmentation trained on the refinement dataset.

	P	R	D
Mean	0,857	0,537	0,641
Var	0,021	0,017	0,010
Min	0,523	0,298	0,455
Max	1,000	0,706	0,802

Comparing performances obtained by the SVM configuration optimised for inter-patient analysis, shown in Table 3.2, we noticed that performances of the two classifiers are in this case comparable.

Analysing in detail, the segmentations performed we notice that the masks produced by V-Net have a higher level of spatial consistency than those produced by SVM, making the use of MO ineffective. As an example, Figure 3.12 shows results produced by SVM e V-Net highlighting differences in the computed segmentation masks.

As regard hyperparameters and execution time, the pre-training task takes about

150 hours on an NVIDIA™ 1080Ti with 11 GB of RAM, and the training with the in-house data takes about 7 hours. In training we used the stochastic Adam optimizer, the shuffle in every epoch, InitialLearnRate = 1e-3, LearnRateSchedule = piecewise, LearnRateDropPeriod = 5, RateDropFactor = 0.97.

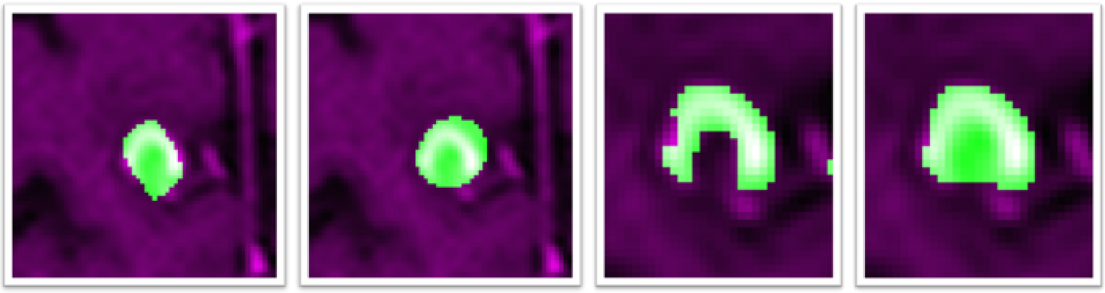


Figure 3.12: Crop of the source slice of T1c MR Volume with superimposed reference mask (a), segmentation mask produced by V-Net (b), segmentation mask produced by SVM without (c) and with MO (d). The source slice is the last image on the right shown in Figure 3.7 (c).

3.3 The Software System: Graphical User Interface (GUI) and sample program runs

The above-illustrated computational procedures are integrated into a unified framework to support the segmentation of BM within the RT workflow (see Figure 3.13).

Both the SVM-based and deep learning-based procedures have been implemented in different system configurations to extend the evaluation of the automated results obtained through an accurate field test with novel clinical subjects.

The software design started with the collection and analysis of requirements to outline the user model and operational conditions. We assessed cognitive and perceptual processes, attitudes and limitations involved in visual inspection and analysis tasks. Operation requirements concerning hardware facilities and input and output devices are inherited from protocols in use in neuroradiology domain. The multi-window system was implemented in MATLAB (MathWorks®).

Figure 3.14 shows the system interfaces documenting the initial phases of the session and input/output facilities. Visualisation tools are available for a reliable inspection of the selected MRI scan. By default, the three orthogonal planes of the corresponding central slice are simultaneously visualised. Several colour maps can be easily selected to modify the appearance of the axial slice.

On the left part of the GUI we found options made available by the system to develop segmentation task. The user selects the range of slices that must be processed

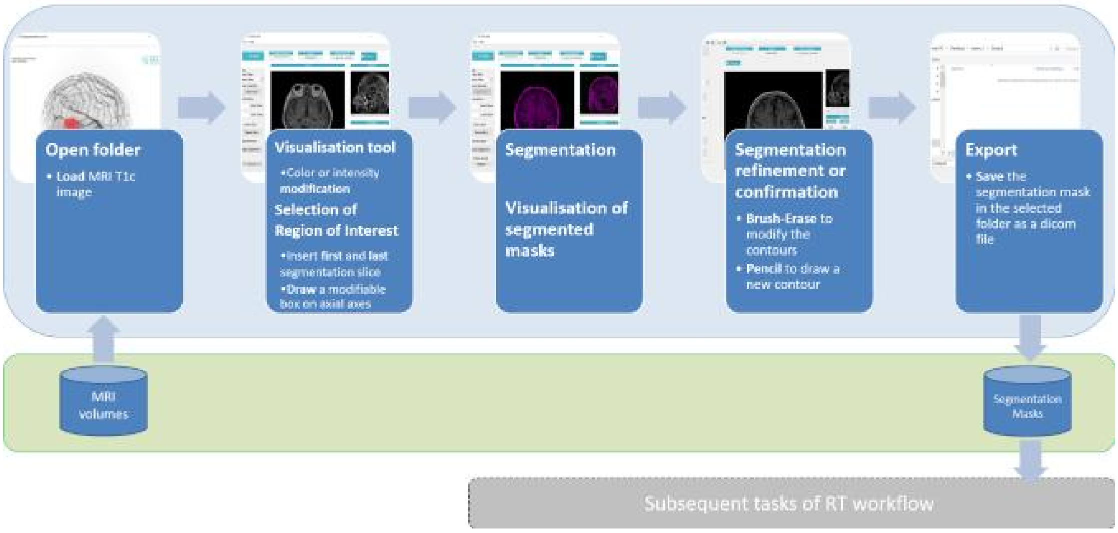
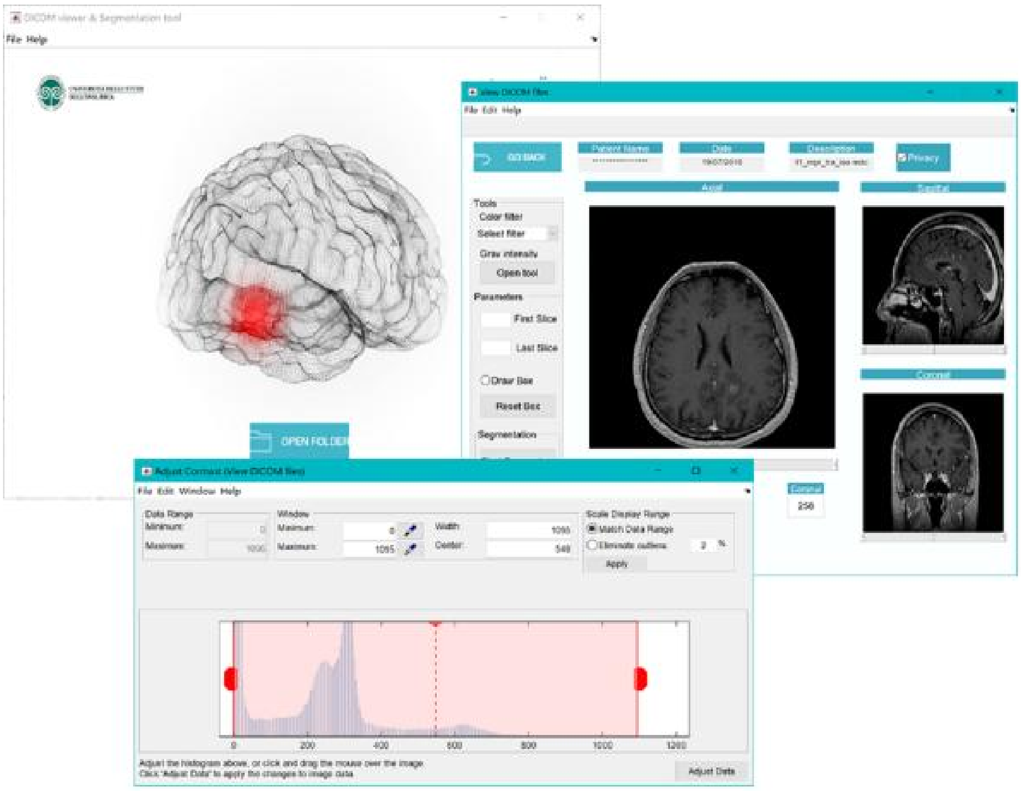

Open folder
• Load MRI T1c image
Visualisation tool
•Color or intensity modification
Selection of Region of Interest
•Insert first and last segmentation slice
•Draw a modifiable box on axial axes
Segmentation
Visualisation of segmented masks
Segmentation refinement or confirmation
• Brush-Erase to modify the contours
• Pencil to draw a new contour
Export
• Save the segmentation mask in the selected folder as a dicom file
MRI volumes
Segmentation Masks
Subsequent tasks of RT workflow

Figure 3.13: The workflow of the functional structure of the software system implementing the segmentation procedure.

by inserting the numbers of first and last slices and proceed in the VoI specification (check on "Drow Box") by using zooming facilities if the case.

Segmentation procedure is run, and the contour of mask obtained is superimposed on the selected slices in all the projections as illustrated in Figure 3.15.



Date: 19/07/2018

Description: t1_mpr_tra_iso reb

Coronal: 256

Data Range Minimum: 0 Maximum: 1095

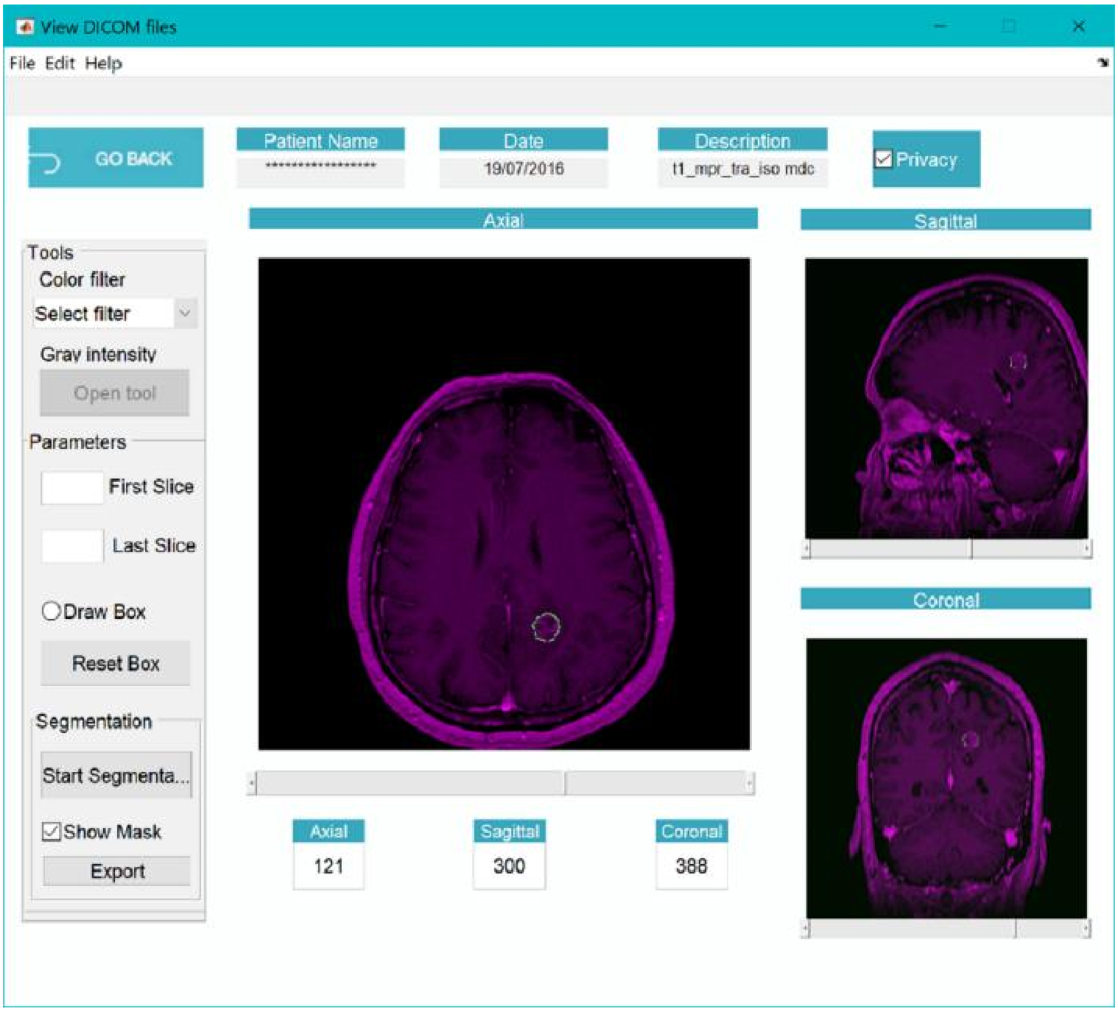
Window Minimum: 0 Maximum: 1095 Width: 1095 Center: 548

● Match Data Range
○ Eliminate outliers 2 %

Adjust the histogram above, or click and drag the mouse over the image
Click "Adjust Data" to apply the changes to image data

Figure 3.14: Screenshots of the system interfaces documenting the initial input/output phases.

The system allows the user to export the segmentation mask, to refine the segmentation results, to proceed in a further segmentation in the same volume or to return to the initial phase by unchecking "show Mask" command.



View DICOM files

Patient Name: *****************

Date: 19/07/2016

Description: t1_mpr_tra_iso mdc

☑ Privacy

Axial: 121

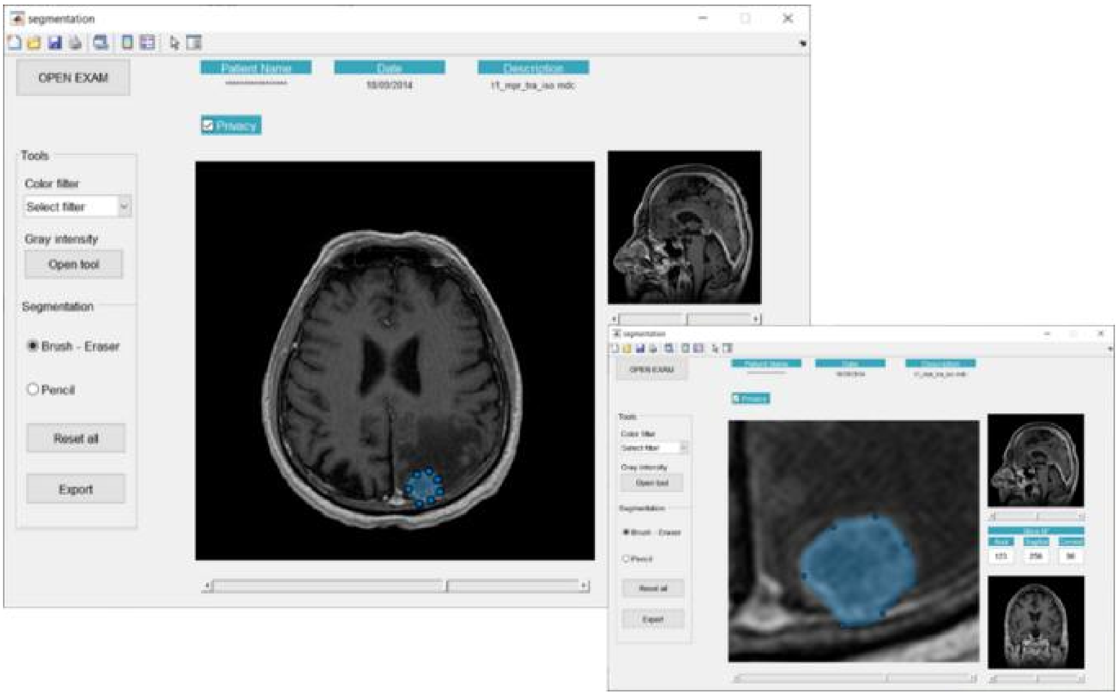
Sagittal: 300

Coronal: 388

Figure 3.15: Screenshot of the system interfaces documenting the visualisation of the segmentation masks computed by SVM.

Figure 3.16 shows options made available by the system to validate the automated segmentation. The segmented mask is shown superimposed on corresponding T1c source slice. The user is allowed to modify the contour by checking on “Brush-Eraser” command or delineating the entire segment in case the automated result occupies a wrong position (check Pencil).

The software is preliminary installed at the radiotherapy unit of the Hospital ASST-settelaghi, Varese (Italy) for a field test and then made available online with links in our institutional website.



Patient Name: ***************
Date: 18/03/2014
Description: t1_mpr_tra_iso mdc
☑ Privacy

Figure 3.16: Screenshot of the system interfaces documenting the visualisation of the segmentation masks computed by SVM.

3.4 Discussion and conclusions

As seen in our experimental context, brain metastases can be semi-automatically segmented using automated processing of T1c scans of MR images. The main goal of the proposed procedure was to facilitate the segmentation task within the radiation therapy workflow, actively supporting radiologist in the delineation of lesions. Our results showed that both a feature-based and a deep network approach are promising for the segmentation of MRI brain metastases achieving both an acceptable, although a not very good level of performance. Nevertheless, it is worth to note that SVM and V-Net achieved best results under quite different conditions. In the inter-patient analysis, SVM appears to be unable to manage a high level of heterogeneity in training, producing best results when trained only on individual cases, plus omitting heterogeneous voxels belonging to necrosis within the pathological area. On the contrary, V-Net model achieved an acceptable level of accuracy when trained on multiple cases and selecting voxels from the entire pathological area. Moreover, the two optimised models show comparable performances, but segmentation results are qualitatively different in many cases, as illustrated in figure 3.12. Differences are emphasized when segmenting highly heterogeneous lesions. In line with results obtained in previous works focused on MRI brain tumour segmentation, we found that SVM can capture complex multivariate relationships in the data, but classification results may suffer from spatial inconsistencies. The use of post-processing refinement stages is usually suggested, as in our strategy that contemplates the use of morphological operators. V-Net segmentations are more homogeneous on average and results obtained confirm that no additional processing is required.

We identified some limitations of the study. A very small number of in-house data was available, and the interpretation of the results would inevitably have to allow for that. However, as a general consideration, if the use of the deep network, pre-trained on neighbouring but not equivalent data and then fine-tuned with a very small dataset, allowed us to obtain acceptable performances, it is reasonable to think that it is possible to improve these results significantly when a broader dataset becomes available. For SVM, instead, it is difficult to foresee that there will be a benefit from the use of a wider training set having shown limitations in learning from varied heterogeneous data.

Moreover, both the classifiers made under-estimation and over-estimation errors when the pathology occupies a small volume (under the 100 elements) and has a heterogeneous context. These errors are because of misclassifications between the vessel and lesion voxels.

Probably, the simultaneous use of different MRI modalities could improve performances than when using single MRI modalities as experimented in [14], but the use of multimodal procedure would make it difficult the implementation of the automated segmentation in the radiotherapy clinical practice. Both the learning models showed advantages and drawbacks, but in the light of a great number of data available, a deep learning algorithm appears preferable.

The main conclusions of our experimental work can be summarised as follows:

- SVM shows good performances with a smaller training set, but it is unable to manage a high level of heterogeneity in the data and requires post-processing refinement stages.

- The V-Net model shows good performances when trained on multiple heterogeneous cases but requires data augmentations and transfer learning procedures to optimise performances.

- Users have found the segmentation strategy implemented either with SVM+Morphological Operator or V-Net, satisfactory and have considered misclassification errors acceptable and manageable with interactive phases devoted to manual refinements of the automated results.

The automated procedure is installed in the RT workflow to measure benefits in supporting brain metastasis delineation. Taking data is going and retraining of the segmentation procedure will be possible.

4

Mammography segmentation in Breast Cancer Treatment

This research work was presented at the SPRA thematic conference on pattern recognition and application to be held in Rome in November[30]. The main topic related to this work is illustrate the development of a deep learning study aimed to process and classify lesions in mammograms. Mammography has a central role in screening and diagnosis of breast lesions, allowing early detection of the pathology and reduction of fatal cases. Deep Convolutional Neural Networks have shown a great potentiality to address the issue of early detection of breast cancer with an acceptable level of accuracy and reproducibility. In the present work, we illustrate the development of a deep learning study aimed to process and classify lesions in mammograms with the use of slender neural networks not yet used in literature. For this reason, a traditional convolution network was compared with a novel one obtained making use of much more efficient depth wise separable convolution layers. Preliminary numerical results are detailed and future plans outlined.

4.1 Dataset

A large annotated set of mammograms with pixel-level annotations are needed to train CNNs for cancer detection and test their performances. Several public datasets of diagnosed patients with cancer breast are available. The digital database for mammography screening (DDSM) [15] and the Mammographic Image Analysis Society (MIAS) [68] are free mammography datasets available online. The following are available on request: INbreast database [54], Breast Cancer Digital Repository (BCDR) [50] and Image Retrieval in Medical Applications (IRMA) [56]. This other public and datasets are useful, and they are limited in terms of size and accessibility. This last set of datasets have a limited size. In the present work, we decided to use as reference dataset of our experiments, DDSM, since it is the one with the most significant number of cases available. However, the analysis of this dataset revealed some limitations, such as non-standard format and imprecise segmentation of the tumour area. To overcome these limitations, we have employed a subset of DDSM, named CIBIS (Curated Breast Imaging Subset of DDSM) subset [43], publicly available in The Cancer Imaging Archive (TCIA) [17].
The CBIS-DDSM is a significant public digitised screen-film mammography (SFM) database, including training and test cases of calcifications and masses. Being a subset of the DDSM, it includes both bilateral craniocaudal (CC) and mediolateral oblique (MLO) mammograms views, with an image size found equal to 3118x5001 in DICOM format. Each case was reviewed by a mammogram, also performing an update of the ROI segmentations [43].
The dataset includes 753 calcification cases and 891 mass cases; this number of data allows the use of deep learning techniques to train an algorithm for recognising pathological areas.

4.2 Preprocessing

The images of CBIS-DDSM are SFM, characterised by low contrast and noisy, with the presence of dust or written. In an attempt to simplify the classification task and improve performances, we have developed a pre-processing phase aimed to reduce the image to enhance quality, and limit the area visualised in the image to the breast area.

Preliminary images have been scaled to reduce the original large size. Four reduction factors were tested, reducing the original size at 12.5%, 25%, 50% 75%. The best resolution factor used was 12.5% which optimise both in terms of computational speed and work loose, as a side effect we obtain a reduction of noise.

The segmentation algorithm [60] adopted identifies the breast area addressing the following tasks: detection of the background and the detection of the chest wall.

The first task separates skin and air present in mammograms [47]; this task is applied to both the views. The algorithm uses the intensity of the pixels as a random variable and, uses the Anderson-Darling test to identify the foreground and background pixels area. Fig.4.1 shows the breast boundaries obtained in the MLO and CC views.

The second task separates the more homogeneous muscle regions from the breast regions. The Hough transform was used, which identifies the separation line of the two areas with different homogeneity [4]. The breast segmentation phase ends by performing the intersection of the masks obtained from the two tasks described above, and the result is shown in Fig. 4.1.

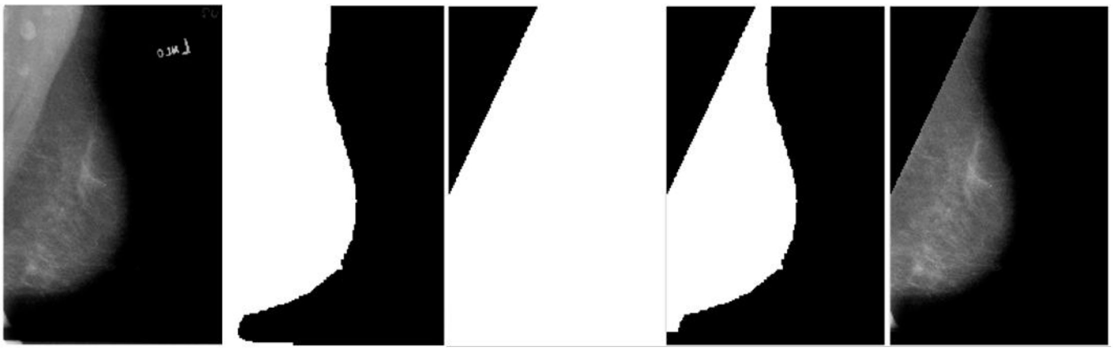


Figure 4.1: From left to right, the steps for breast detect. First step: segmentation with threshold; second step: segmentation with Hough transform; third step: intersection of the two masks and final result applied on the image.

Finally, we randomly extract patches of size 256x256 from the segmented area of the original images to create the dataset with which to train and test CNN. The crops extracted from each image are labeled pathological if they have a diseased area greater than 10% of the overall crop area. An equal number of healthy and pathological patches for both mass and calcification were extracted to have a balanced representation of classes. For each image, we have extracted a maximum of 250 healthy patches and 250 pathological patches taking the edge of mass and calcification as central points to be used for CNN training.

4.3 Proposed Methods

In order to solve the crop classification problem two Convolutional Neural Network architectures have been proposed and compared.

4.3.1 Networks Architecture

The first proposed architecture is similar to other neural network proposed for classification, and could be divided into two consecutive stages. The first stage, represented by a cascade of convolutional filters, analyses the input image in order to extract some relevant features. On the other hand, the last one, represented by a single fully connected layer, executes the actual classification. As can be seen from Tab. 4.1, the proposed network is composed of 8 convolutional layers, using 3x3 kernel filters and ReLu activation functions, coupled with batch normalization layers for regularization reasons. Most of these layers make use of a stride of 2 in order to reduce the dimension of the data from 256x256x1 to 2x2x128 (where the first two values represent the spatial dimension and the last one the number of channels/filters). In the final stage of the network, the last max pooling layer output is flattened in order to obtain a 512 vector on which a fully connected layer is applied in order to obtain a scalar output. Finally, on the obtained scalar value a sigmoid function is applied in order to get an output in the $[0, 1]$ range. This output value represents the probability attributed by the network to the input crop as depicting a pathological situation (mass or calcification). The second proposed method exploits depth-wise separable convolution layers, firstly introduced in [36]. These kind of layers are able to substitute convolutional layers by approximating their behaviour (using much fewer parameters) by adopting separable convolution layers followed by 1x1 convolutions, and more details could be found in [36]. The difference between the two architectures is exclusively in the use of Separable Convolution layers instead of traditional ones. As reported in Tab. 4.1 and Tab. 4.2 in this model, much fewer parameters are used, the difference is about one order of magnitude.

Table 4.1: Traditional CNN architecture. Layers reported in blue are considered task specific.

Layer	Output shape	Params #
Input	256x256x1	0
Conv (s1) + BN	256x256x32	448
Conv (s2) + BN	128x128x64	18,725
Conv (s2) + BN	64x64x128	743,68
Conv (s2) + BN	32x32x128	148,096
Conv (s2) + BN	16x16x128	148,096
Conv (s2) + BN	8x8x128	148,096
Conv (s2) + BN	4x4x128	148,096
Conv (s2) + BN	2x2x128	148,096
Flatten	512	0
Dense	1	513
Sigmoid	1	0
	Total params #	834,561

Table 4.2: Separable Convolution Network architecture. Layers reported in blue are considered task specific.

Layer	Output shape	Params #
Input	256x256x1	0
SepConv (s1) + BN	256x256x32	173
SepConv (s2) + BN	128x128x64	2,720
SepConv (s2) + BN	64x64x128	9,536
SepConv (s2) + BN	32x32x128	18,560
SepConv (s2) + BN	16x16x128	18,560
SepConv (s2) + BN	8x8x128	18,560
SepConv (s2) + BN	4x4x128	18,560
SepConv (s2) + BN	2x2x128	18,560
Flatten	512	0
Dense	1	513
Sigmoid	1	0
	Total params #	105,742

4.3.2 Training Procedure

Both networks described in the previous section have been trained using the same procedure. Considering the two tasks of mass and calcification classification, two models for each architecture have been trained. For each architecture, a two-step training approach has been adopted. In particular, in the first stage, the total number of 450'000 256x256 crops have been used. Among these 225'000 represented mammography portions in which no pathology was present, 112'500 depicted a crop of the exam in which masses were present while the remaining 112'500 were related to calcification. A small portion of this set, 20%, was randomly selected in order to create the validation set and so were removed from training. The two networks (one able to detect masses and the other calcifications) were trained jointly sharing the first five layers. In particular, layers coloured in black in Tab. 4.1 and Tab. 4.2 were considered shared between the two tasks (mass and calcification classification) while layers reported in blue were trained separately between the two networks, since they are considered task-specific. This training procedure was adopted in order to fully exploits the training dataset size. Binary cross-entropy and stochastic gradient descent (SGD) were chosen respectively as loss function and optimization algorithm. Learning rate was set to 0.01 and minibatch of 32 samples were used. An early stop criteria was adopted in order to avoid overfitting. In particular, the first step of the training was stopped as soon as one of the two networks showed overfitting behaviour (in particular validation loss started to increase). In the second stage of the training, the loss minimization was resumed starting from the weight obtained in the first stage using the same dataset and training conditions except for two main differences. The learning rate was reduced to 0.0005 in order to fine-tune the results obtained in the previous stage in the proximity of the local minimum already found. Moreover, the value of the weights related to the shared layers is kept frozen (or fixed) in order to provide a shared structure for the two networks.

4.4 Results

In this section results related to the models described in Sec.4.3 are reported.

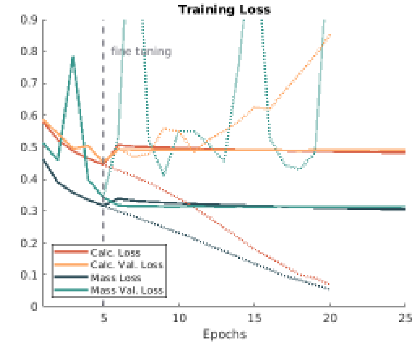


Figure 4.2: Convolution training results. Dashed lines represent values obtained without fine tuning.

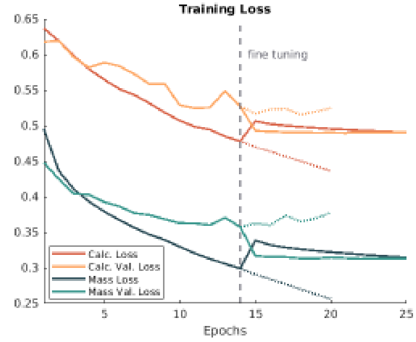


Figure 4.3: Separable convolution training results. Dashed lines represent values obtained without fine tuning.

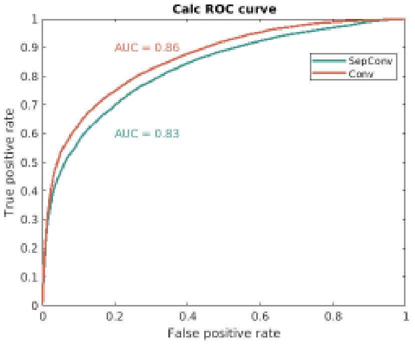


Figure 4.4: ROC curve for calcification classification task in INbreast.

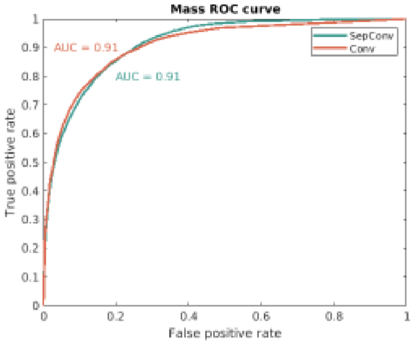


Figure 4.5: ROC curve for mass classification task.

4.4.1 Training Results

The training procedure described in Sec.4.3.2 leads to the results depicted in Fig.4.3 and Fig.4.2. In particular, Fig.4.3 depicts the loss values obtained with Separable convolutional layers while in Fig.4.2 the ones obtained with traditional convolutional layers are reported. Dashed lines represent values obtained without the adoption of fine tuning. As can be observed for both models, the early stop procedure coupled with fine-tuning secure the reaching of a local minimum without overfitting.

4.4.2 Evaluation Results

Following the training stage described above, the four networks (2 able to deal with mass classification problem and the other two related to calcification) were tested on patches extracted from CBIS dataset test images. In particular, each one of the ROC curves reported in Fig.4.4 and Fig.4.5 is obtained using 10'000 negative and 10'000 positive patches, created as reported in Sec.4.2. As can be observed from Fig.4.5, both methods produce similar results, both have AUC (Area Under Curve) scores equal to 0.91, even if the network obtained with SepConv layers uses significantly fewer parameters (as explained in Sec.4.3.1). The network based on traditional convolutions proved to work slightly better in the left part of the curve (small false positive) while the one that uses separable convolution is able to outperform the other in the right part of the graph. Regarding the calcificatCion classification task, which is more difficult in respect to the mass one, as shown in Fig.4.4, traditional convolution based network slightly outperforms the network that uses separable convolutions.

4.4.3 Experiment with Full Field Digital Mammography (FFDM)

A further experiment was conducted using the transfer learning technique by taking a pre-trained network and partly retraining it on new data. This experiment was conducted on the INbreast dataset, where, unlike the CBIS-DDSM dataset with SFM images, the FFDM images are digital and high-resolution. The main difference is that the CBIS-DDSM images indicate the general position of the lesions, without precise segmentation, while in the INbreast database the segmentations of the ROI are pixel-based. For the retraining procedures shown in Fig.4.6 for the traditional convolutions and in Fig.4.7 for the Separable Convolutions, the same dataset extraction procedure was performed as described in Sec.4.2 and Sec.4.3.2. In particular, the total number of 110'000 256x256 crops have been used. Among these 55'000 represented mammography portions in which no pathology was present, 18'000 depicted a crop of the exam in which masses were present while the remaining 37'000 were related to calcification. A small portion of this set, 20%, was randomly selected in order to create the validation set and so were removed from training. Therefore, thanks to the transference learning technique, used on the neural networks describe in Tab.4.1 and Tab.4.2 with the same test procedure described in Sec.4.4.2 on the INbreast dataset, the results obtained are visible in Fig.4.8 and in Fig.4.9. As can be observed from Fig.4.9, both methods produce similar results, both have AUC (Area Under Curve) scores equal to 0.94, even if the network obtained with SepConv layers uses significantly fewer parameters (as explained in Sec.4.3.1). Regarding the calcification classification task, which is more difficult for the mass one, as shown in Fig.4.8, the traditional network with AUC scores equal to 0.95, based on convolution, outweighs the network it uses separable convolutions with

AUC equivalent to 0.87. Compared to the SFM images, performance improvement is detected, and a corresponding difference between the ROC curves for the detection of calcifications, probably due to the point-like nature of the pathology.

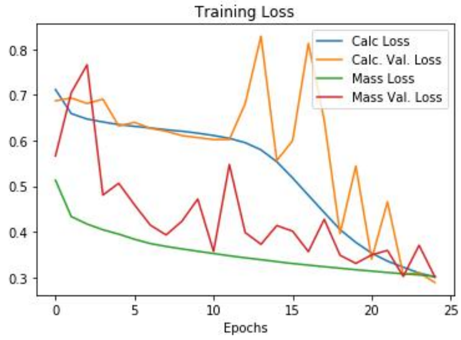


Figure 4.6: Convolution retraining results with INbreast dataset.

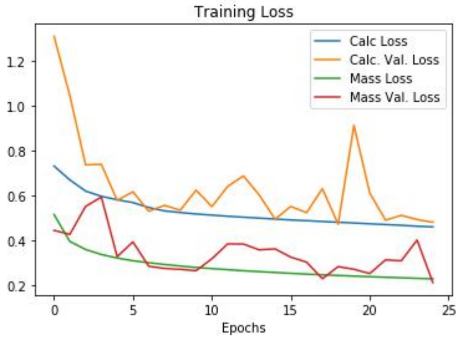


Figure 4.7: Separable convolution retraining results with INbreast dataset.

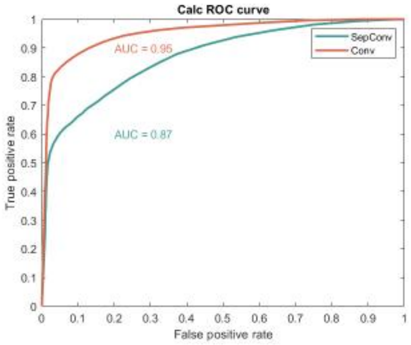


Figure 4.8: ROC curve for calcification classification task with INbreast dataset.

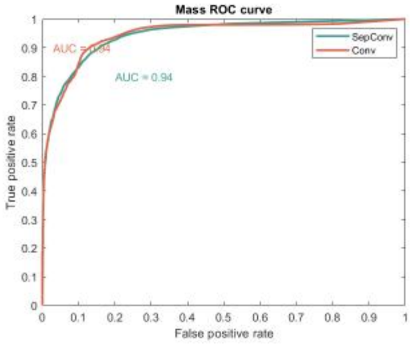


Figure 4.9: ROC curve for mass classification task with INbreast dataset.

4.5 Detection Workflow

The classification of patches within the entire mammography has been developed by following the workflow described below (Fig.4.10). In the first step, the breast is identified from the original image with a segmentation described in sec. 4.2, to speed up the classification operations. In the second step, only the segmented part, including the breast, is divided into partially overlapped tile images of 256x256 crops using sliding windows of 64px. In the third phase, each box is classified by CNN, and the result is combined in order to create a heat map. In particular, the heatmap value in a specific pixel location, Hij, is obtained as the average between the N outputs obtained applying the classification method f to the N crops Ck containing that particular pixel.

$$\mathrm{H_{i,j}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i,j} \in \mathrm{C_k}} \mathrm{f(C_k)} \tag{4.1}$$

The workflow for lesion detection has been integrated into the medical application, described in subsection 4.5.1

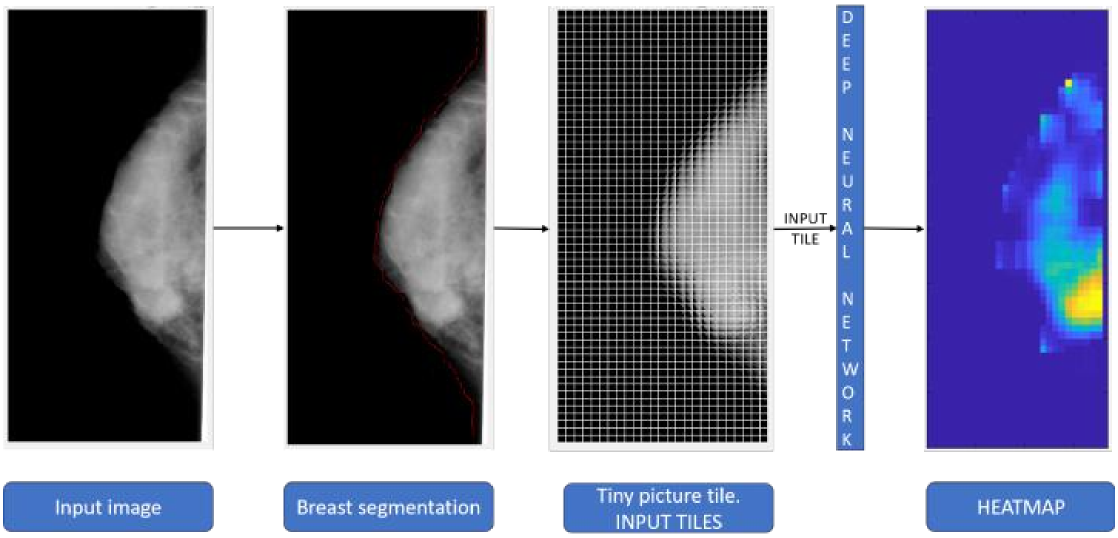


Figure 4.10: The workflow for detection lesion in the CIBIS mammogram.

4.5.1 The Software System: Graphical User Interface (GUI) and sample program runs

The above-illustrated computational procedures are integrated into a unified framework, to support the segmentation of mass and calc in the mammography screening workflow (see Fig. 4.11).

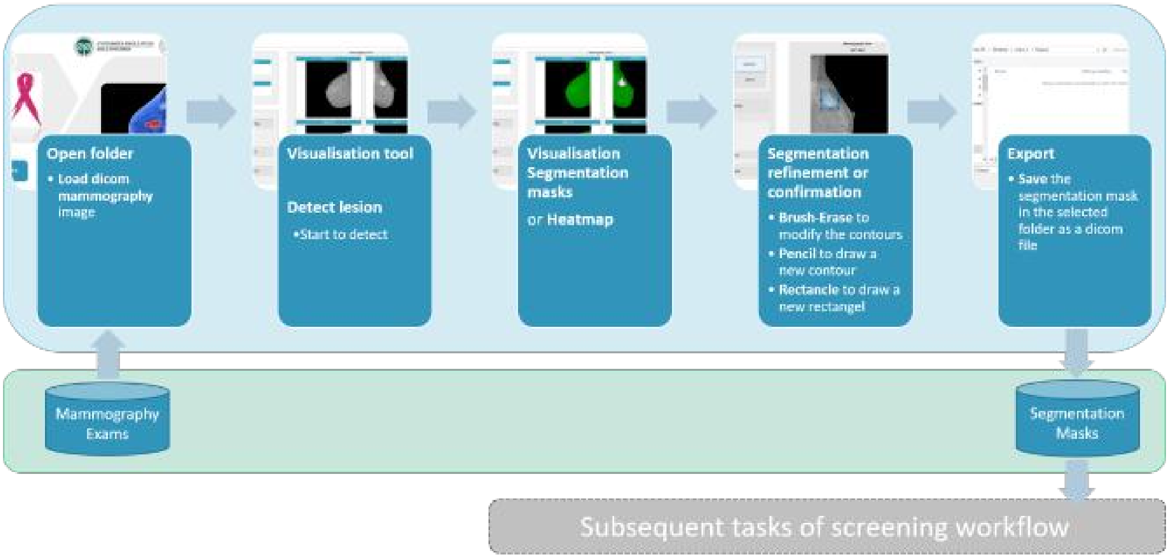

Open folder
• Load dicom mammography image
Visualisation tool
Detect lesion
•Start to detect
Visualisation Segmentation masks
or Heatmap
Segmentation refinement or confirmation
• Brush-Erase to modify the contours
• Pencil to draw a new contour
• Rectancle to draw a new rectangel
Export
• Save the segmentation mask in the selected folder as a dicom file
Mammography Exams
Segmentation Masks
Subsequent tasks of screening workflow

Figure 4.11: The workflow of the functional structure of the software system implementing the segmentation procedure.

The software design started with the collection and analysis of requirements to outline the user model and operational conditions. We assessed cognitive and perceptual processes, attitudes and limitations involved in visual inspection and analysis tasks. Operation requirements concerning hardware facilities and input and output devices are obtained from protocols in use in mammography domain. The system was implemented in MATLAB (MathWorks®).
Fig. 4.12 shows the system interfaces documenting the initial phases of the session and input/output facilities. Visualisation tools are available for a reliable inspection of the four images bilateral craniocaudal (CC) left and right and the corresponding mediolateral oblique (MLO) mammograms views.

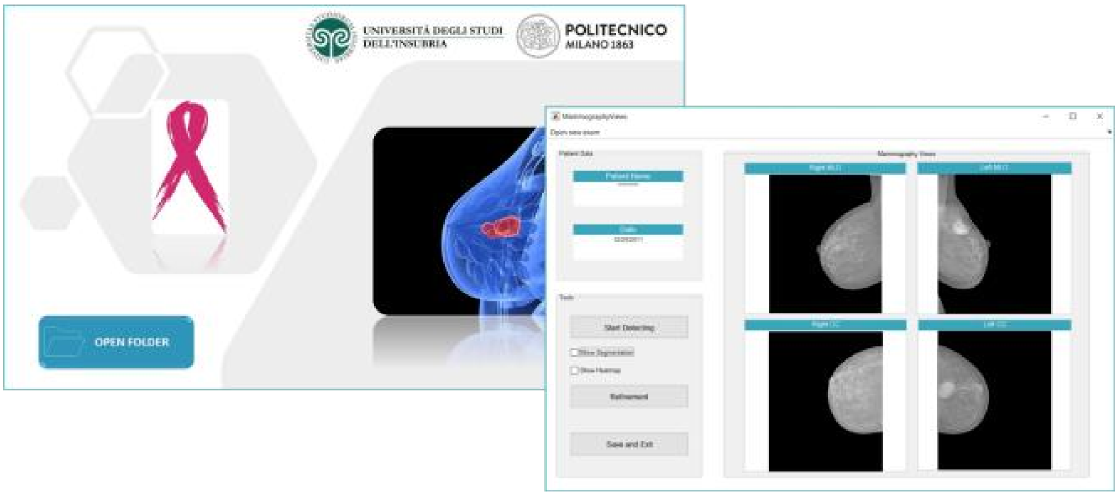


Figure 4.12: Screenshots of the system interfaces documenting the initial input/output phases.

On the left part of the GUI, we found options made available by the system to develop detecting task. With "Start Detecting" the segmentation procedure is performed, and the boundary of the mask obtained is superimposed on the four projections, as shown in Fig.4.13.

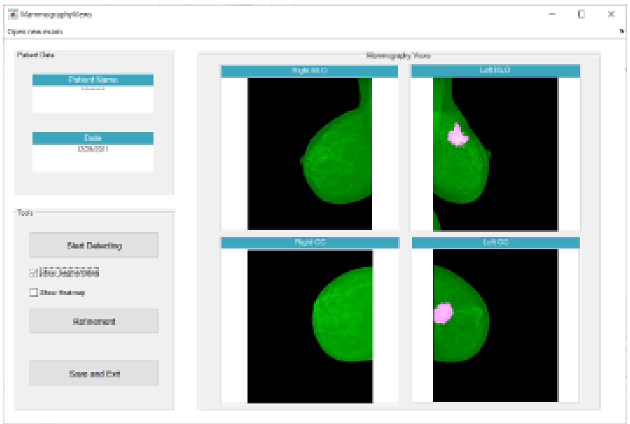
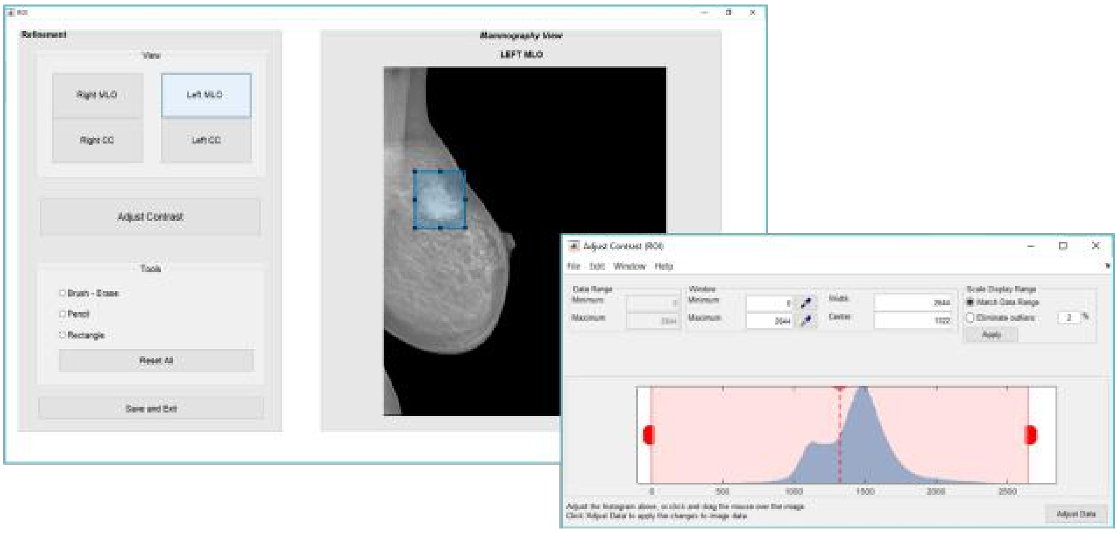


Figure 4.13: Screenshot of the system interfaces documenting the visualisation of the segmentation masks computed by CNN.

The system allows the user to export the segmentation mask, to refine the segmentation results, to proceed in a further segmentation in the same view or to return to the initial phase by unchecking "show segmentation" command.



Data Range
Minimum 0
Maximum 2644

Window
Minimum 0
Maximum 2644
Width 2644
Center 1322

Scale Display Range
● Match Data Range
○ Eliminate outliers 2 %

Adjust the histogram above, or click and drag the mouse over the image.
Click 'Adjust Data' to apply the changes to image data.

Figure 4.14: Screenshots of the System interfaces documenting the validation of the computed rectangular segmentation masks.

Figure 4.14 shows options made available by the system to validate the automated segmentation. The segmented mask is shown superimposed on corresponding projection selected with the ability to adjust the contrast. The user is allowed to modify the contour by checking on “Brush-Eraser” command or delineating the entire segment in case the automated result occupies a wrong position (check Pencil) or draw a new rectangle on the lesion.

4.6 Conclusion

As seen in our experimental context, breast lesions can be automatically detected by means of deep learning procedures. Preliminary results obtained are encouraging and basing on them future work is planned. We aim at developing global methods, i.e. algorithms able to analyze a clinal exam in its entirety, by combining results obtained locally. In particular, by using a sliding windows approach, we aim at obtaining a classification heatmap by combining and averaging the output of the local (i.e. crop) classifiers described in this work.

5 Conclusions and Future Works

5.1 General Conclusions

AI-based diagnostics uses the patient's unique history as a baseline against which small deviations signal a possible health condition that needs further investigation and treatment. AI is likely initially to be adopted as an aid, rather than a substitute, for human doctors. It will increase doctors' diagnosis, but in the meantime, it will also provide valuable information so that AI can continuously learn and improve. This ongoing interaction between human doctors and AI diagnostics will enhance the accuracy of the systems and, over time, provide enough confidence for humans to fully delegate the task to the AI system to operate autonomously. Fully automatic systems have both legal and moral interactions and have not yet been addressed at this time. The first application developed for the segmentation of metastases was developed thanks to the availability offered by the ASST hospital, which provided images of a particular pathology not treated in the literature with the aid of artificial intelligence algorithms. The algorithms used were at first the same as those used for gliomas, a pathology that is always cerebral but with a larger volume of space occupied. For the SVM, the results obtained were accepted by the experts, as for the V-Net thanks to the fact that manual modification of the segmentation could then be carried out. The V-Net has been used for this type of task thanks to the fact that its loss function is based on the Dice index, widely used in medicine to measure performance. For the second task of mammography screening, we wanted to tackle a problem extensively treated with neural networks in literature, addressing it, however with the use of slender neural networks not yet used. It was thus comparing a traditional network with one with separable convoluted levels. This has led to results comparable to traditional neural networks, with the particularity of decreasing the calculation times and more easily inspected during the training phase, given the lower parameter space. This algorithm was integrated into an application for medical use to be tested directly in the hospital to collect new ad hoc data to carry out an update of the network.

5.2 Future works

Future extensions of MRI segmentation in metastases can be made by acquiring other images made with a sagittal or coronal acquisition. With this new dataset, you could see how the models used (V-Net and SVM) behave if, with a re-training, the results of the classification made improve. However, being a new dataset, they may not respect the volumetric specifications of the axial acquisitions, and therefore a new set of algorithmic models should be tested to perform segmentation.

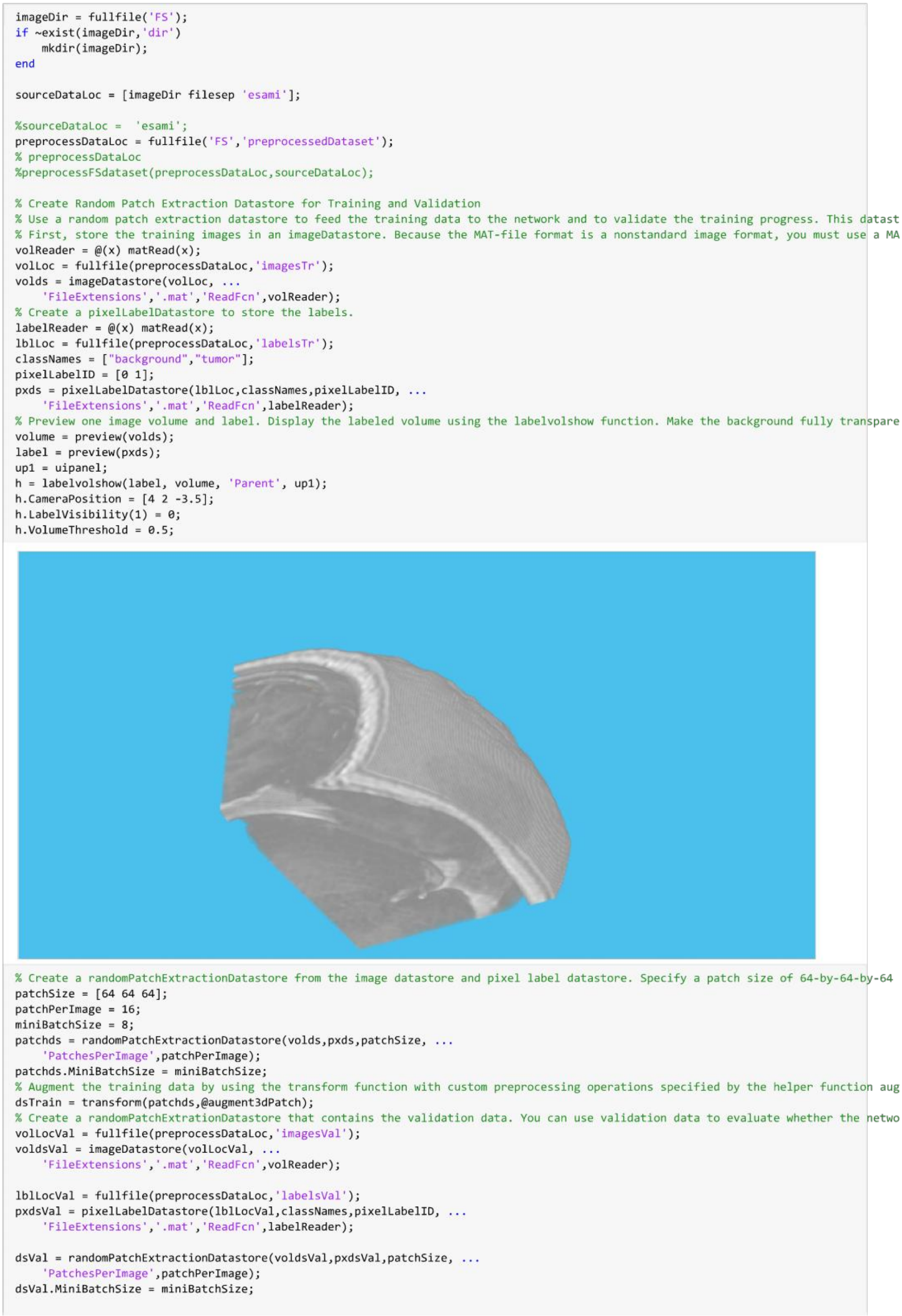
As for research in the field of mammography screening, with the arrival of a dataset with digital mammograms provided by the hospital, it will be possible to retrain the network and test the algorithms directly in the field for immediate feedback that can be used to refine the neural network.

6
Appendixes

6.1 MATLAB Code of chapter 3

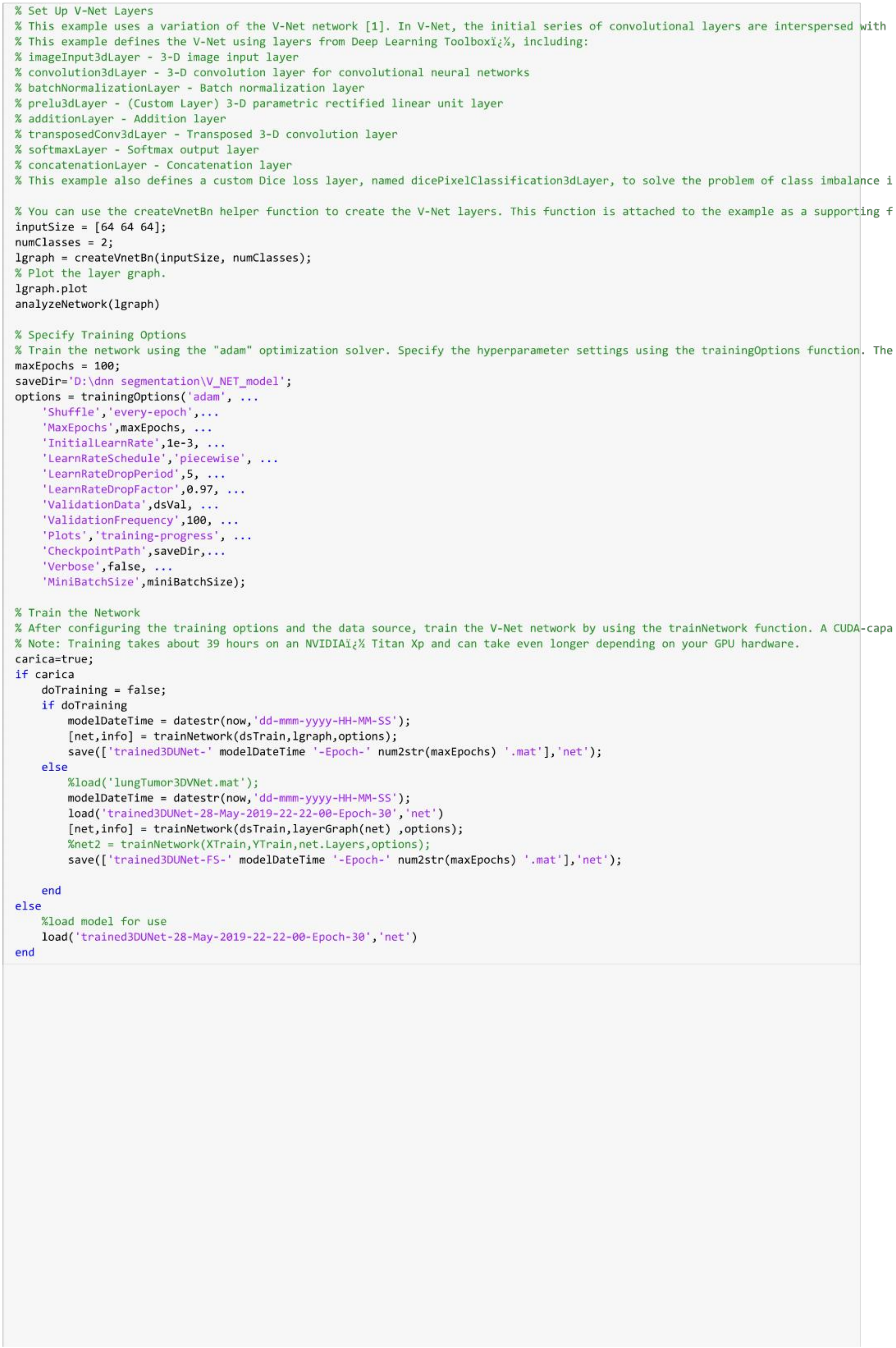
Below is an example of the analysis of the results of the V-Net pre-training epochs using pre-processed T1c scans of BraTS data. In this experiment, the number of epochs is 100, and the preprocessed dataset (43 Volumes in-house) was split again into 82% for training, 6% for validation and 12% for the test:

- 34 Volumes for the training set.
- 3 Volumes for the validation set.
- 6 Volumes for the test set.

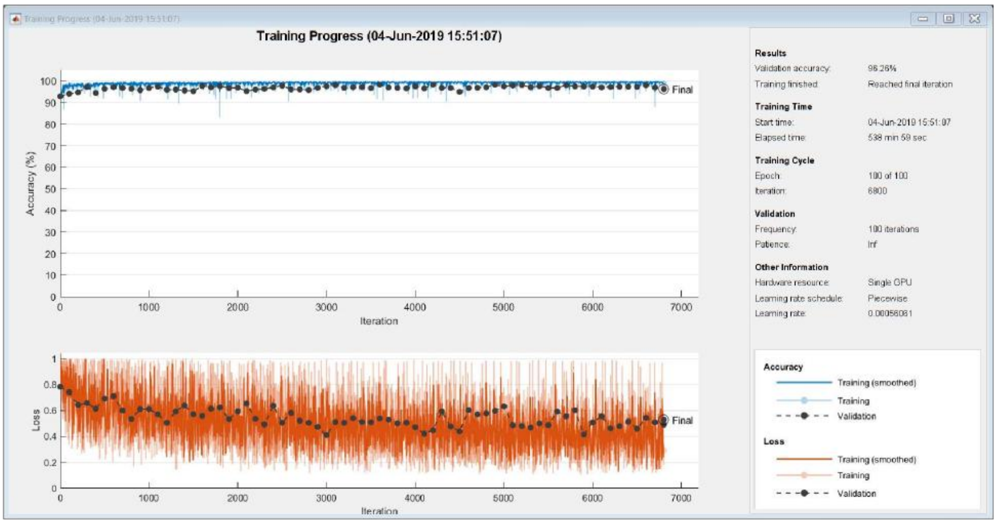


```
imageDir = fullfile('FS');
if ~exist(imageDir,'dir')
    mkdir(imageDir);
end

sourceDataLoc = [imageDir filesep 'esami'];

%sourceDataLoc =  'esami';
preprocessDataLoc = fullfile('FS','preprocessedDataset');
% preprocessDataLoc
%preprocessFSdataset(preprocessDataLoc,sourceDataLoc);

% Create Random Patch Extraction Datastore for Training and Validation
% Use a random patch extraction datastore to feed the training data to the network and to validate the training progress. This datast
% First, store the training images in an imageDatastore. Because the MAT-file format is a nonstandard image format, you must use a MA
volReader = @(x) matRead(x);
volLoc = fullfile(preprocessDataLoc,'imagesTr');
volds = imageDatastore(volLoc, ...
    'FileExtensions','.mat','ReadFcn',volReader);
% Create a pixelLabelDatastore to store the labels.
labelReader = @(x) matRead(x);
lblLoc = fullfile(preprocessDataLoc,'labelsTr');
classNames = ["background","tumor"];
pixelLabelID = [0 1];
pxds = pixelLabelDatastore(lblLoc,classNames,pixelLabelID, ...
    'FileExtensions','.mat','ReadFcn',labelReader);
% Preview one image volume and label. Display the labeled volume using the labelvolshow function. Make the background fully transpare
volume = preview(volds);
label = preview(pxds);
up1 = uipanel;
h = labelvolshow(label, volume, 'Parent', up1);
h.CameraPosition = [4 2 -3.5];
h.LabelVisibility(1) = 0;
h.VolumeThreshold = 0.5;
```

```
% Create a randomPatchExtractionDatastore from the image datastore and pixel label datastore. Specify a patch size of 64-by-64-by-64
patchSize = [64 64 64];
patchPerImage = 16;
miniBatchSize = 8;
patchds = randomPatchExtractionDatastore(volds,pxds,patchSize, ...
    'PatchesPerImage',patchPerImage);
patchds.MiniBatchSize = miniBatchSize;
% Augment the training data by using the transform function with custom preprocessing operations specified by the helper function aug
dsTrain = transform(patchds,@augment3dPatch);
% Create a randomPatchExtrationDatastore that contains the validation data. You can use validation data to evaluate whether the netwo
volLocVal = fullfile(preprocessDataLoc,'imagesVal');
voldsVal = imageDatastore(volLocVal, ...
    'FileExtensions','.mat','ReadFcn',volReader);

lblLocVal = fullfile(preprocessDataLoc,'labelsVal');
pxdsVal = pixelLabelDatastore(lblLocVal,classNames,pixelLabelID, ...
    'FileExtensions','.mat','ReadFcn',labelReader);

dsVal = randomPatchExtractionDatastore(voldsVal,pxdsVal,patchSize, ...
    'PatchesPerImage',patchPerImage);
dsVal.MiniBatchSize = miniBatchSize;
```



```
% Set Up V-Net Layers
% This example uses a variation of the V-Net network [1]. In V-Net, the initial series of convolutional layers are interspersed with
% This example defines the V-Net using layers from Deep Learning Toolboxï¿½, including:
% imageInput3dLayer - 3-D image input layer
% convolution3dLayer - 3-D convolution layer for convolutional neural networks
% batchNormalizationLayer - Batch normalization layer
% prelu3dLayer - (Custom Layer) 3-D parametric rectified linear unit layer
% additionLayer - Addition layer
% transposedConv3dLayer - Transposed 3-D convolution layer
% softmaxLayer - Softmax output layer
% concatenationLayer - Concatenation layer
% This example also defines a custom Dice loss layer, named dicePixelClassification3dLayer, to solve the problem of class imbalance i

% You can use the createVnetBn helper function to create the V-Net layers. This function is attached to the example as a supporting f
inputSize = [64 64 64];
numClasses = 2;
lgraph = createVnetBn(inputSize, numClasses);
% Plot the layer graph.
lgraph.plot
analyzeNetwork(lgraph)

% Specify Training Options
% Train the network using the "adam" optimization solver. Specify the hyperparameter settings using the trainingOptions function. The
maxEpochs = 100;
saveDir='D:\dnn segmentation\V_NET_model';
options = trainingOptions('adam', ...
    'Shuffle','every-epoch',...
    'MaxEpochs',maxEpochs, ...
    'InitialLearnRate',1e-3, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',5, ...
    'LearnRateDropFactor',0.97, ...
    'ValidationData',dsVal, ...
    'ValidationFrequency',100, ...
    'Plots','training-progress', ...
    'CheckpointPath',saveDir,...
    'Verbose',false, ...
    'MiniBatchSize',miniBatchSize);

% Train the Network
% After configuring the training options and the data source, train the V-Net network by using the trainNetwork function. A CUDA-capa
% Note: Training takes about 39 hours on an NVIDIAï¿½ Titan Xp and can take even longer depending on your GPU hardware.
carica=true;
if carica
    doTraining = false;
    if doTraining
        modelDateTime = datestr(now,'dd-mmm-yyyy-HH-MM-SS');
        [net,info] = trainNetwork(dsTrain,lgraph,options);
        save(['trained3DUNet-' modelDateTime '-Epoch-' num2str(maxEpochs) '.mat'],'net');
    else
        %load('lungTumor3DVNet.mat');
        modelDateTime = datestr(now,'dd-mmm-yyyy-HH-MM-SS');
        load('trained3DUNet-28-May-2019-22-22-00-Epoch-30','net')
        [net,info] = trainNetwork(dsTrain,layerGraph(net) ,options);
        %net2 = trainNetwork(XTrain,YTrain,net.Layers,options);
        save(['trained3DUNet-FS-' modelDateTime '-Epoch-' num2str(maxEpochs) '.mat'],'net');

    end
else
    %load model for use
    load('trained3DUNet-28-May-2019-22-22-00-Epoch-30','net')
end
```



Results

Validation accuracy: 96.26%
Training finished: Reached final iteration

Training Time

Start time: 04-Jun-2019 15:51:07
Elapsed time: 538 min 59 sec

Training Cycle

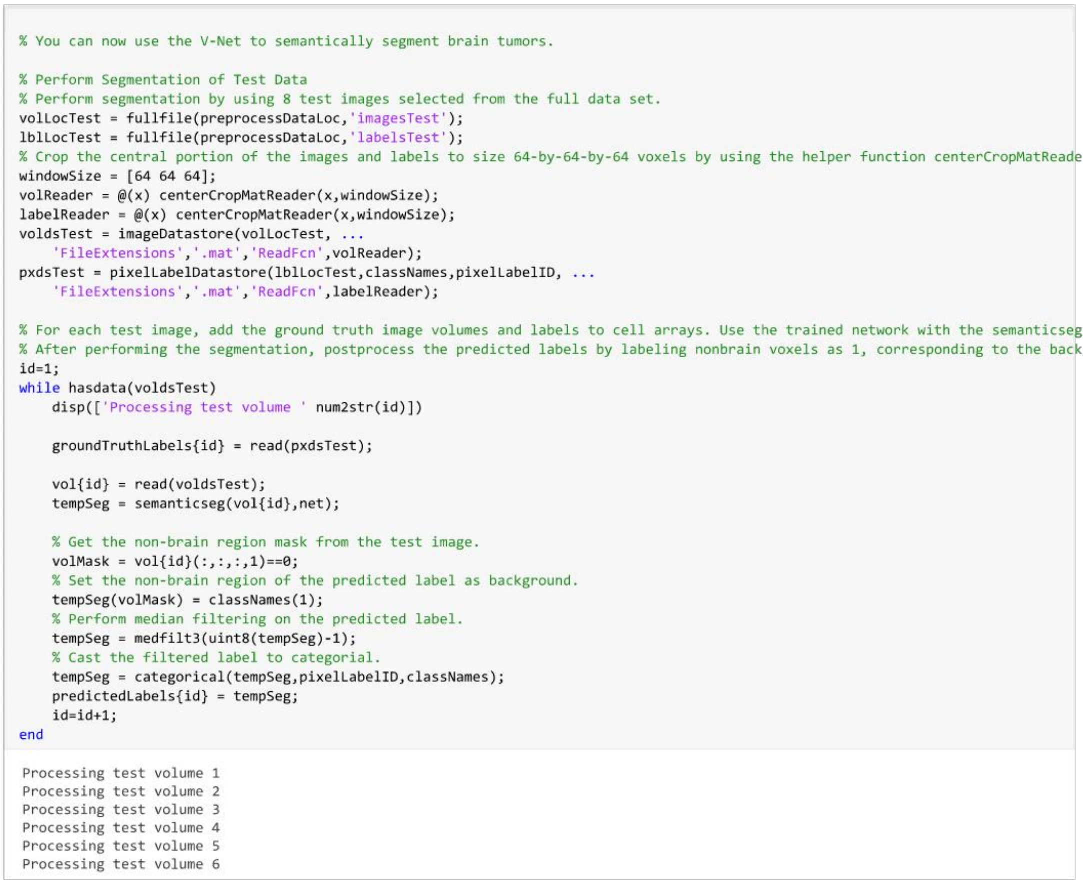
Epoch: 100 of 100
Iteration: 6800

Validation

Frequency: 100 iterations
Patience: Inf

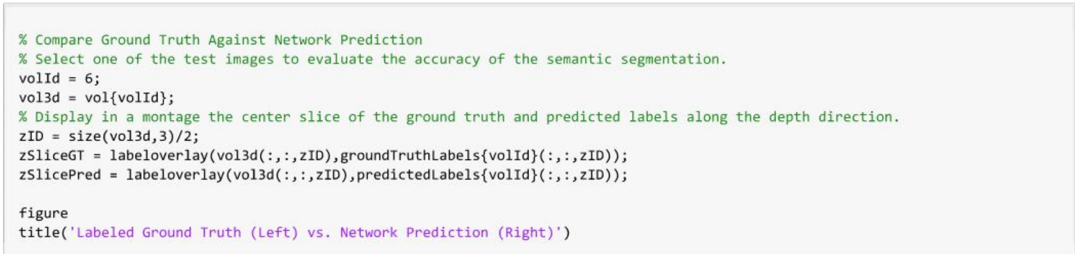
Other Information

Hardware resource: Single GPU
Learning rate schedule: Piecewise
Learning rate: 0.00056081

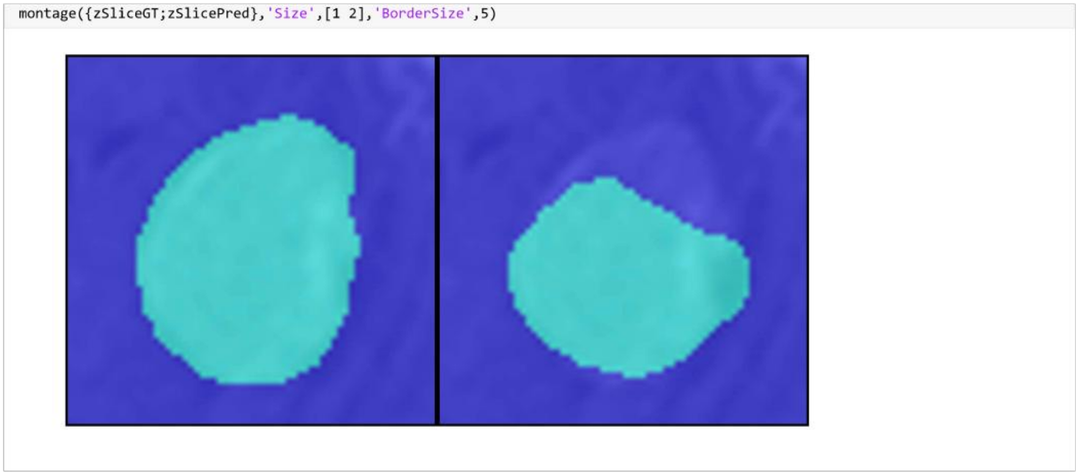


```
% You can now use the V-Net to semantically segment brain tumors.

% Perform Segmentation of Test Data
% Perform segmentation by using 8 test images selected from the full data set.
volLocTest = fullfile(preprocessDataLoc,'imagesTest');
lblLocTest = fullfile(preprocessDataLoc,'labelsTest');
% Crop the central portion of the images and labels to size 64-by-64-by-64 voxels by using the helper function centerCropMatReader. T
windowSize = [64 64 64];
volReader = @(x) centerCropMatReader(x,windowSize);
labelReader = @(x) centerCropMatReader(x,windowSize);
voldsTest = imageDatastore(volLocTest, ...
    'FileExtensions','.mat','ReadFcn',volReader);
pxdsTest = pixelLabelDatastore(lblLocTest,classNames,pixelLabelID, ...
    'FileExtensions','.mat','ReadFcn',labelReader);

% For each test image, add the ground truth image volumes and labels to cell arrays. Use the trained network with the semanticseg fun
% After performing the segmentation, postprocess the predicted labels by labeling nonbrain voxels as 1, corresponding to the backgrou
id=1;
while hasdata(voldsTest)
    disp(['Processing test volume ' num2str(id)])

    groundTruthLabels{id} = read(pxdsTest);

    vol{id} = read(voldsTest);
    tempSeg = semanticseg(vol{id},net);

    % Get the non-brain region mask from the test image.
    volMask = vol{id}(:,:,:,1)==0;
    % Set the non-brain region of the predicted label as background.
    tempSeg(volMask) = classNames(1);
    % Perform median filtering on the predicted label.
    tempSeg = medfilt3(uint8(tempSeg)-1);
    % Cast the filtered label to categorial.
    tempSeg = categorical(tempSeg,pixelLabelID,classNames);
    predictedLabels{id} = tempSeg;
    id=id+1;
end
```

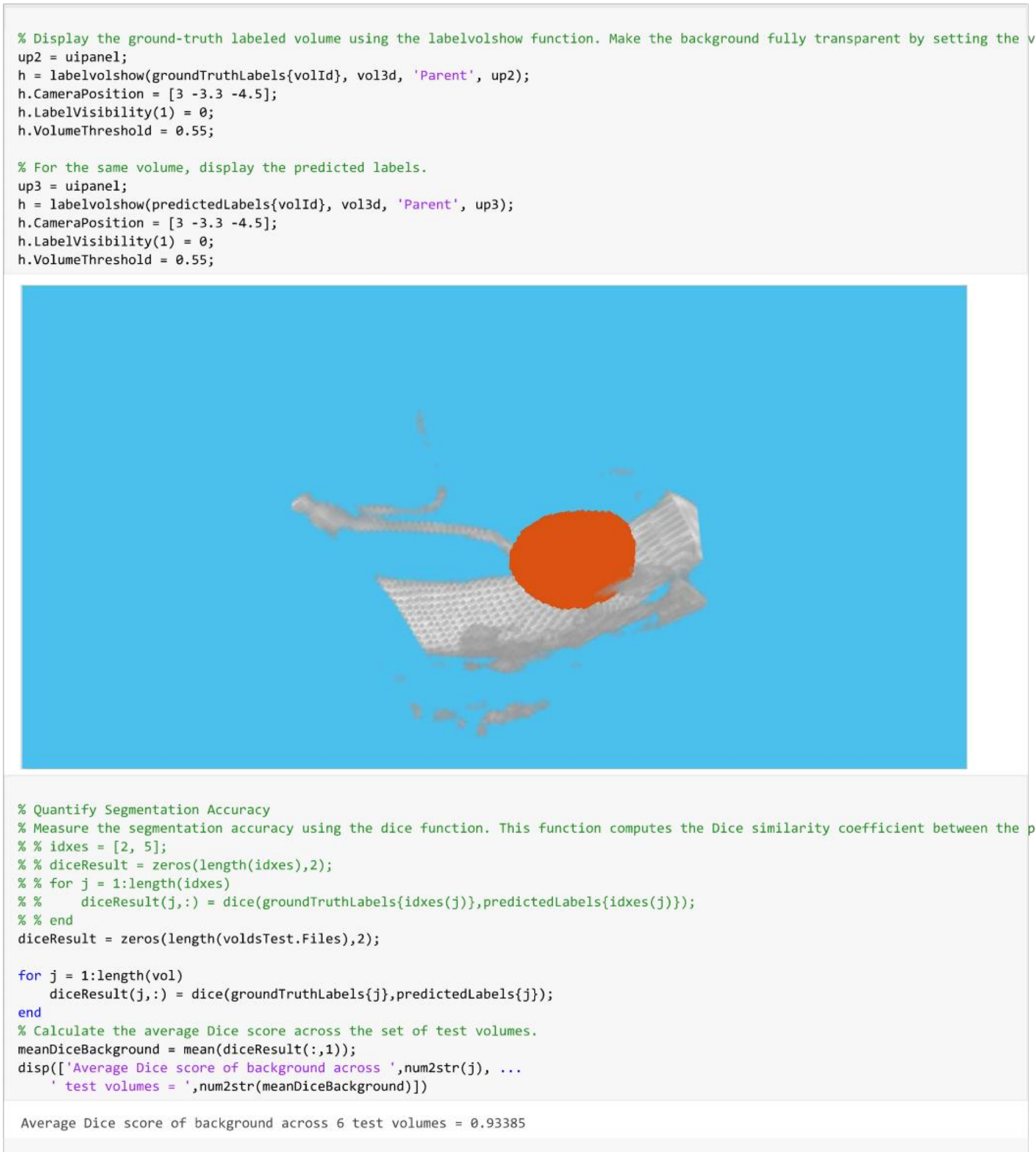
```
Processing test volume 1
Processing test volume 2
Processing test volume 3
Processing test volume 4
Processing test volume 5
Processing test volume 6
```



```
% Compare Ground Truth Against Network Prediction
% Select one of the test images to evaluate the accuracy of the semantic segmentation.
volId = 6;
vol3d = vol{volId};
% Display in a montage the center slice of the ground truth and predicted labels along the depth direction.
zID = size(vol3d,3)/2;
zSliceGT = labeloverlay(vol3d(:,:,zID),groundTruthLabels{volId}(:,:,zID));
zSlicePred = labeloverlay(vol3d(:,:,zID),predictedLabels{volId}(:,:,zID));

figure
title('Labeled Ground Truth (Left) vs. Network Prediction (Right)')
```



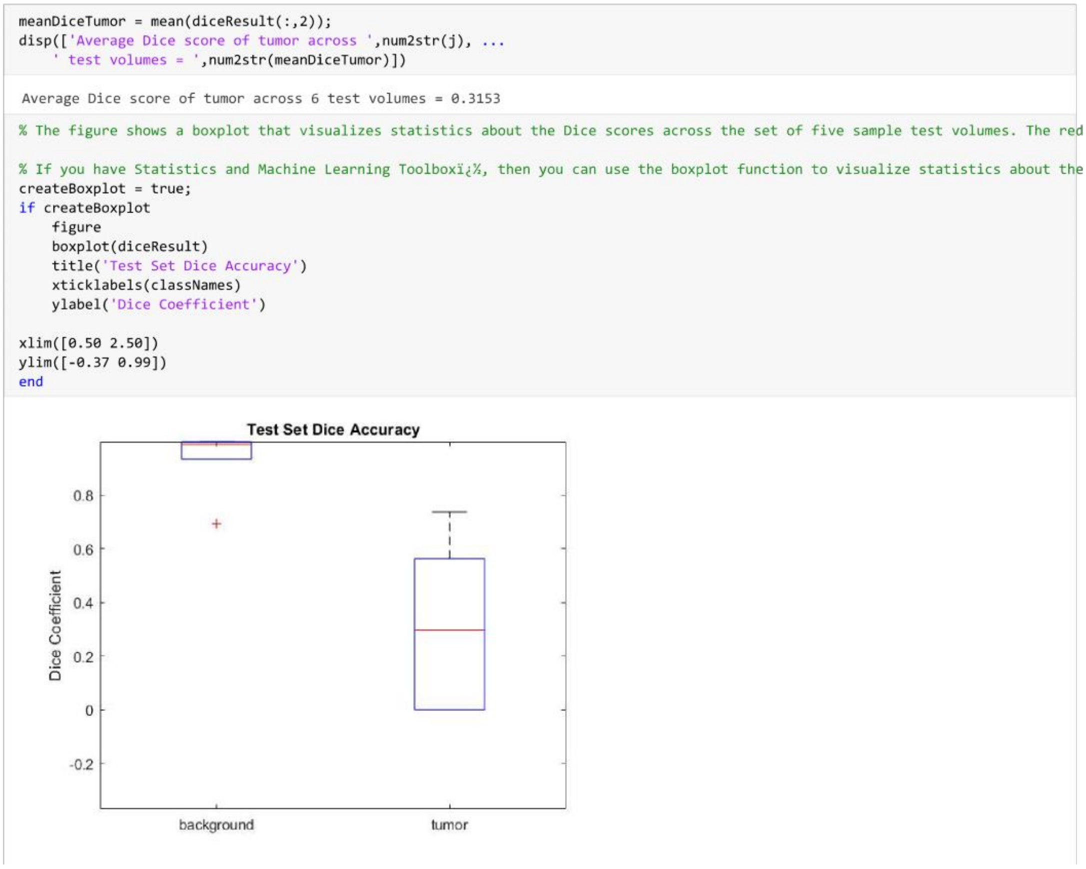
```
montage({zSliceGT;zSlicePred},'Size',[1 2],'BorderSize',5)
```



```
% Display the ground-truth labeled volume using the labelvolshow function. Make the background fully transparent by setting the visib
up2 = uipanel;
h = labelvolshow(groundTruthLabels{volId}, vol3d, 'Parent', up2);
h.CameraPosition = [3 -3.3 -4.5];
h.LabelVisibility(1) = 0;
h.VolumeThreshold = 0.55;

% For the same volume, display the predicted labels.
up3 = uipanel;
h = labelvolshow(predictedLabels{volId}, vol3d, 'Parent', up3);
h.CameraPosition = [3 -3.3 -4.5];
h.LabelVisibility(1) = 0;
h.VolumeThreshold = 0.55;
```

```
% Quantify Segmentation Accuracy
% Measure the segmentation accuracy using the dice function. This function computes the Dice similarity coefficient between the predi
% % idxes = [2, 5];
% % diceResult = zeros(length(idxes),2);
% % for j = 1:length(idxes)
% %     diceResult(j,:) = dice(groundTruthLabels{idxes(j)},predictedLabels{idxes(j)});
% % end
diceResult = zeros(length(voldsTest.Files),2);

for j = 1:length(vol)
    diceResult(j,:) = dice(groundTruthLabels{j},predictedLabels{j});
end
% Calculate the average Dice score across the set of test volumes.
meanDiceBackground = mean(diceResult(:,1));
disp(['Average Dice score of background across ',num2str(j), ...
    ' test volumes = ',num2str(meanDiceBackground)])
```

```
Average Dice score of background across 6 test volumes = 0.93385
```



```
meanDiceTumor = mean(diceResult(:,2));
disp(['Average Dice score of tumor across ',num2str(j), ...
    ' test volumes = ',num2str(meanDiceTumor)])
```

```
Average Dice score of tumor across 6 test volumes = 0.3153
```

```
% The figure shows a boxplot that visualizes statistics about the Dice scores across the set of five sample test volumes. The red lin

% If you have Statistics and Machine Learning Toolboxï¿½, then you can use the boxplot function to visualize statistics about the Dic
createBoxplot = true;
if createBoxplot
    figure
    boxplot(diceResult)
    title('Test Set Dice Accuracy')
    xticklabels(classNames)
    ylabel('Dice Coefficient')

xlim([0.50 2.50])
ylim([-0.37 0.99])
end
```

Bibliography

[1] Dina Abdelhafiz, Clifford Yang, Reda Ammar, and Sheida Nabavi. Deep convolutional neural networks for mammography: advances, challenges and applications. *BMC bioinformatics*, 20(11):281, 2019.

[2] Zeynettin Akkus, Alfiia Galimzianova, Assaf Hoogi, Daniel L. Rubin, and Bradley J. Erickson. Deep learning for brain MRI segmentation: State of the art and future directions. *J Digit Imaging*, 30(4):449–459, 2017.

[3] Min Sun Bae, Woo Kyung Moon, Jung Min Chang, Hye Ryoung Koo, Won Hwa Kim, Nariya Cho, Ann Yi, Bo La Yun, Su Hyun Lee, Mi Young Kim, et al. Breast cancer detected with screening us: reasons for nondetection at mammography. *Radiology*, 270(2):369–377, 2014.

[4] D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111 – 122, 1981.

[5] Stefan Bauer, Lutz-P. Nolte, and Mauricio Reyes. Fully automatic segmentation of brain tumor images using support vector machine classification in combination with hierarchical conditional random field regularization. *Med Image Comput Comput Assist Interv*, 14:354–361, 2011.

[6] Stefan Bauer, Roland Wiest, Lutz-P. Nolte, and Mauricio Reyes. A survey of MRI-based medical image analysis for brain tumor studies. *Phys Med Biol*, 58(13):R97–129, 2013.

[7] Norbert Bergner, Bernd F. M. Romeike, Rupert Reichart, Rolf Kalff, Christoph Krafft, and Jürgen Popp. Tumor margin identification and prediction of the primary tumor from brain metastases using FTIR imaging and support vector machines. *Analyst*, 138(14):3983–3990, 2013.

[8] Elisabetta Binaghi, Valentina Pedoia, and Sergio Balbi. Meningioma and peritumoral edema segmentation of preoperative MRI brain scans. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 6(4):362–370, 2018.

[9] André Bleau and Lewis Leon. Watershed-based segmentation and region merging. *Computer Vision and Image Understanding*, 77:317–370, 2000.

[10] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[11] Mireille Broeders, Sue Moss, Lennarth Nyström, Sisse Njor, Håkan Jonsson, Ellen Paap, Nathalie Massat, Stephen Duffy, Elsebeth Lynge, and Eugenio Paci. The impact of mammographic screening on breast cancer mortality in europe: a review of observational studies. *Journal of medical screening*, 19(1_suppl):14–25, 2012.

[12] Daniel N. Cagney, Allison M. Martin, Paul J. Catalano, Amanda J. Redig, Nancy U. Lin, Eudocia Q. Lee, Patrick Y. Wen, Ian F. Dunn, Wenya Linda Bi, Stephanie E. Weiss, Daphne A. Haas-Kogan, Brian M. Alexander, and Ayal A. Aizer. Incidence and prognosis of patients with brain metastases at diagnosis of systemic malignancy: a population-based study. *Neuro-oncology*, 19(11):1511–1521, 2017.

[13] H. Cai, R. Verma, Y. Ou, S. Lee, E. R. Melhem, and C. Davatzikos. Probabilistic segmentation of brain tumors based on multi-modality magnetic resonance images. In *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 600–603, 2007.

[14] Odelin Charron, Alex Lallement, Delphine Jarnet, Vincent Noblet, Jean-Baptiste Clavier, and Philippe Meyer. Automatic detection and segmentation of brain metastases on multimodal MR images with a deep convolutional neural network. *Comput. Biol. Med.*, 95:43–54, 2018.

[15] Ioanna Christoyianni, Emmanouil Constantinou, and Evangelos Dermatas. Automatic detection of abnormal tissue in bilateral mammograms using neural networks. In George A. Vouros and Themistoklis Panayiotopoulos, editors, *Methods and Applications of Artificial Intelligence*, pages 267–275, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[16] Dan CireşAn, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural networks*, 32:333–338, 2012.

[17] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, et al. The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26(6):1045–1057, 2013.

[18] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[19] Joseph De Brabanter, Bart De Moor, Johan AK Suykens, Tony Van Gestel, and Joos PL Vandewalle. *Least squares support vector machines*. World scientific, 2002.

[20] Nick Efford. *Digital image processing: a practical introduction using java (with CD-ROM)*. Addison-Wesley Longman Publishing Co., Inc., 2000.

[21] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[22] Joshua J Fenton, Stephen H Taplin, Patricia A Carney, Linn Abraham, Edward A Sickles, Carl D'Orsi, Eric A Berns, Gary Cutter, R Edward Hendrick, William E Barlow, et al. Influence of computer-aided detection on performance of screening mammography. *New England Journal of Medicine*, 356(14):1399–1409, 2007.

[23] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[24] K. Ganesan, U. R. Acharya, C. K. Chua, L. C. Min, K. T. Abraham, and K. Ng. Computer-aided breast cancer detection using mammograms: A review. *IEEE Reviews in Biomedical Engineering*, 6:77–98, 2013.

[25] Krzysztof J. Geras, Ritse M. Mann, and Linda Moy. Artificial intelligence for mammography and digital breast tomosynthesis: Current concepts and future perspectives. *Radiology*, 293(2):246–259, 2019. PMID: 31549948.

[26] Ezequiel Geremia, Bjoern Menze, Marcel Prastawa, Marc-André Weber, Antonio Criminisi, and Nicholas Ayache. Brain tumor cell density estimation from multimodal MR images based on a synthetic tumor growth model. 2012.

[27] Dimitris Glotsos, Jussi Tohka, Panagiota Ravazoula, Dionisis Cavouras, and George Nikiforidis. Automated diagnosis of brain tumours astrocytomas using probabilistic neural network clustering and support vector machines. *Int J Neural Syst*, 15(1):1–11, 2005.

[28] Gloria Gonella., Elisabetta Binaghi., Paola Nocera., and Cinzia Mordacchini. Semi-automatic segmentation of mri brain metastases combining support vector machine and morphological operators. In *Proceedings of the 11th International Joint Conference on Computational Intelligence - Volume 1: NCTA, (IJCCI 2019)*, pages 457–463. INSTICC, SciTePress, 2019.

[29] Gloria Gonella, Elisabetta Binaghi, Paola Nocera, and Cinzia Mordacchini. Semi-automatic segmentation of MRI brain metastases combining support vector machine and morphological operators. *Accept at International Conference on Neural Computation Theory and Application, 17-19 September 2019*, page 7, 2019.

[30] Gloria Gonella, Marco Paracchini, Elisabetta Binaghi, and Marco Marcon. Breast lesion detection from mammograms using deep convolutional neural networks. *Accept at International Conference on Pattern Recognition and Applications, 2-6 November 2020*, 2020.

[31] Nelly Gordillo, Eduard Montseny, and Pilar Sobrevilla. State of the art survey on MRI brain tumor segmentation. *Magn Reson Imaging*, 31(8):1426–1438, 2013.

[32] Harry S. Greenberg, William F. Chandler, and Howard M. Sandler. *Brain Tumors*. Contemporary Neurology Series. Oxford University Press, 1999.

[33] Endre Grøvik, Darvin Yi, Michael Iv, Elizabeth Tong, Daniel Rubin, and Greg Zaharchuk. Deep learning enables automatic detection and segmentation of brain metastases on multisequence MRI. *Journal of Magnetic Resonance Imaging*, 2019.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv:1512.03385 [cs]*, 2015.

[35] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[36] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[37] Elisee Ilunga Mbuyamba, Jorge Cruz-Duarte, Juan Avina-Cervantes, Carlos Rodrigo Correa-Cely, Dirk Lindner, and Claire Chalopin. Active contours driven by cuckoo search strategy for brain tumour images segmentation. *Expert Systems with Applications*, 56, 2016.

[38] Michael Jermyn, Kelvin Mok, Jeanne Mercier, Joannie Desroches, Julien Pichette, Karl Saint-Arnaud, Liane Bernstein, Marie-Christine Guiot, Kevin Petrecca, and Frederic Leblond. Intraoperative brain cancer detection with raman spectroscopy in humans. *Sci Transl Med*, 7(274):274ra19, 2015.

[39] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ 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.

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[42] Carmen Kut, Kaisorn L. Chaichana, Jiefeng Xi, Shaan M. Raza, Xiaobu Ye, Elliot R. McVeigh, Fausto J. Rodriguez, Alfredo Quinones-Hinojosa, and Xingde Li. Detection of human brain cancer infiltration ex vivo and in vivo using quantitative optical coherence tomography. *Sci Transl Med*, 7(292):292ra100, 2015.

[43] Rebecca Lee, Francisco Gimenez, Assaf Hoogi, Kanae Miyake, Mia Gorovoy, and Daniel Rubin. A curated mammography data set for use in computer-aided detection and diagnosis research. *Scientific Data*, 4:170177, 12 2017.

[44] Constance D Lehman, Robert D Wellman, Diana SM Buist, Karla Kerlikowske, Anna NA Tosteson, and Diana L Miglioretti. Diagnostic accuracy of digital screening mammography with and without computer-aided detection. *JAMA internal medicine*, 175(11):1828–1837, 2015.

[45] Y Li, H Chen, L Cao, and J Ma. A survey of computer-aided detection of breast cancer with mammography. *J Health Med Inf*, 4(7), 2016.

[46] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A. W. M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88, 2017.

[47] Li Liu, Jian Tao Wang, and Tianhui Wang. Breast and pectoral muscle contours detection based on goodness of fit measure. *2011 5th International Conference on Bioinformatics and Biomedical Engineering*, pages 1–4, 2011.

[48] Yan Liu, Strahinja Stojadinovic, Brian Hrycushko, Zabi Wardak, Steven Lau, Weiguo Lu, Yulong Yan, Steve B. Jiang, Xin Zhen, Robert Timmerman, Lucien Nedzi, and Xuejun Gu. A deep convolutional neural network-based automatic delineation strategy for multiple brain metastases stereotactic radiosurgery. *PLoS ONE*, 12(10):e0185844, 2017.

[49] Yan Liu, Strahinja Stojadinovic, Brian Hrycushko, Zabi Wardak, Weiguo Lu, Yulong Yan, Steve B. Jiang, Robert Timmerman, Ramzi Abdulrahman, Lucien Nedzi,

and Xuejun Gu. Automatic metastatic brain tumor segmentation for stereotactic radiosurgery applications. *Phys Med Biol*, 61(24):8440–8461, 2016.

[50] MA Guevara Lopez, N Posada, Daniel C Moura, Raúl Ramos Pollán, José M Franco Valiente, César Suárez Ortega, M Solar, Guillermo Diaz-Herrero, IMAP Ramos, J Loureiro, et al. Bcdr: a breast cancer digital repository. In *15th International conference on experimental mechanics*, 2012.

[51] Max Losch. Detection and segmentation of brain metastases with deep convolutional networks, 2015.

[52] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, Ç. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput. The multimodal brain tumor image segmentation benchmark (brats). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, 2015.

[53] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016.

[54] Inês C Moreira, Igor Amaral, Inês Domingues, António Cardoso, Maria Joao Cardoso, and Jaime S Cardoso. Inbreast: toward a full-field digital mammographic database. *Academic radiology*, 19(2):236–248, 2012.

[55] Feng Ning, Damien Delhomme, Yann LeCun, Fabio Piano, Léon Bottou, and Paolo Emilio Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371, 2005.

[56] Júlia E. E. Oliveira, Mark O. Gueld, Arnaldo de A. Araújo, Bastian Ott, and Thomas M. Deserno. Toward a standard reference database for computer-aided mammography. In Maryellen L. Giger and Nico Karssemeijer, editors, *Medical Imaging 2008: Computer-Aided Diagnosis*, volume 6915, pages 606 – 614. International Society for Optics and Photonics, SPIE, 2008.

[57] Kei Otsuka. 3-d deep learning : Lung tumor segmentation - file exchange - MATLAB central. https://www.mathworks.com/matlabcentral/fileexchange/71521-3-d-deep-learning-lung-tumor-segmentation (accessed on 14 July 2019).

[58] Valentina Pedoia, Sergio Balbi, and Elisabetta Binaghi. Fully automatic brain tumor segmentation by using competitive EM and graph cut. volume 9279, pages 568–578, 2015.

[59] U. Perez, E. Arana, and D. Moratal. Brain metastases detection algorithms in magnetic resonance imaging. *IEEE Latin America Transactions*, 14(3):1109–1114, 2016.

[60] Said Pertuz, German F Torres, Rulla Tamimi, and Joni Kamarainen. Open framework for mammography-based breast cancer risk assessment. In *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 1–4. IEEE, 2019.

[61] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999.

[62] S. Ruan, S. Lebonvallet, A. Merabet, and J. Constans. Tumor segmentation from a multispectral mri images by using support vector machine classification. In *2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1236–1239, 2007.

[63] Jainy Sachdeva, Vinod Kumar, Indra Gupta, Niranjan Khandelwal, and Chirag Kamal Ahuja. Segmentation, feature extraction, and multiclass brain tumor classification. *J Digit Imaging*, 26(6):1141–1150, 2013.

[64] Bernhard Schölkopf, Alexander J. Smola, Managing Director of the Max Planck Institute for Biological Cybernetics in Tubingen Germany Profe Bernhard Scholkopf, and Francis Bach. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002. Google-Books-ID: y8ORL3DWt4sC.

[65] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[66] Alexander Statnikov, Lily Wang, and Constantin F Aliferis. A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC Bioinformatics*, 9:319, 2008.

[67] Badri Narayan Subudhi, Veerakumar Thangaraj, Esakkirajan Sankaralingam, and Ashish Ghosh. Tumor or abnormality identification from magnetic resonance images using statistical region fusion based segmentation. *Magn Reson Imaging*, 34(9):1292–1304, 2016.

[68] J Suckling et al. The mammographic image analysis society digital mammogram database digital mammography ed ag gale, sm astley, dr dance and ay cairns, 1994.

[69] Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *arXiv:1707.03237 [cs]*, 10553:240–248, 2017.

[70] Richard Sutton. Two problems with back propagation and other steepest descent learning procedures for networks. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society, 1986*, pages 823–832, 1986.

[71] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995.

[72] Mihran Tuceryan and Arjun Jain. Texture analysis. *The Handbook of Pattern Recognition and Computer Vision*, pages 207–248, 02 1998.

[73] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2000. https://www.springer.com/gp/book/9780387987804 (accessed on 14 July 2019).

[74] Ragini Verma, Evangelia I. Zacharaki, Yangming Ou, Hongmin Cai, Sanjeev Chawla, Seung-Koo Lee, Elias R. Melhem, Ronald Wolf, and Christos Davatzikos. Multiparametric tissue characterization of brain neoplasms and their recurrence using pattern classification of MR images. *Acad Radiol*, 15(8):966–977, 2008.

[75] Christian Wachinger and Polina Golland. Atlas-based under-segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 315–322. Springer, 2014.

[76] Ka-Chun Wong, Junyi Chen, Jiao Zhang, Jiecong Lin, Shankai Yan, Shxiong Zhang, Xiangtao Li, Cheng Liang, Chengbin Peng, Qiuzhen Lin, Sam Kwong, and Jun Yu. Early cancer detection from multianalyte blood test results. *iScience*, 15:332–341, 2019.

[77] Darko Zikic, Ben Glocker, Ender Konukoglu, Antonio Criminisi, C. Demiralp, J. Shotton, O. M. Thomas, T. Das, R. Jena, and S. J. Price. Decision forests for tissue-specific segmentation of high-grade gliomas in multi-channel MR. *Med Image Comput Comput Assist Interv*, 15:369–376, 2012.