Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies

FIIT-182905-72174

Bc. Michal Kováčik

Organ Segmentation in 3D Medical Data Using Methods of Computer Vision

Master's thesis

Supervisor: doc. Ing. Vanda Benešová, PhD.

April 2019

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies

FIIT-182905-72174

Bc. Michal Kováčik

Organ Segmentation in 3D Medical Data Using Methods of Computer Vision

Master's thesis

Study programme: Intelligent Software Systems
Field of Study: 9.2.5 Software Engineering
Place: Department of Applied Informatics, FIIT STU Bratislava
Supervisor: doc. Ing. Vanda Benešová, PhD.

April 2019

STU FIIT
SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Zadanie diplomovej práce

Meno študenta: **Bc. Michal Kováčik**

Študijný program: Inteligentné softvérové systémy

Študijný odbor: Softvérové inžinierstvo – hlavný študijný odbor
Umelá inteligencia – vedľajší študijný odbor

Názov práce: **Segmentácia orgánov z trojrozmerných medicínskych dát metódami počítačového videnia**

Samostatnou výskumnou a vývojovou činnosťou v rámci predmetov Diplomový projekt I, II, III vypracujte diplomovú prácu na tému, vyjadrenú vyššie uvedeným názvom tak, aby ste dosiahli tieto ciele:

Všeobecný cieľ:

Vypracovaním diplomovej práce preukážte, ako ste si osvojili metódy a postupy riešenia relatívne rozsiahlych projektov, schopnosť samostatne a tvorivo riešiť zložité úlohy aj výskumného charakteru v súlade so súčasnými metódami a postupmi študovaného odboru využívanými v príslušnej oblasti a schopnosť samostatne, tvorivo a kriticky pristupovať k analýze možných riešení a k tvorbe modelov.

Špecifický cieľ:

Vytvorte riešenie zodpovedajúce návrhu textu zadania, ktorý je prílohou tohto zadania. Návrh bližšie opisuje tému vyjadrenú názvom. Tento opis je záväzný, má však rámcový charakter, aby vznikol dostatočný priestor pre Vašu tvorivosť.

Riaďte sa pokynmi Vášho vedúceho.

Pokiaľ v priebehu riešenia, opierajúc sa o hlbšie poznanie súčasného stavu v príslušnej oblasti, alebo o priebežné výsledky Vášho riešenia, alebo o iné závažné skutočnosti, dospejete spoločne s Vaším vedúcim k presvedčeniu, že niečo v texte zadania a/alebo v názve by sa malo zmeniť, navrhnite zmenu. Zmena je spravidla možná len pri dosiahnutí kontrolného bodu.

Miesto vypracovania: Ústav počítačového inžinierstva a aplikovanej informatiky, FIIT STU Bratislava

Vedúci práce: **doc. Ing. Vanda Benešová, PhD.**

Termíny odovzdania:

Podľa harmonogramu štúdia platného pre semester, v ktorom máte príslušný predmet (Diplomový projekt I, II, III) absolvovať podľa Vášho študijného plánu

Predmety odovzdania:

V každom predmete dokument podľa pokynov na www.fiit.stuba.sk v časti: home > Informácie o > štúdiu > harmonogram štúdia > diplomový projekt.

V Bratislave dňa 12. 2. 2018

SLOVENSKÁ TECHNICKÁ UNIVERZITA
V BRATISLAVE
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4
1

prof. Ing. Pavol Návrat, PhD.
riaditeľ Ústavu informatiky, informačných systémov
a softvérového inžinierstva

STU FIIT
SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Návrh zadania diplomovej práce

Finálna verzia do diplomovej práce [1]

Študent:

Meno, priezvisko, tituly: Michal Kováčik, Bc.
Študijný program: Inteligentné softvérové systémy
Kontakt: xkovacikm2@gmail.com

Výskumník:

Meno, priezvisko, tituly: Vanda Benešová, doc. Ing. PhD.

Projekt:

Názov:	Segmentácia orgánov z trojrozmerných medicínskych dát metódami počítačového videnia
Názov v angličtine:	Organ Segmentation in 3D Medical Data Using Methods of Computer Vision
Miesto vypracovania:	Ústav počítačového inžinierstva a aplikovanej informatiky, FIIT STU, Bratislava
Oblasť problematiky:	Počítačové videnie, Umelá inteligencia

Text návrhu zadania[2]

Spracovanie medicínskych radiologických dát je v súčasnej dobe veľmi aktívna oblasť výskumu, ktorá využíva rôzne metódy počítačového videnia. Jednou z významných problémových oblastí, dôležitých pre diagnostiku a posúdenie liečby v klinickej praxi, je spracovanie trojrozmerných dát z ultrazvukového snímača (US), magnetickej rezonancie (MRI) alebo počítačovej tomografie (CT). Kľúčové problémové oblasti v tomto prípade sú: registrácia anatomických orgánov, detekcia a segmentácia orgánov, anatomických anomálií, prípadne patologických zmien, napr. tumoru.

Analyzujte súčasný stav problematiky segmentácie orgánov v radiologických snímkach. Zamerajte sa predovšetkým na metódy využívajúce umelú inteligenciu. Navrhnite metódu na segmentáciu jedného konkrétneho typu anatomického orgánu, alebo orgánov anatomického regiónu (hlava, hrudný kôš, brušná dutina). Overte možnosti tradičných segmentačných prístupov, ako aj využitie neurónových sietí, prípadne ich vzájomnej kombinácie. Predpokladajte dvojrozmerné alebo trojrozmerné dáta z CT alebo MRI na vstupe spracovania. Metódu, ktorú ste navrhli realizujte softvérovým prototypom s využitím knižnice OpenCV, ITK alebo iných dostupných knižníc, ktoré sú vhodné na spracovanie a analýzu medicínskych dát.

Navrhnutú metódu overte experimentom s reálnymi dátami. Vyhodnoťte úspešnosť segmentácie, robustnosť metódy ako aj časovú výpočtovú náročnosť spracovania. Svoje výsledky porovnajte s relevantnými publikovanými riešeniami v tejto oblasti.

---

[1] Vytlačiť obojstranne na jeden list papiera

[2] 150-200 slov (1200-1700 znakov), ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia

Literatúra[3]

- HU, Peijun, et al. Automatic abdominal multi-organ segmentation using deep convolutional neural network and time-implicit level sets. International journal of computer assisted radiology and surgery, 2017, 12.3: 399-411.
- MOGHBEL, Mehrdad, et al. Review of liver segmentation and computer assisted detection/diagnosis methods in computed tomography. Artificial Intelligence Review, 2017, 1-41.

Vyššie je uvedený návrh diplomového projektu, ktorý vypracoval(a) Bc. Michal Kováčik, konzultoval(a) a osvojil(a) si ho doc. Ing. Vanda Benešová, PhD. a súhlasí, že bude takýto projekt viesť v prípade, že bude pridelený tomuto študentovi.

V Bratislave dňa 10.1.2018

Podpis študenta

Podpis výskumníka

Vyjadrenie garanta predmetov Diplomový projekt I, II, III

Návrh zadania schválený: áno / ~~nie~~[4]

Dňa: 12. 2. 2018

Podpis garanta predmetov

[3] 2 vedecké zdroje, každý v samostatnej rubrike a s údajmi zodpovedajúcimi bibliografickým odkazom podľa normy STN ISO 690, ktoré sa viažu k téme zadania a preukazujú výskumnú povahu problému a jeho aktuálnosť (uveďte všetky potrebné údaje na identifikáciu zdroja, pričom uprednostnite vedecké príspevky v časopisoch a medzinárodných konferenciách)

[4] Nehodiace sa prečiarknite

Anotácia

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
Študijný program: Inteligentné Softvérové Systémy
Autor: Bc. Michal Kováčik
Diplomová práca: Segmentácia orgánov z trojrozmerných medicínskych dát metódami počítačového videnia
Vedúci práce: doc. Ing. Vanda Benešová, PhD.
April 2019

Diplomová práca sa zaoberá vytvorením softvérového nástroja na automatickú segmentáciu orgánov v dutinách ľudského tela. Nástroj prijíma vstupy z 3D medicínskych snímkov získaných prístrojom CT. Cieľom práce je umožniť segmentáciu a automatické anotovanie veľkých datasetov, bez asistencie človeka, ako prípravu pre ďalšie spracúvanie špecializovnými algoritmami na vyhľadávanie patologických nálezov v konktrétnych orgánoch.

Dokument analyzuje prístupy na riešenie problémov tohoto typu ako aj existujúce riešenia. Venuje sa návrhu vlastného riešenia a overovaniu jeho výkonnosti. Vlastné riešenie využíva architektúry neurónových sietí, ako sú konvolučné neurónové siete. Vstupy z prístrojov normalizuje aby riešenie nebolo závislé od konkrétneho skenera a výstup z neurónovej siete je ešte ďalej spracúvaný, aby sa zvýšila presnosť výsledku.

Annotation

Slovak University of Technology Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree Course: Intelligent Software Systems
Author: Bc. Michal Kováčik
Master's thesis: Organ Segmentation in 3D Medical Data Using Methods of Computer Vision
Supervisor: doc. Ing. Vanda Benešová, PhD.
April 2019

Master's thesis project is concerned with creation of software tool for automatic organ segmentation in human body cavities. Tool accepts inputs from 3D medical images obtained by scanners like CT. Final goal is to allow segmentation an automatic anotation of large datasets, without need for human asistance, as preprocessing for specialized algorithms for detection of patalogical deformities in specific organs.

Document analyzes methods for similar segmentation problems as well as existing solutions. It proposes own solution and evaluates it's performance. This solution is making use of neural networks architectures, like convolutional neural networks. Inputs from scanners are normalised to provide independence from scanner manufacturers. Outputs are further proccessed by Computer Vision methods, to further improve segmentation results.

ACKNOWLEDGMENTS

It is with immense gratitude that I acknowledge the support and help of my supervisor, Ms Vanda Benešová for her guidance, involvement and experience throughout this work.

DECLARATION
I hereby declare, that I wrote this master's thesis independently, with the help of literature listed at the end of this document and consultations with my supervisor.

.....
Bc. Michal Kováčik

Contents

List of Figures

Chapter 1

Introduction to organ segmentation

Detection and determining exact border of organs from medical images (CT, MRI) is important for diagnostics as well as during operation preparation [1].

Organ segmentation for cavities, such as abdomen or thorax is a challenging task. Organs like kidneys, lungs, liver, spleen or pankreas are especially daunting, because they are very well repleted with blood which causes irregularities in volumetric data and blurs texture. Patient movements, like breathing, as well as different scanners settings and irregular tissue composition are also making things more difficult [2].

One of motivations for automatic organ segmentation is saving time and cognitive load of experts. Detection of primary or secondary tumors, eg. in liver, yields best results using semi-automatic interactive method, that is very time demanding. Radiologists have to manually, slice by slice, mark organ boundaries. Reason for widespread use of interactive segmentation methods is that even though fully automatic deep learning methods have reached state of the art performance on challenges, it is not sufficiently accurate, nor robuts enough for clinical use, because of differences between scanners and patients alike [3].

Similar issue is aiming a ray of ionizing radiation during radiation therapy of oncological diseases. To precisely aim the ray, without damaging any important tissue and minimizing collateral damage to healthy regions, precise boundaries of organs and tissues is required. Multiple trajectories needs to be calculated, to avoid overexponation of single healthy area. This trajectories are often determined with genetic algorithms, which fit functions needs as much information as possible, for correct evaluation of possibilities [4].

Ultimate goal for our effort is to provide automatic organ segmentation method for unlabeled data so organ specific computer vision operations could be applied, like detection and segmentation of pathological anomalies in concrete organs. However, even if we only reduce time needed for human assistance for such a task, it would be a huge help for experts.

Chapter 2

Analysis

2.1 Capturing patients volume by detectors

There are many methods to capture and visualize volume of patient. By method they utilize, they can be divided into categories [5]:

- ultrasound (doppler) methods:
  - angiography, sonography, echokardiography
- x-ray based methods
  - skiascopy, classical tomography, Computerized Tomography(CT)
- nuclear based methods
  - gamagraphy, Positron Emission Tomography(PET)
- magnetic based methods
  - Nuclear Magnetic Resonance Imaging (NMRI)

This paper focuses mainly on segmentation of volumentric data obtained from CT and (N)MRI. To understand how some of the segmentation methods work it is essential to understand basic principles of aforementioned machines.

2.1.1 Modus operandi of Computerized Tomography

The idea behind CT scanner was proposed by Godfrey Hounsfield. Since a lot of information is lost during traditional x-ray process, simply because 3D body of patient is projected to a 2D photographic paper. He proposed, that with multiple measurements across multiple angles, it should be possible to distinguish between various soft tissues formations [6].

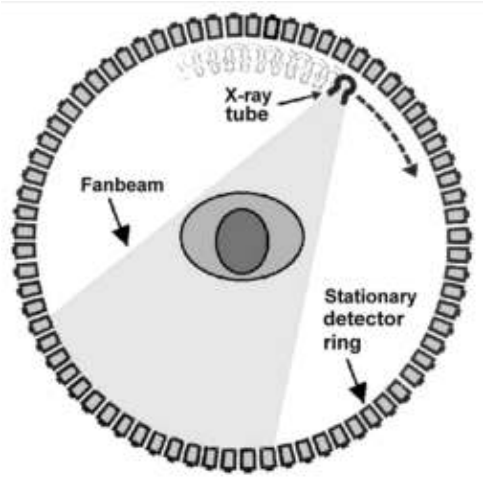


Figure 2.1: Detector ring for 4. generation of CT scanner [6].

Modern approach, called helical scanning, is a process when scanner gantry is made of stationary detector ring with X-ray tube constantly moving around the gantry in circles and patient is smoothly drawn through on moving table. This allows rapid scans for Region of Interest(ROI).

This concept is allows to control thickness of slices captured by regulation table movement as well as tube rotation. So if there is overlap of beams, then $thickness = \frac{movement}{rotations}$. However some overlap is usually present, to increase accuracy of measurements.

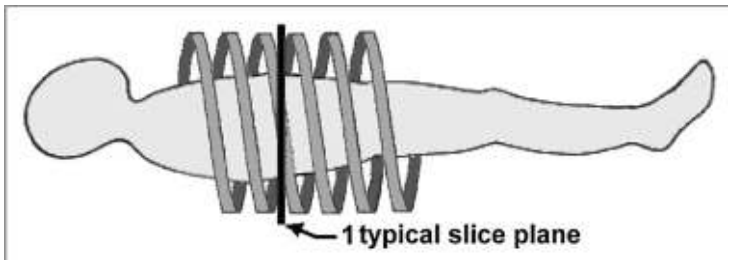


Figure 2.2: Helical CT spiral scan [6].

Density of voxels measurements

What detectors of CT machine measure, is absorption of radiation, which is difference between amount of radiation produced by X-ray tube and value detected by sensors. This absorption can be written as sum of partial absorptions by each voxel in path of beam. Since voxel size is known, and beams trajectories are also known, it is possible to determine attenuation coefficient μ_i of each voxel [6].

Therefore if N_0 is intensity of beam entering first voxel in beam's trajectory, and w_1 is width of voxel and μ_1 is attenuation, then intensity of beam leaving the voxel is:

$$N_1 = N_0 e^{-w_1 \mu_1}$$

Figure 2.3: Attenuation formula for single voxel: N_1 - new intensity after crossing voxel, N_0 - original attenuation, w_1 - width of the voxel μ_1 - attenuation coefficient

This can be chained for multiple voxels in path of beam resulting in:

$$N_i = N_0 e^{\Pi_{i=1}^{n} w_i \mu_i}$$

Figure 2.4: Attenuation formula for i^{th} voxel: N_i - new intensity after crossing i voxels, N_0 - original attenuation, w_i - width of the voxel i μ_i - attenuation coefficient of voxel i

When expression is divided by N_0 and *logaritmus naturalis* is applied, simple linear equations are acquired.

$$-ln(\frac{N_i}{N_0}) = \sum_{i=1}^{n} w_i \mu_i$$

Figure 2.5: Linear attenuation formula: N_i - new intensity after crossing i voxels, N_0 - original attenuation, w_i - width of the voxel i μ_i - attenuation coefficient of voxel i

These attenuation measurements are then normalised to CT number value, in Hounsfield units. It is calculated as:

$$HU = \frac{\mu_{voxel} - \mu_{water}}{\mu_{water}} \cdot 1000$$

Figure 2.6: Hounsfield units conversion formula: μ_{water} - calibration measurement provided by manufacturer, μ_{voxel} - value measured per voxel

2.1.2 Modus operandi of Magnetic Resonance Imaging

Magnetic resonance imaging uses physical phenomenon called Nuclear Magnetic Resonance. Since people were scared of the word nuclear, this word was removed and this method is now simple known as MRI. Since it is not using ionizing radiation, it is considered to be safer than CT [7].

Nucleus of atom is composed of neutrons and protons that all rotate around own axis in motion called spin. Since protons wield positive charge, its rotation produces magnetic field and *magnetic moment*. Magnetic moment is characterizing magnetic dipole.

In electrostatic field positively charged nucleus will negatively charged electron move in orbits, thus creates current loop equivalent to magnetic dipole. This creates *magnetic moment* for electron.

Nuclei with even nucleon number do not affect surroundings magneticaly, because their magnetic moments cancell each other out. However, atoms with odd nucleon numbers keep their magnetic moments. This is mostly hydrogen ^1H, which has only 1 proton and has relatively large magnetic moment. Since there are 2 atoms of hydrogen in single molecule of water, and human body consist mostly of water, it is the best candidate for MR imaging [7].

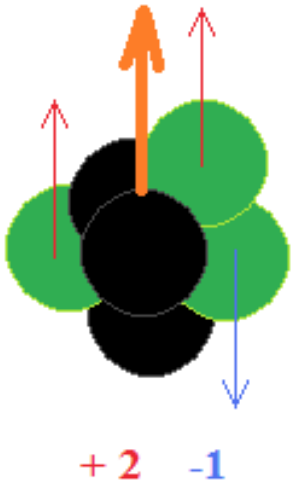


Figure 2.7: Magnetic moment of atom with odd number of protons. (from `https://www.wikiskripta.eu`)

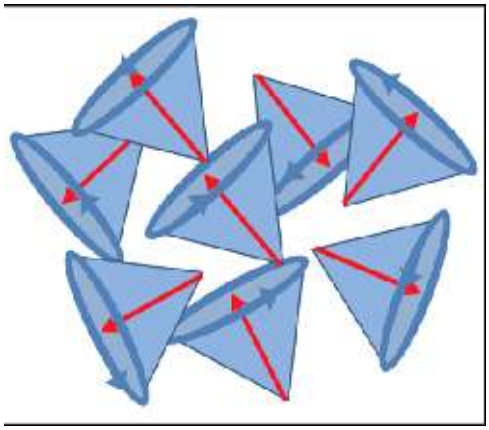
When nucleus is put in strong magnetic field, rotation axes of protons are turned to by parallel with line of force of outer magnetic field. Most of them is in position, where magnetic moment is oriented in agreement with vector of external magnetic field, because oposite direction is energy expensive.

Not only do protons have spin, they also move on the surface of virtual cone. Frequency of this movement is **Larmor frequency**. This frequency is affected by:

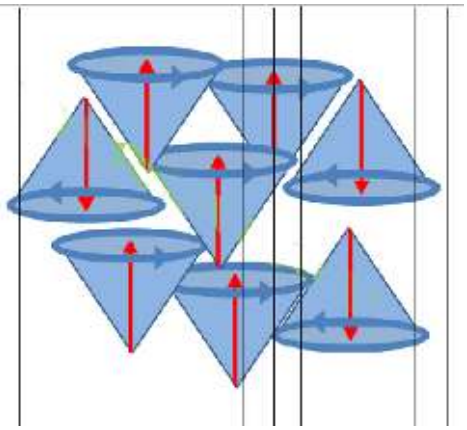
1. intensity of outer magnetic field
2. type of nucleus (constant)

For hydrogen ^1H in magnetic field $B = 1.5T$ the frequency $f \sim 64MHz$

When atoms of hydrogen are oriented in parallel with external field, and they are on the same Larmor frequency, they start to resonate. This resonation is measurable by detectors. When external magnetic



Dipoles with randomly oriented magnetic moments. (from `https://www.wikiskripta.eu`)



Dipoles with oriented magnetic moments by external magnetic field (from `https://www.wikiskripta.eu`)

field is removed, this resonation slowly starts to decay. Time when this resonations reaches 67% of its value is called **T1** and when it reaches 37%, then its called **T2**.

Time periods of T1 and T2 are different for various tissues, as well as different values of external magnetic field (e.g. in table 2.1).

Table 2.1: Values of T1 and T2 for various tissues in external magnetic field 3T.

Tissue	T1(ms)	T2(ms)
grey matter	1200	80
white matter	800	70
liquor cerebrospinalis	4000	600
arterial blood	1700	120
veins blood	1500	40

2.2 Challenges of organ segmentation

Organs in cavities like thorax or abdomen, or even in pelvic cavity can be deformed, or shifted in random directions. This is caused e.g. by breathing, cardiac activity, or simply by the volume of bladder or stomach. It is also important to account for uniqueness of each human being, as we all have slightly different anatomic structure, and come in all shapes and sizes.

Even greater problem than shape is volume density captured by detectors, like CT. Since soft tissues are very similar in density, and organs like spleen or liver are very well saturated with blood, it is very hard to decide exact boundaries of certain organs, without using methods like contrast enhancement techniques, like injection of contrast solution to *vena portae* [1].

However the most apparent challenge is lack of expert labeled data available, which prevents creation of general model, that could cope with all aforementioned challenges.

2.3 Traditional segmentation methods

Many segmementation methods exists, some of them are more specialized, because they use huge amount of information about the shape of image. The others tend to be more generally purposed [8].

Medical images are grayscaled images with usual dynamic range of 16 bits (depending on a machine, could be more or less for older machines), where gray level represents measured values of the scanner.

For CT that is radiation dampening and for MRI it is time of normalisation. However for computer vision purposes, we can threat them as if they were regular grayscale images.

2.3.1 Thresholding

Thresholding is method, that relies on assumption, that foreground object can be separated from the background solemnly by brightness. As mentioned earlier, tissues such as bones, or muscles have all different density captured by CT and measured in Haunsfield Units. This density level is projected as brightness therefore it is possible to use thresholding as a method [9].

Simples and most general method method is **Global thresholding**. This method creates a binary mask of foreground simply by setting all values of pixels (or voxels if 3D image is processed) below certain threshold(θ) to 0.

$$g(x) = \begin{cases} 1 & f(x) \geq \theta \\ 0 & f(x) < \theta \end{cases}$$

Figure 2.8: Global thresholding function: θ - threshold value, f(x) - pixel/voxel intensity

To select proper thresholding value, we can use prior knowledge of densities of tissues in Hounsfield units(HU). Some densities of certain tissues are shown in table 2.2.

Table 2.2: Hounsfield unit values for certain materials and tissues [10].

Material	HU value
Air	- 1000
Lung	- 700
Fat	- 100
Water	0
Kidney	20
White matter	20
Grey matter	37
Muscle	40
Blood	40
Liver	60
Bone	$>$ 400

Segmentation of certain organ means, that for correct mask we need to cut off not only values below threshold but also above it. Therefore the formula for such task requires a slight modification to:

$$g(x) = \begin{cases} 1 & \theta_1 < f(x) \leq \theta_2 \\ 0 & else \end{cases}$$

Figure 2.9: Dual cut global thresholding function: θ_i - threshold value, f(x) - pixel/voxel intensity

As standalone, this methods results are too crude, and contain a lot of artifacts. However if used in combination for some other methods, such as active contour, or region growing, it can provide seeds to initialize them without requiring any interaction from user's side.

2.3.2 Region growing

Region growing is based on assumption, that pixels (or voxels) that belongs to single organ are connected and similar. What similar means is system specific, and needs to configured for each different segmentation usage. That units are connected usually means that they are spatialy nearing the object. To evalue connectivity, neighbouring function is used. If neighbouring pixels are as similar, they are considered part of the object [9].

For region to start growing, method requires starting point called *seed*. This can be provided by user interactively. This attitude however renders method to be only semi-automatic. There is a way to automatically select the seed points, picking pixels(voxels) with lowest gradient, which means their surroundings is rather homogenous [11]. This approach tends to oversegment.

There are several neighbouring functions, that define object neighbourhood. Those used in common are 4-connected neighbourhood and 8-connected neighbourhood for 2D that can be easily extrapolated into 3D by just adding another dimension.

Defining similarity can again be done user, however unlike selecting seed, this task is not intuitive and will require some trial and error, which can be frustrating and time demanding. To automatically derive similarity criterion Pohle and Toennies [11] proposed a method that does so autonomously. They defined similarity ass likelihood of belonging to a gaussian distribution of gray values with mean being the absorption level in CT and magnetization in MR. Standard deviation accounts for the noise.

2.3.3 Active contour

Active contour algorithm works with minimizing the energy of input contours. In 2D space contours are ordered sets of tuples of coordinates $c_i = [x_i, y_i]$, forming a line. The energy of the contour, algorithm is minimizing, is defined as sum of external and internal energies of each its component [12].

$$E_c = \sum_i (E_{internal}(c, i) + E_{external}(x_i, y_i))$$

Figure 2.10: Active contour optimization formula: E_c - total energy, $E_{internal}$ – $internal energy of input contours$,$E_{external}$ – $external energy of contour point$

Internal energy of contour is computed from shape of the contour, and can be controled by parameters, such as tension, or rigidity. This allows to prefer smooth contours over those containing sharp edges.

External energy can be described as metric of interesting feature in image, where less interesting features have a higher energy and vice versa. Python scikit image library[1] uses negative magnitude of image gradient obtained by convolution with Sobel filter. Such implementation attracts contours toward the edges.

[1] `https://github.com/scikit-image/scikit-image/blob/master/skimage/segmentation/active_contour_model.py`

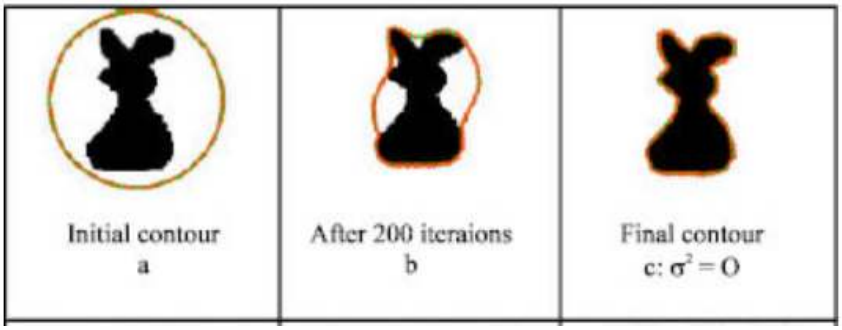


Figure 2.11: Depiction of active contour algorithm. Leftmost image is initial state, middle one depicts state after 200 iterations of energy minimization and right one is final segmentation result [13].

Solution is then found by iteratively moving parts of the contours in way that decreases the total energy of the contour over time. Same approach as optimizing weights in simple neural networks is used called *gradient descent*. Each iteration steps in negative gradient with maximum step allowed as parameter calculates the force to apply on each point of contour.

$$\overline{c}_{i,t+1} = \overline{c}_{i,t} + F(\overline{c}_{i,t})$$

Figure 2.12: Active contour optimization step: $\overline{c}_{i,t+1}$ - new position of contour point, $\overline{c}_{i,t}$ - old position of contour point, $F(\overline{c}_{i,t})$ - force to apply to contour point

2.3.4 Watershed algorithm

Watershed algorithm treats image as topographic map. Therefore each pixels (voxels) value is like height of terrain. Imagine that rain starts over such terrain. As drop of water hits points on the ground, there are 3 possible outcomes [12]:

1. water stays where it dropped (point of local minimum)
2. water will move downhill in one single way (catchment basin)
3. water will split and move downhill multiple ways (divide lines)

As the time flows and more and more water is added, catchment basins will be flooded and eventually, only the divide lines will remain. Divide lines will never be flooded, because they raise with the water level. When highest point of map is reached, flood stops and remaining land is considered a boundary between objects.

This analogy is easily explained for 2D images, but to implement it in 3D only means broadening of neighbourhood of pixel in another dimension and algorithm works without any modifications required.

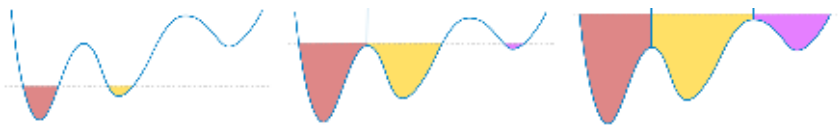


Figure 2.13: Watershed segmentation schema. Every point, that is not part of divide line, is considered part of catchment basin eventually.

Watershed algorithm tends to oversegment input picture. To avoid such situation, it is possible to use some preprocessing morphological operations, or use merging of adjacent regions, if they are similar [9].

Resulting contours are guaranteed to be closed unlike in active contours algorithm.

2.3.5 Statistical shape models

Many organs in human body have roughly similar shape and position with only a little variations. These organs can be modeled to a *statistical shape model* from dataset of already segmented images. Segmentation is then performed as fitting model onto input data [14].

Method is more complex, then those discussed above, however it is much more robust to noise and image artifacts [15]. Model produced does not only contain mean shape but also it's variations. This mean is computed from *point distribution model* which is a set of points located on same anatomical positions for each medical scan.

Let's put point distribution model into a single matrix where $\vec{m}_i$ is single row of coordinates: $\vec{m}_i = (x_i, y_i, z_i)$ then computing mean shape is as simple as:

$$\overline{m} = \frac{1}{N} \sum_{i=1}^{N} \vec{m}_i$$

Figure 2.14: Statistical shape model formula: $\overline{m}$ - ssm model, $\vec{m}_i$ - single row of coordinates

Shape variations can be extracted using principal component analysis and then storing computed eigenvalues of variations. Valid shapes can be then computed from stored mean value by adding shape parameter and eigenvalue.

Unlike atlases, statistical shape models use only shape information for creating model, whereas atlases use information like tissue distribution and aditional anatomical knowledge, like position of other organs [12].

2.3.6 Graph cut

Graph cut is considered useful multidimensional optimization algorithm, that can, unlike active contour, preserve sharp edges if necessary [16]. This algorithm is designed to capture hypersurfaces of N-dimensional graphs, therefore using it on 3D medical image is only matter of correct setup.

Consider 3D image, where each voxel is grap node forming set V and each voxel has connection to its neighbouring voxels, forming set E. Image can then be represented as graph $G = V, E$. Except for their neighbours, each node has connection to special *terminal nodes* call **source** (s) and **sink** (t).

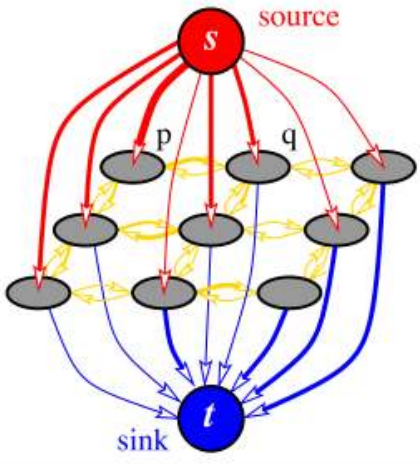


Figure 2.15: Graph construction [16].

Edges are weighted so that voxels of the background have small weight for one of the terminal nodes and large for the other. Weights for the edges between voxels of the foreground are treated in opposite

fashion. Similar pixel have their weights higher, than distinct ones.

With graph constructed, it is now possible to use minimum cut algorithm, that will divide graph into 2 parts, removing edges in such fashion, that the sum of weights of removed edges is minimal [12]. OpenCV implementation builds graph using 4-neighbourhood of pixels and than uses Edmonds-Karp algorithm to find minimum cut [2].

Separating image using mincut provides only binary segmentation. In addition, this algorithm, like region growing, requires *seeds* of background and foreground, therefore requires interaction.

2.3.7 Atlas based segmentation

Atlas based segmentation transforms segmentation problem into registration problem. Algorithm maps every voxel of input image onto the atlas. The atlas is exemplary image, that has been segmented, and is considered *ground truth*. After mapping is done, each voxel of input image is assigned class of atlas one it is mapped to [12].

There are 2 basic approaches for image registration:

- **Intensity-based registration** measures how well are images mapped to one another with *mutual information*. It is based on expectation, that regions mapped one to another have similar intensity distribution [12].
- **Feature-based registration** is minimizing difference between 2 sets of points iteratively. Similarly to statistical shape models, it requires same anatomic points as input. Algorithm consists of 2 steps:
  1. each point in 1^{th} set is assigned to the closest point in 2^{nd} set
  2. these pairs are then used to calculate transformation to move points closer to each other.

2.4 Machine learning based segmentation methods

Machine learning algorithms are usually divided into 2 categories:

1. **supervised learning** or learning with teacher. This algorithm receives dataset $D = \{x, y\}_{n=1}^{N}$, where x is vector of features and y is expected output. Then algorithms tries to find parameters of model Θ so that the loss function is minimized. Loss function is defined as $L(t, y)$, where t is correct solution and y is output of model function $f(x, \Theta)$ [17].
2. **unsupervised learning** algorithms process data without expected outputs provided. They train to find patterns in the dataset or reduce dimensions of the data. Main usecases are Principal Component Analysis, clustering or associations.

Neural networks are foundation for most methods of machine learning and deep learning. They are composed of many neurons with *net* activation function and parameters Θ, that are set of weights for input of the neuron. Activation function is elementwise nonlinearity function, such as ReLu or sigmoid, applied on result of matrix multiplication of vector of weights and inputs. Bias b is added to *net* if necessary.

$$net = w^T \cdot x + b$$

Figure 2.16: Neural network activation formula: net - activation function, w - weights of neuron, x - neuron input, b - bias

When connected in parallel, this neurons form a layer. This layers can be connected in series then they form **deep** network.

Convolutional neural network has shared weigths across network. That means that model has no need to learn again and again detectors for same object, which has many occurencies in the same picture. This also reduces number of parameters to set. On each layer, input image is convolved with set

[2] `https://github.com/rajatsaxena/OpenCV/blob/master/graphcut.py`

of kernels W and biases B, and each convolution generates new set of features. This process is repeated for each layer.

$$X_k = W_{k-1} * X_{k-1} + b_{k-1}$$

Figure 2.17: CNN features computation formula: X - map of features, k - layer, W - weights, b - bias

Recurrent neural networks in addition store state of the network h in time t, that was produced as result of previous state h_{t-1} and input x_t:

$$h_t = W \cdot x_t + R \cdot h_{t-1} + b$$

Figure 2.18: Recurrent neural network formula: h - state of network, t - time, x - input, W - weights, R - immutable weights, b - bias

where matrices of weights W and R are immutable. Networks build on these architectures show a lot of promise in medical applications [18].

2.4.1 Hierarchical 3D Fully Convolutional Network (FCN)

Traditional FCN if trained for segmentation of multiple organs, will have many neurons trained only for differentiating foreground and background of volume, in order to reduce loss function. This network is capable of segmentation, however it will be very innacurate for smaller organs. To cope with the problem, it is possible to add another FCN and let it fine-tune the results from the first one [19].

To further reduce the number of voxels, network needs to anotate, it is possible to remove 50-60% of volume by simple thresholding, keeping only the volume of body. This greatly reduced space is then used as an input for the first stage FCN that will, instead of segmenting concrete organs, anotate the smallest possible candidate region, where organs should be located [19].

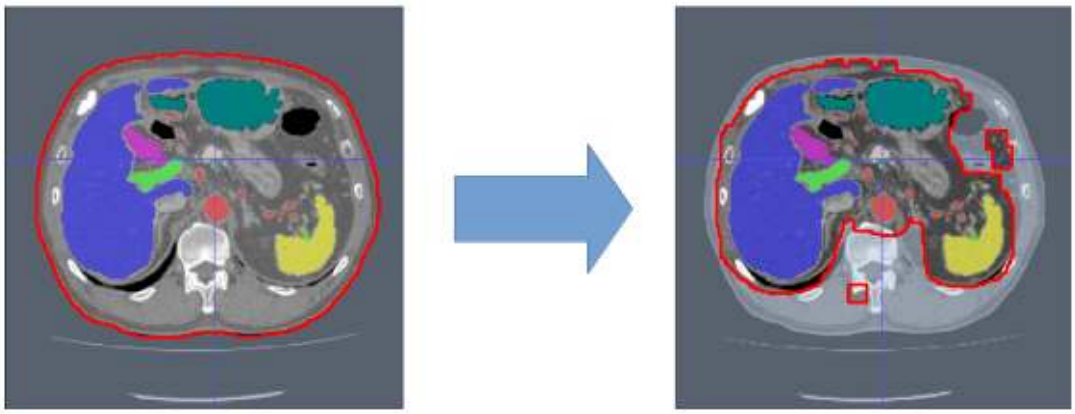


Figure 2.19: Input and output of first-stage FCN in hierarchy. Red are labeled candidate regions for organs location [19].

Many FCN architectures can be utilized to be connected in hierarchical fashion, Roth [19] used U-Net (very similar to V-Net) architecture. Using this fine-tuning approach with 2 stages outperformed 2D FCN architectures, as well as traditional, clinically used methods, such as atlas based registration.

2.4.2 V-Net

V-Net is an instance of fully convolutional network trained end-to-end, that has been modified not to process volumetric data slicewise, but uses volumetric convolution. This means that convolution kernels are not square but cubic instead [20].

Network is divided in 2 parts, in picture 2.20 it's displayed as left and right. Left part from top to bottom takes care of feature extraction with set of kernels and then compression of volume, so lower layers, can examine higher level features. Since each layer does compression, they all work with different resolutions [20].

Compression is performed with convolution again. To halve the volume of sample, one can use e.g. 2x2x2 kernel with stride 2. This approach removes need for max-pooling layers, since convolution is used instead, which means, that to map output back to inputs in training phase, using simple de-convolution is enough. This means smaller memory footprint [20].

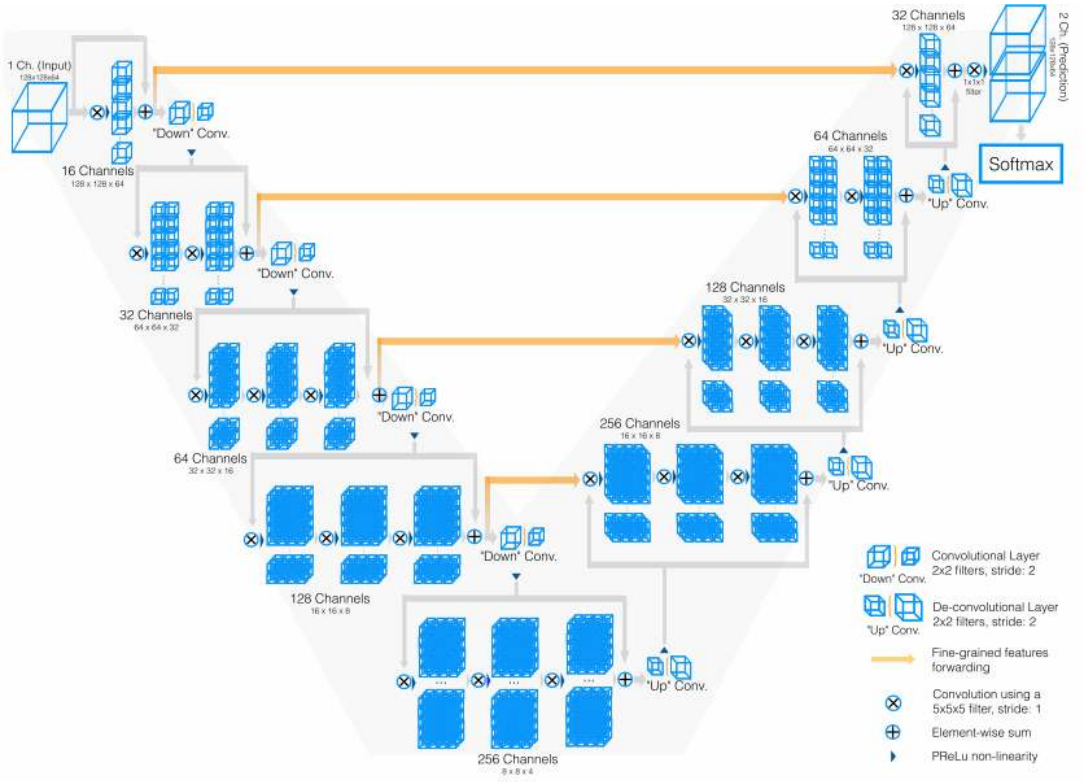


Figure 2.20: V-Net schema. Cubes represent 3D convolution kernels [20].

Remaining half of network uses features gathered from their counterparts at equivalent layer in V model. Together with feature maps from lower layers of the right side, it compiles information for volumetric segmentation. As opposite to left side, after each layer deconvolution is applied, to decompress volume, doubling it's size [20].

Last convolutional layer outputs same volume as was put in with probabilities of each voxel to belong to background or foreground. This is achieved by applying soft-max on each voxel.

This type of network for prostate segmentation task achieved same score as best performing model for same challenge, based on statistical shape models. They both reached on average 0.87 Dice score [3].

[3] `https://promise12.grand-challenge.org/results/`

2.4.3 Voxelwise Residual Network (VoxResNet)

Deep residual networks, use so called skip connections to propagate signal directly across the blocks of layers, by-passing non-linear transformations. Therefore input for *i-th* layer would look like $x_i = H_i(x_{i-1}) + x_{i-1}$, where H_i is nonlinear transformation. This allows gradient to flow directly from back to front using identity function [21].

VoxResNet is composed of stacked residual modules (image 2.21). Each module performs addition of input and transformed features before output. Only small convolutional kernels are used (3x3x3), because they have been observed to yield the best results. Since convolution is performed with stride 2, downsampling is also performed, enabling to capture more context [22].

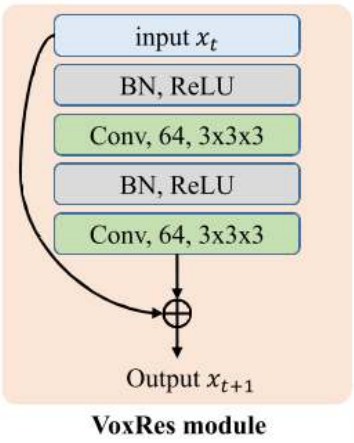


Figure 2.21: VoxRes Module schema [22].

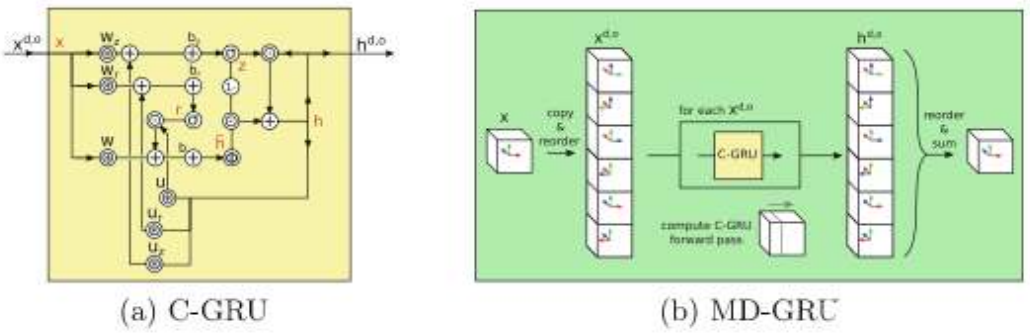
To allow capturing and using high level contextual information, it is possible to train one instance of VoxResNet on subvolumes, and then use *propability maps* generated by it in second instance with full volumes of data. This way it is possible to refine segmentation and to remove the outliers [22].

2.4.4 Multi-dimensional Gated Recurrent Units (MD-GRU)

Gated Recurrent Unit (GRU) can be viewed as simplified version of LSTM which, instead of input and forget gates, uses only update gate and combines hidden and cell state. Combining those states means lower memory requirements, therefore much larger inputs can be provided as well as much larger networks can be designed [23].

Traditional GRU is defined as calculating the *reset gate* and *update gate* using values from previous state and then calculating value of new state of layer using output from previous layers as well as values of updated gates. To be able to process 3D volumetric data, as well as convolution, GRU can be modified to Convolutional GRU (C-GRU) [23].

MD-GRU is composed of data dimensionality times C-GRUs. And since one is required for each direction, total count of C-GRUs is times 2.



(a) C-GRU (b) MD-GRU

Figure 2.22: (a) Schema of C-GRU computations (b) MD-GRU for 3D data, composed of 6 C-GRUs [23].

MD-GRU on MRBrainS challenge outperformed models like LSTM or U-Net reaching rank 7 in challenge. There are better performing models, however none of them has yet available articles to analyze [4].

2.4.5 U-Net

Similar to V-net, U-net is extension of fully convolutional network, but instead of always downsampling the input, here max-pooling is replaced by upsampling, which results in higher resolution of result. This means, network is split to 2 parts:

- **Contracting part** is doing convolution followed by ReLu and max-pooling with downsampling as classic convolutional network. Every downsampling also doubles feature channels.
- **Expansive part** is, contrary to contracting, doing up-sampling and up-convolution, that halves the feature channels.

However, feature map always has to be cropped, due to convolutions losing the border pixels in every step [24].

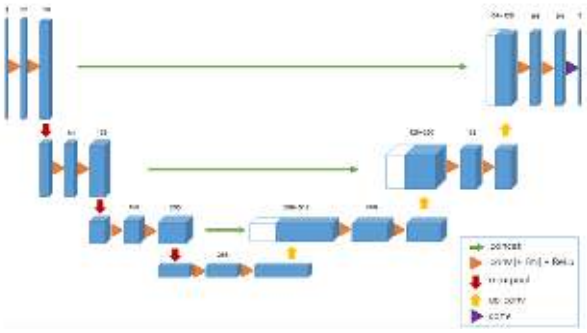


Figure 2.23: The 3D-Unet architecture [25].

Aforementioned model works with 2D data. However this approach can be easily scaled from 2D to 3D, by adding additional dimension to each operation. This means, that 3x3 convolution will become 3x3x3, ReLu is performed on each cell of matrix, so it does not need any scaling and 2x2 max-pooling with stride 2 will become 2x2x2 with stride 2 in all dimensions. Passing features from contracting to equal expansive path [25].

[4] `http://mrbrains13.isi.uu.nl/results.php`

2.5 Data preprocessing

Almost all described methods in chapter use some kind of preprocessing method. Most common method is *cropping volumes* to reduce memory as well as computational burden, significantly speeding up the training process. Simple method as applying thresholding and applying resulting binary mask, may reduce space by 40%.

Cropping as much as 40% of total volume from background, will also heavily change class balance, because relevant organs will have much higher relative volume, compared to background, that could cause machine learning methods to focus on background instead of organs.

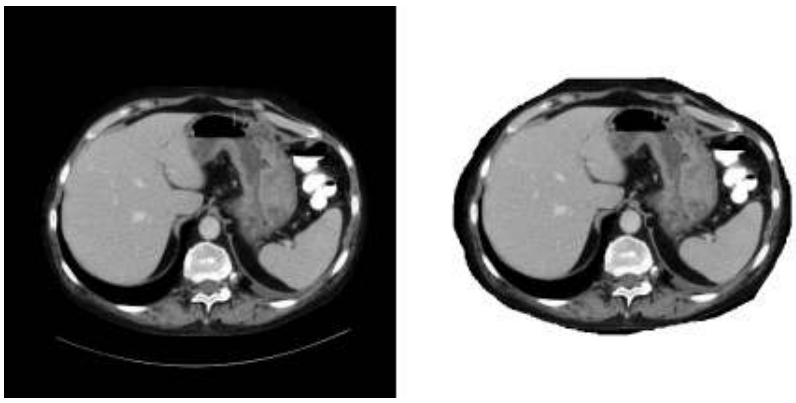


Figure 2.24: Cropping volume of CT to reduce processing requirements.

CT images in DICOM format are not guaranteed to be in Haunsfield units, since not every scanner has the same properties. To convert into then, it is possible to extract from DICOM metadata values of RescaleSlope and RescaleIntercept and transform them with linear transformation to Haunsfield units. This allows using known density values of tissues as additional information.

Data from scanners tends to be very noisy, because granularity of detectors is finite. To cope with that, it is possible to use filters such as mean filter or Gaussian filter, to supress the noise influence.

To enhance edges in ify areas of soft tissues, one can use e.g subtraction of Gaussian smoothed image [22]. Doing may reduce training of feature detectors of CNN, or improve results of traditional algorithms.

Creating supervoxels, or superpixel per slices, using Self Organizing Maps or algorithms like SLIC, may be useful to reduce memory and time requirements for some algorithms like graph cut, since it has to deal with significantly lower number of nodes and edges.

2.6 Related work

2.6.1 U-Net Convolutional Networks for Biomedical Image Segmentation

Ronneberger et al.[24] improved upon "fully convolutional network" so that it requires very few training images and yields very good segmentation results. Their idea was to supplement contracting layers with expanding layers and therefore to replace max-pooling layers with upsamling.

Upsampling using high resolution features, from complementary contracting layer, means that localisation and use of context without compromising trade-off is possible. This "by-passes" between contracting and expanding layers create U-shaped architecture.

To further battle the small dataset problem, they employed elastic deformations to training images, that help network to deal with deformations, even if not originaly present in the dataset. This include shift and rotation and grey value variations.

This method scored the best on EM segmentation challenge, using serial section of electron microscopy of the drosophila ventral nerve, and on ISBI cell tracking challenge, focusing on glioblastoma.

2.6.2 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation

Çiçek et al.[25] proposed a method, that allows to train 3D Unet using annotation, that are not present on each slice of 3D data, but rather much more sparsely. This network can not only be used to autonomous segmentation, but also to densify sparse annotation. This means, that radiologist needs to produce much less annotations, skipping multiple slices, saving time.

This method uses 3D volumes as input data, providing much more context information for network to use. It gained huge performance gain to equivalent 2D implementation, for task of fully-automated segmentation.

Chapter 3

Dataset description

Dataset used for segmentation in this work is provided by AAPM Thoracic Auto-segmentation Challenge. It consists of 36 patients scans obtained in 3 different studies per 12 patients. Dataset is provided in standard DICOM format, and are manually anotated by radiologist. These annotations will serve as the ground truth for further evaluation of proposed method.

All images are from CT, however metadata attached is inconsistent and confusing, claiming that some images are from PET scan and some are from MRI. Most images are from healthy patients, however there are some abnormalities, like collapsed lung in some cases (image 3.1).

Since challenge focuses on segmentation of organs of thorax, only volumes of thorax are provided, usually ranging from C2 to L2 vertebra. It's dimensions are 512x512x140 where pixel spacing in transversal plane is rectangular grid with side of 0.976mm and slice thickness of 3mm.

Figure 3.1: Study 3 patient 12 - collapsed right lung in transversal, frontal, sagittal slices and 3D render.

Each patient consists of N `.dcm` files, where N is number of transversal slices, numbered from 0 to N-1. These files contains metadata about patient position during examination, slice position in volume in cranio-caudal direction, orientation information (left and right side).

Train dataset has one additional `.dcm` file, that contains manually drawn annotations for 5 organs: heart, left lung, right lung, spinal cord and esophagus. These annotations are represented as series of points forming polygons, for each cranio-caudal slice.

Almost every patient's annotation from dataset contains some sort of error, such as ordering mismatch for polygon points, or blank spaces where there should be none. These are mostly visible when viewing volumes in different slice rotation than transversal (image 3.2).

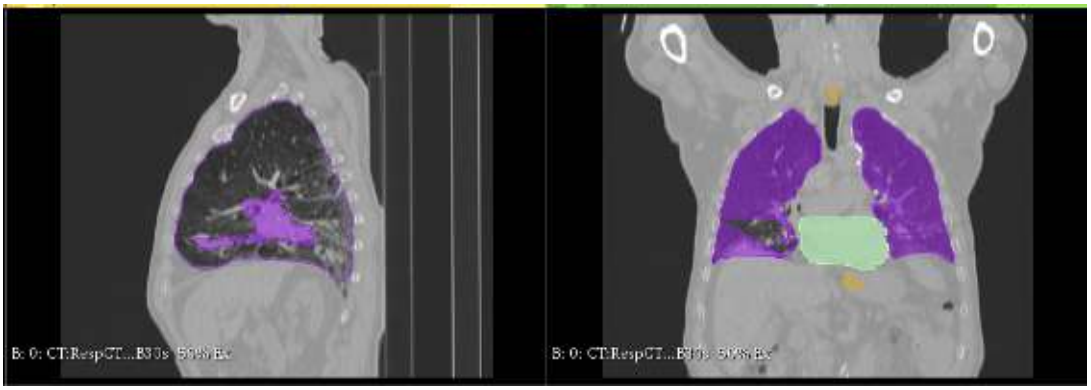


Figure 3.2: Common annotation errors, such as blank spaces(right), or inverted annotation(left).

3.1 AAPM Challenge

AAPM Thoracic Auto-segmentation Challenge is organized by American Association of physicists in medicine and it's goal is to provide auto-segmentation tool to segment organs at risk from CT images for radiation treatment planning [1].

Challenge focuses on segmenting 4 main organs in thorax:

- **oesophagus** - 23 to 28cm long tube-like organ connecting *pharynx* and *ventriculus*. It connects to *pharynx* at C6 vertebra and descends through thorax in front of *columna vertebralis* until at Th11 it finally ends in *ostium cardiacum* giving onto *ventriculus*. As far as CT granularity goes, it consists of muscle tissue.
- **cor** - muscular organ with 4 cavities, working as a pump. It is situated behind *os sternum* in *mediastinum*. 2/3 of its volume is left of medial line and 1/3 is on the right. *Cor* is shaped like irregular cone with apex pointing to the fifth intercostal space. Thinnest part are *atriaes* with only 2.5mm. This migh be in worst case scenario, below CTs resolution and must be accounted for.
- **pulmo** - pair organ that provides transfer of gases between air and blood. Each part *pulmo dexter et sinister* divides into lobes and then into even smaller segments. However for segmentation purposes, that high granularity is unnecessary. It is surrounded by ribs and fills almost whole thoracic cavity. While breating, volume and density of lungs changes drasticaly and needs to be accounted for.
- **medulla spinalis** - goes through *canalis spinalis* from C1 to L2. Like brain, it consists from gray and white matter. It is approximately 40 to 50 cm long.

[1] `http://aapmchallenges.cloudapp.net/competitions/3#learn_the_details-overview`

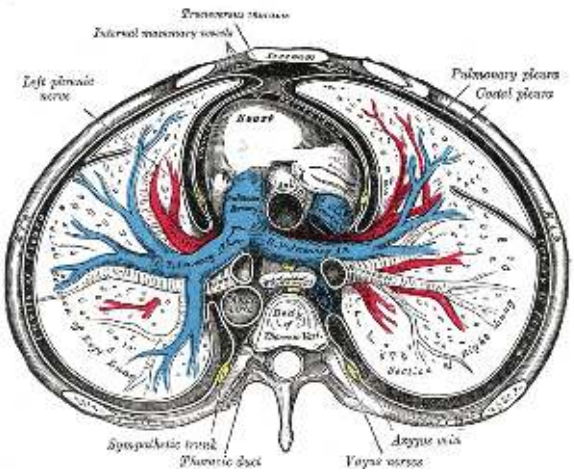


Figure 3.3: Atlas drawing of transversal plane of thoracic cavity depicting spatial relations of organs of the region. (from `https://www.wikiskripta.eu`)

Exact instructions for segmentation are enclosed in attachments.

Chapter 4

Proposed Method: 3D UNet adaptations

It has been years since Neural Networks employing deep learning have surpased traditional computer vision algorithms for complex tasks, such as semantic segmentation or multiclass classification. Since 2012 when AlexNet emerged, state-of-the-art became rapidly advanced by deep learning [26].

Lately, in multiple medical segmentation challenges, **3D Unet** has been successfuly used. Tasks like segmentation of pancreas [27], prostate [20], xenopus kidneys [25], neural cell structures in electron microscope stacks [24], and many more. Most of them have also published their implementations on github.

As mentioned in 2.6, 3D Unet requires very few training images, which perfectly matches described dataset used for this paper and most of real-world medical applications. All this reasons lead us to conclusion to use **3D Unet as our primary method**, and to further boost it, with preprocessing using some of the traditional methods of computer vision.

As a starting point for organ segmentation we use remplementation by Imran Ahmed available on github[1] of Unet proposed by Çiçek [25]. Originally it was trained dataset from Automated Segmentation of Prostate Structures ISBI Challenge. Since the challenge was ended long before this reimplementation was available, it is not possible to compare it to competition winner on hidden evaluation dataset.

4.1 Unet Architecture

Unet is implemented exactly as described in [25] (see 4.1), having 4 layers 0-3, each consisting of constricting and expanding opposite. Only difference being, that operations such as max pooling or convolution are performed in 3D, preserving more contextual information.

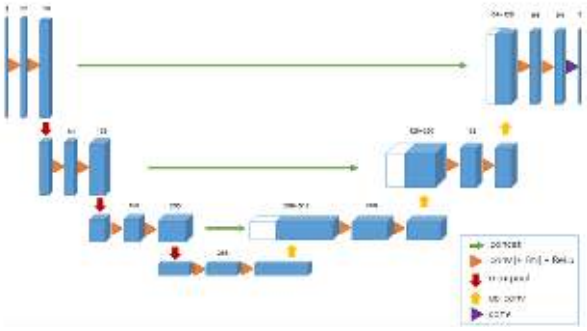


Figure 4.1: The 3D-Unet architecture overview [25].

[1] `https://github.com/96imranahmed/3D-Unet`

Each compressing layer performs 2 times convolution, normalization and ReLu, then max-pooling before entering the next layer. Also dropout of input for following layer is performed, to reduce the chance of overfitting.

Expanding layers first perform single upconvolution or transposed convolution. Then concatenation with features from opposing layer is performed. Similar to compressing layers, 2 times convolution, normalization and ReLu is performed.

On the last expanding layer, one extra 3D convolution is performed, with number-of-classes filters. More implementation details about implementation are available in documentation Technical documentation 3D-Unet reimplementation.

Features from last expanding layer could either be translated to organ mask using one-hot encoding, since there is number-of-classes output features, or concatenated with topological information and passed to fully connected layer.

4.2 Annotations extraction

To be able to measure the success of these methods, it was necessary to extract the annotations from provided dataset, which proved to be a suprisingly difficult and time consuming task.

No easy nor automated way was found to extract the annotations that was working, because labels have been hardwired into to data, even if they were in separate file, in binary form. Only tool, except for Slicer, able to read them was some obscure, poorly documented library in Matlab, and even if it was able to extract the data in some form, it couldn't be serialized properly, without loosing orientation information.

To extract it, each single patient had to be loaded to DICOM visualisation tool Slicer, and for each organ, had to created a binary labelmap from the polygon representation. This binary labelmap then had to be exported separately for each organ, for each patient.

Another problem with annotations was inconsistent class numbering. This means that for certain scan organs were labeled like: 1 - heart, 2 - left lung etc. and on another scan heart was 3, left lung 1 and so on (see image 4.3). This problem was discovered when trying to train Unet and loss function was jumping up and down. (see image 4.2)

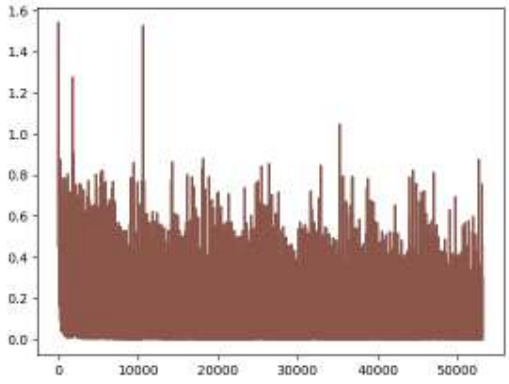


Figure 4.2: Loss function jumping up and down across training episodes.

Fixing this issue required exporting middle transversal slice (high probability of capturing all thoracic organs in this slice) of each scan to spreadsheets and then creating a ruleset for cells to color organs in slice with consistent colors and then creating translation table from actual label number to expected value.

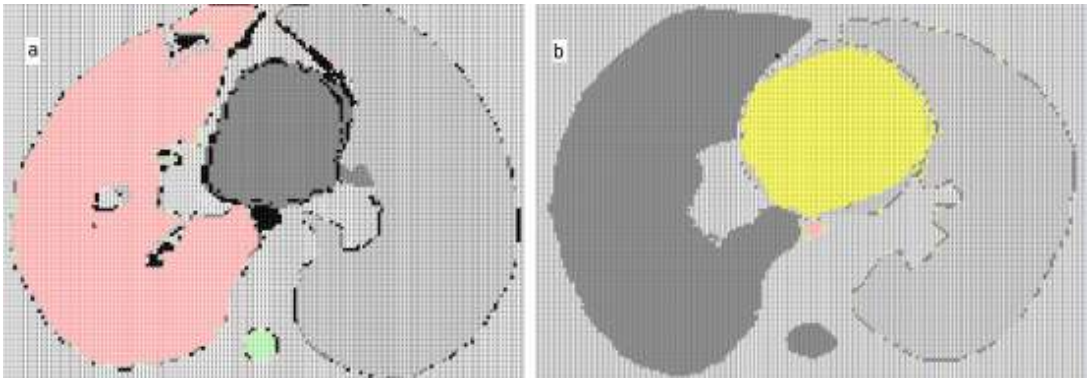


Figure 4.3: Different coloring of the same slice on different scans using the same excel style-by-value ruleset.

Metadata in DICOM format also include positional data about patient. When extracting raw voxel values, they come in slices. Those slices however are not guaranteed to be sorted correctly (in cranio-caudal direction), because SlicePosition attribute only describes relative positions of slices. This means that each slice has correct neighbours, but sometimes first slice is from superior direction and sometimes from inferior, depending on the scanner machine.

To sort them correctly, we need to use attribute SliceLocation, which describes Z (Z is cranio-caudal axis) coordinate. Depending on the location of scan, scanner settings and patient position and size, starting value can range from -1000 to 1000 (observed in my data) for slice in same position. However it is guaranteed to decrease in caudal direction.

4.3 Data preprocessing used for training

Since data is consisting from multiple studies and is collected from multiple CT scanners, with different brightness settings and different sizes, it is necessary to normalize it.

4.3.1 Brightness normalisation

First, we cast data to Hounsfield units Using formula: $image = image \cdot slope + intercept$ This guarantees same level of brightness for same tissues regardless of scanner. This should, in theory, also allow us to train/test using multiple different datasets, if we would obtain any.

4.3.2 Rescaling

Second, we rescale scans so that they fit into VRAM of available computer. Since we are using Nvidia Titan V with 12GB of VRAM, maximum scan size can be composed from approx. 3 millions of voxels (single valued) for Unet with described architecture.

Since scans are not from the same machine, their resolution is different, therefore it is necessary to rescale them even if we would put patches as subtractions of original data. Originaly data is size around 130x512x512 voxels +-10% in each axis, which requires approx. 100GB of VRAM.

For scan, rescaling is done using `scipy.ndimage.interpolation.zoom` with order of spline interpolation of 3. For labels same function with spline interpolation of 0 and without prefiltering. Using filter and higher order interpolation, leads to seeding foreign labels on the edges of organs, due to intruction of noise, which is not necessarily bad on input data, however very bad for labels.

We have tried rescaling to multiple sizes, from using cube of 500^3, cube of 360^3 serving data in smaller patches of 120x120x120, with tradeoff of loosing context but having higher resolution, and eventually using 60x240x240 which is roughly half the size in each dimension, making memory requirements satisfiable, cutting them from approx. 100GB to 12GB of VRAM for whole scan.

4.3.3 Distance transform

For loss function (see: 4.6.3) based on distance we need to calculate 3D distance transform maps for each individual organ for each scan. Distance transform map says for each voxel in binary organ mask in labels, how distant it is from closest voxel belonging to organ, so it is possible to compute punishment for Unet based on distance from actual organ. There are several ways of achieving this.

Naive approach is using k-times morphological operation of dilation on binary mask for each organ. This approach however is extremely slow for larger structuring elements to compute and requires huge amount of iterations for smaller structuring elements, and distance transform map produced is quantized.

More advaced method is using Multi Label Anisotropic Euclidean Distance Transform 3D(MLAEDT-3D)[2] that can compute continuous euclidean distance transform in 6 passes, making it linear time algorithm.

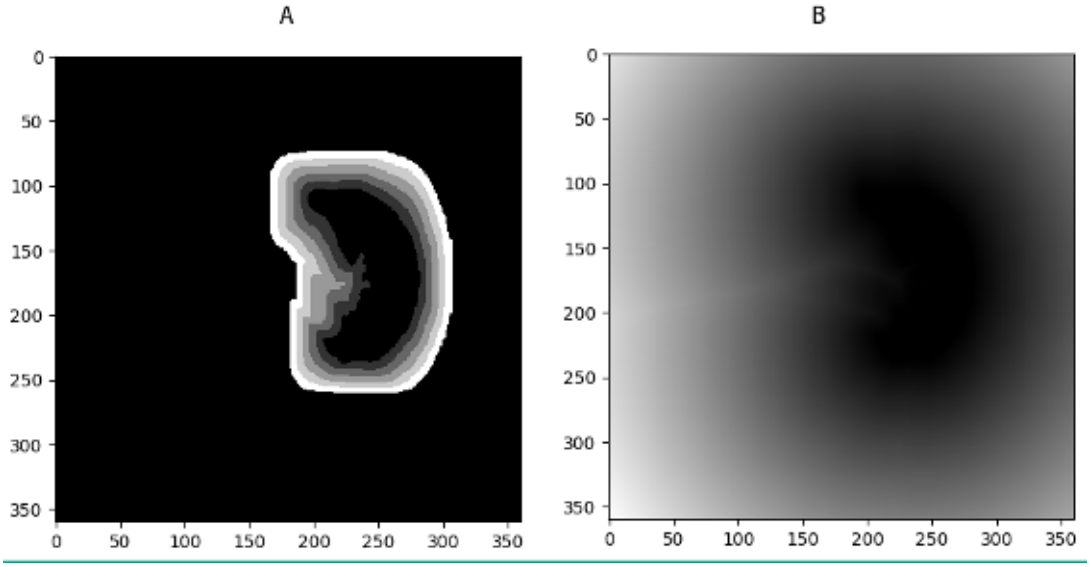


Figure 4.4: Distance transform visualisation for left lung. Brighter white indicates further distance from organ boundary. A: Distance transform produced by dilation in 10 iterations. B: Continuous euclidean distance transform produced by MLAEDT-3D.

4.3.4 Coversion to binary label masks

Original labels are provided as multilabel mask for each scan, where each organ is assigned a number. Since our solution is mixture of experts, each trained for single organ, conversion to binary label masks, one for each individual organ is necessary.

Simplest way is to create empty label map with same size as multilabel map, and set as truthy positions where extracted organ assigned number is located.

4.4 Dataset partitioning

Only 36 body scans from 3 studies are available and from those 7 are reserved as test data and another 7 as validation data, only 22 scans remain for training.

It is possible to boost training data sample amount at cost of validation data, using e.g. 10-fold cross validation, however since training Unet is very time consuming - taking up to 2 days to fully train, and with both GPU and manpower time limited, we deemed the tradeoff unworthy.

[2] `https://github.com/seung-lab/euclidean-distance-transform-3d`

Both validation and test data are handpicked, to represent each study and every machine used in the study. Also outlayers, such as man with collapsed lung is included not in train, but in validation dataset, so we are able to judge the generalisation level of network.

4.5 Data augmentations

To mitigate the problems of small dataset, such as likelihood of overfitting, or "memorizing" desired output for each input by network, it is necessary to augment original input data as a form of regularization.

We are using 3 types of augmentations, each with own individual probability of triggering on currently augmented image. Triggering each type of augmentation with individual probability means chance of producing more variants for same image, when some of the augmentations may not apply.

All abovementioned augmentations, except for grayscaling, need to be applied for annotations as well, with same parameters by default

To further increase data variability, we generated augmented data just-in-time, rather than staticaly before training and reusing them. However this is very time consuming, since augmentation is performed at every iteration.

4.5.1 Rotation

Probability of triggering rotation is 0.6. Rotation is done along first two axes, meaning rotations in transversal and saggital plane, since that is only reasonable plane of rotation of real body in scanner (represents laying on the side, rather than on back). Rotations in facial plane are unexpected, since scanning is done on straight leveled pad.

Angle of rotation, if rotation is triggered, is generated as random integer between 10 and -10 degrees and then applied using `scipy.ndimage.interpolation.rotation`.

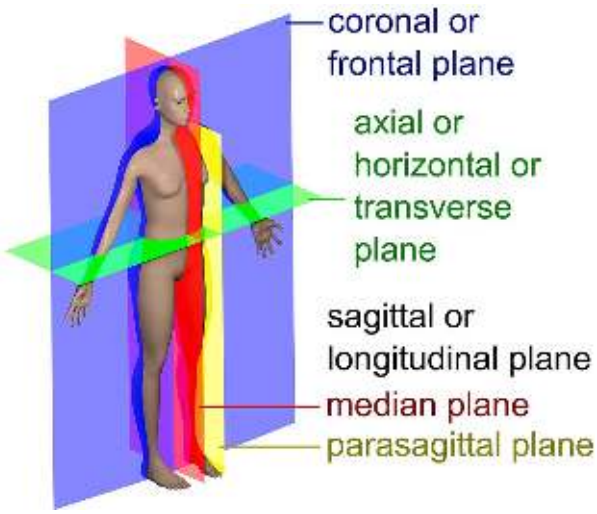


Figure 4.5: Termini situm et directionem partium corporis indicantes: Names of location and directions of body parts. Augmentation rotations are done saggital and transversal plane. *(By Mikael Häggström, used with permission.)*

4.5.2 Zoom

Probability of triggering zoom is 0.5. Same zoom function is applied for augmentation purposes as is for data preprocessing (see 4.3.2), but instead of new dimensions, scaling factor is used.

Scaling factor is randomly generated using uniform distribution from interval 1.0 to 1.2.

Since zooming means changing shape of input 3D voxel array, resulting image needs to be cropped to preserve required shape for Unet input. For each dimension width of bezels to crop is computed as $\frac{n-r}{2}$ where n is new shape after zooming and r is shape required by Unet.

4.5.3 Grayscaling

Grayscaling means slight modifications of voxel brightness values. It is triggered with probability 0.6.

Firstly, random number $k \in < 10, -10 >$ is selected and then 3D matrix M with same shape as scan is initialized, where $M : M_{ij} \in < 0, 1 >$. Grayscale modifier D for scan is then obtained as \$D=k $\cdot$ M\$which can be simply added to original scan, producing augmented result.

4.6 Training

Rescaled and normalized *input images* with their augmented counterparts and corresponding *label maps for single organ* and *distance transform maps* are used to train with *AdamOptimizer* implementation provided by Tensorflow[3]. Loss function is *dice score* computed over the last feature map of Unet and result is *enriched by false positive* matches scores, reached by summing over distance transform map.

Unet training, depending on organ, is very time consuming. 10 000 iterations (single forward-backward pass of one scan and it's online generated augmented versions), takes circa 30 hours in our implementation.

It is possible to train Unet as multi-organ segmentation tool, making output of last convolution produce $n+1$ features for each voxel, where n is number of organs trained for and 1 is for the background. However this approach leaves the space of possibilities very large, and further increases VRAM memory consumption.

One of the first models, we trained in 5+1 output features configuration, was only able to train for lungs, however it couldn't distinguish between left and right lung, and it also considered air surounding body as lungs.

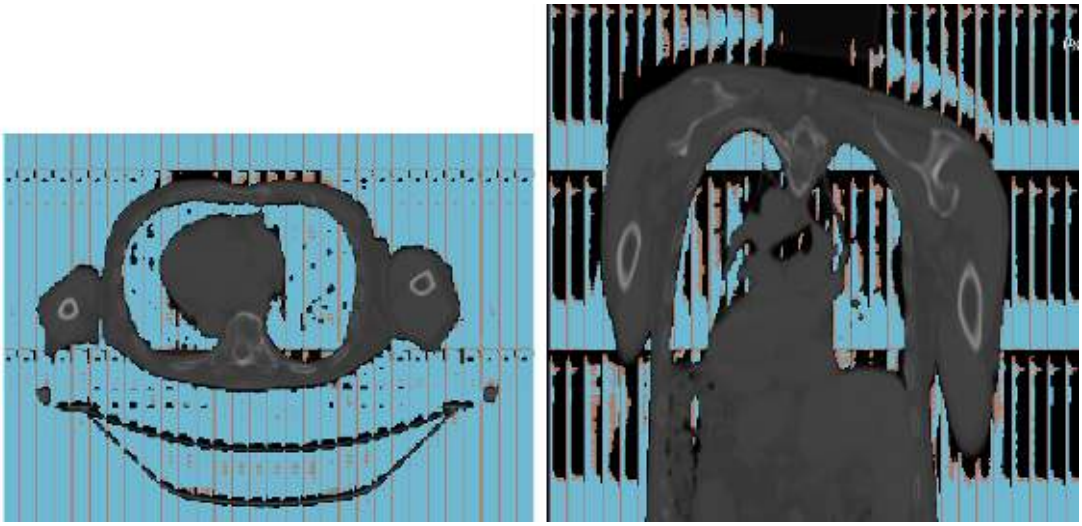


Figure 4.6: 5+1 multilabel organ segmentation. Unet only can segment lungs, without distinguish left and right, and considers air surrounding body as lungs as well, because of missing or having very little spatial context.

Training for only single organ, means saving some memory, which in return means we are able to serve network with larger patches, preserving more spatial context, or using patches with higher resolution, preserving more details. As far superior solution appears to be low-res image with whole context, highly

[3] `https://www.tensorflow.org/api_docs/python/tf/train/AdamOptimizer`

reducing false positive results in inappropriate anatomical areas and in synergy with our loss function, false positives are reduced to borders of compact organs.

4.6.1 Model evaluation metrics

As a metric on both validation and test dataset, we are using Dice-Sorensen score for IOU calculated for both background and segmented organ individually.

Intersection over Union metric: $IOU = \frac{S \cap GT}{S \cup GT}$, where S stands for network segmentation and GT for ground truth labels. It is relatively fast to calculate for binary label masks. There are 2 ways that are used to calculate it (sum stands for sum of all matrix elements, $\odot$ means elementwise matrix multiplication):

1. $IOU = \frac{sum(S \odot GT)}{sum(S)+sum(GT)-sum(S \odot GT)}$ - Jaccard index
2. $IOU = \frac{2 \cdot sum(S \odot GT)}{sum(S)+sum(GT)}$ - Dice-Sorensen score

Both metrics are very similar and both produce results between 0 - worst possible match, having no intersection but non-zero union, and 1 - best possible match, segmentation and ground truth are identical.

Since, there is risk, that when there is "nothing" on segmented image, and it is correctly segmented with mask of zeros, we are facing division by zero problem. We can either set condition that *if* $sum(GT) = 0$ *return* 1 or we can add small ϵ eg. $\epsilon = 10^{-5}$ to both numerator and denominator.

This metric is very punishing, since miss-segmenting single voxel is punished twice. Once it is missing in intersection and second time it is raising union. Therefore by naked eye evaluated segmentation as "near perfect" can have score around 0.9 and could falsly lead to reading the results are percentages, where missing 10% seems like much.

Figure 4.7: Intersection over Union score visualisation. This score is very punishing. (image from: `https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/`)

When reading and interpreting the results, one should also take into consideration, that ground truth data, also contains many errors, introduced by human segmentator e.g. image 3.2

4.6.2 Hyperparameters configuration

There are 2 main types of hyperparameters for this model:

1. Concerning *data* fed to network:
   - *Size and Depth* of scan (2 dimensions are enough to define scan, since base is square) - last model values: 240x60
   - *Size and Depth* of patch (2 dimensions for same reason as above)
   - Number of *classes* in desired output (2 for binary segmentation, more for multiorgan segmentation) - last model values: 240x60, same as scan
   - Number of *input features* in provided input. Simple CT-scans are providing only monochromatic brightness values, therefore 1D feature vector is provided per voxel, but since Unet used for fine-tuning uses the same architecture, and takes also as input results of previous network segmentation, this takes 2D vector of features per voxel.

- *Batch size* - determines how many patches are fed to network concurrently. This value heavily depends on available VRAM. For smaller patch sizes, we could feed up to 6 patches per batch, until the threshold of approx. $3 \cdot 10^6$ of voxels was crossed. For final patch shape (240x240x60) only single patch per batch is viable.
- *Augmentation rate* - determines how many augmentation iterations should be performed per training iteration on single data. The higher augmentation rate, more uniquely distorted data is sent to Unet, however, since we are running augmentations online, it is very time consuming and prolongues training significantly. Currently set to 1 - for each iteration 1 new instance of input data is produced.

2. Concerning *network configuration*:
   - *learning rate* - 0.001, which is default value for AdamOptimizer used in tensorflow[3]
   - Exponential *decay rate* for 1^{st} moment estimates $_1$ - experimentally set to 0.8 (see subsection: 4.6.4)
   - Exponential *decay rate* for 2^{nd} moment estimates $_2$ - experimentally set to 0.9 (see subsection: 4.6.4)
   - *Convolution kernel size* - shape of 3D kernel used for 3D convolutions. We used cube even for non-cube shaped data patches, with side **3 voxels**, making kernel size 27 voxels.
   - *Convolution stride* - defines movement of convolution kernel over features patch in each direction while performing convolution. It is defined as **1 in each direction**.
   - Number of filters - defines how many *image features* are on the output of convolution. First convolution of the first layer of the compression path starts with 8 filters and second convolution with 16 filters. For each subsequent layer in compression path, the value doubles from previous layer. For expansion path, first and second convolutions both start with 64 filters and for each layer, the amount halves (for more detailed values see table: 4.1).
   - *Maxpooling*[4] *kernel size* - defines how many values in neighbourhood of maxpooled value will be used.
   - *Maxpooling*[4] *stride* - defines movement of maxpooling kernel size, and therefore defines resolution reduction in each dimension. For first maxpooling on level zero, for both compression and expansion path are strides different from default 2x2x2, because of first dimension of data patch being 4 times smaller than other 2 dimensions, and downscaling all dimensions equally then causes distorsions due to need of using padding in convolutions. Kernel size is always identical to stride to avoid propagation of local brightness peak to the rest of the image.
   - *Droupout rate* - dropout is form of regularization preventing overfitting and it sets fraction of input, defined by dropout rate, to zero value. At the end of each layer dropout is performed with rate 0.2. Dropout is only applied in training mode.

Patch size implications

Having patches with higher resolution, but small or next to none context, often yielded false positive matches in completely different anatomical areas, than organ is supposed to be in (see image 4.8).

[4]Maxpooling applies for compression path and Upconvolution for expansive path.

Pipeline order	Path	Layer	Layer order	Number of filters	Maxpooling/ Upconvolution kernel size	Maxpooling/ Upconvolution stride
1	compression	0	1	8	None	None
2	compression	0	2	16	None	None
3	compression	1	1	16	1x2x2	1x2x2
4	compression	1	2	32	None	None
5	compression	2	1	32	2x2x2	2x2x2
6	compression	2	2	64	None	None
7	compression	3	1	64	2x2x2	2x2x2
8	compression	3	2	128	None	None
9	expansion	2	1	64	2x2x2	2x2x2
10	expansion	2	2	64	None	None
11	expansion	1	1	32	2x2x2	2x2x2
12	expansion	1	2	32	None	None
13	expansion	0	1	16	1x2x2	1x2x2
14	expansion	0	2	16	None	None
15	expansion	0	3	2	None	None

Table 4.1: Table of numbers of filters for each convolution performed in both compression and expansion layers. Last convolution outputs number of output classes +1 features. Maxpooling kernel size and stride is also provided.

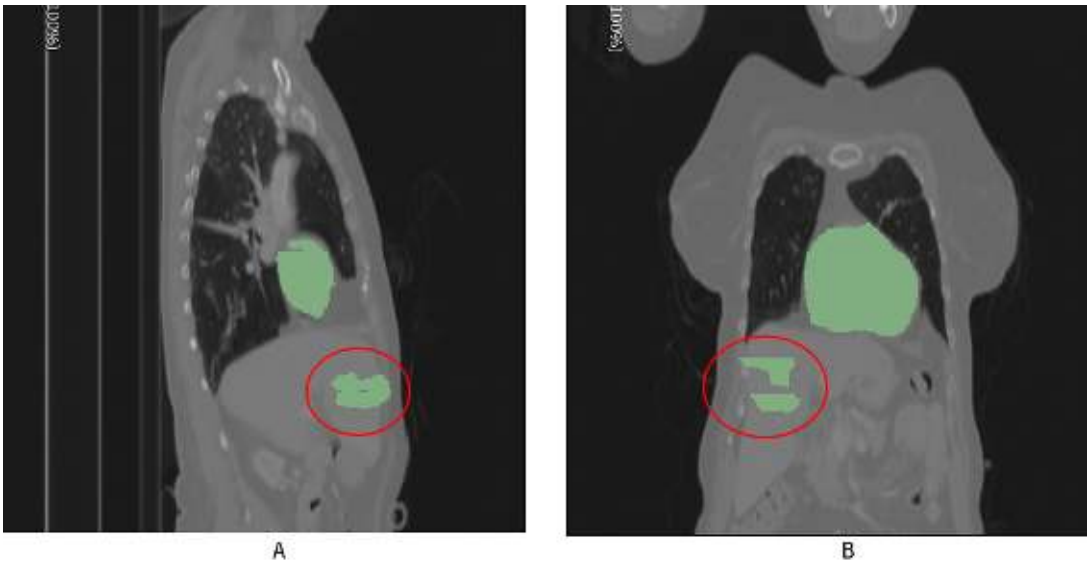


Figure 4.8: Higher details with lower context patches fed to Unet leads to false positive heart results in abdominal cavity (in red circles). A is medial slice and B is facial slice of the same scan. Patch size is 120x120x24

Another problem with this approach, that can be seen on the image, is by naked eye distinguishable boundaries between patches in segmentation, where there are sharp edges, or holes in resulting segmentation.

On the other hand, cutting resolution approx. to 1/8 of original voxel volume of scan, to shape 240x240x60, meant we could preserve much anatomical relationships, at cost of lower details. This leads to much more compact results without holes and patches boundaries, since there are none, and network is not producing false positive matches in completely different anatomical areas.

4.6.3 Loss functions

Final loss function used is *dice score* computed over the softmax of the last feature map of Unet and result is *enriched by false positive* matches scores, reached by summing Hadamard product of predictions and distance transform map.

$$dice = \frac{2 \cdot sum(output \odot target)}{sum(target) + sum(output)}$$

$$distance = sum(predictions \odot dtm)$$

$$loss = dice + distance$$

Figure 4.9: Augmented Loss function formula: dtm - distance transform map

Reason for computing dice score on feature map, rather than on label map is for *result* to be *differentiable*. For target we therefore have to use one-hot encoded ground truth labels.

Distance based part of loss function just amplifies punishment already given by dice score, but rather then just punishing false positives with same value, without taking into consideration compactness of human organ and anatomical positions, this part of loss does just that.

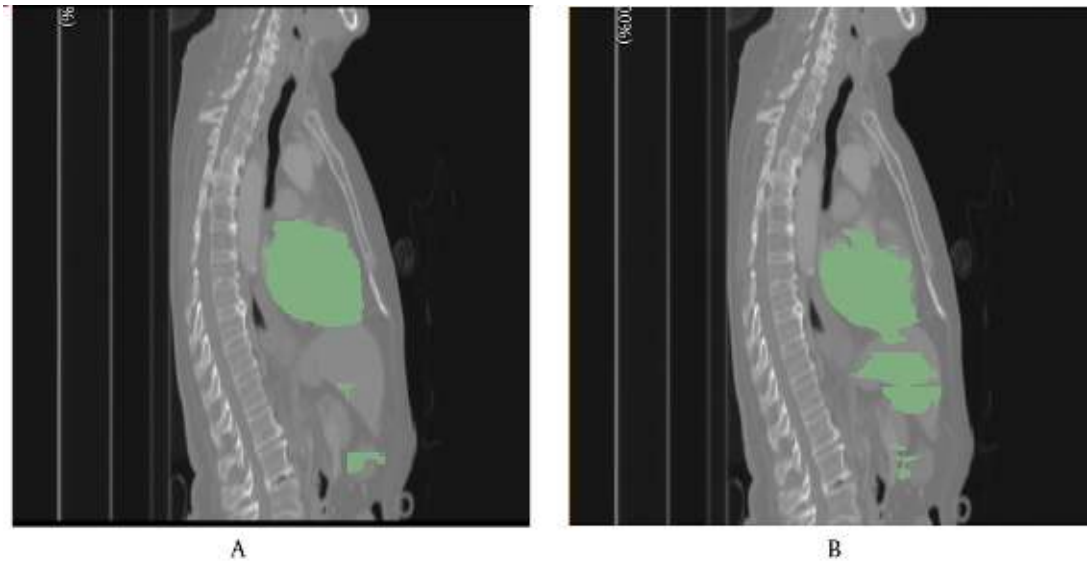


Figure 4.10: False positive results reduction visible on the same slice: A - augmented Dice loss, B - Dice loss.

Default loss function used by Ronneberger [24] was ballanced cross entropy, where it is possible provide class weights, when classes are imbalanced. Class weights are computed as $1 - \frac{n_{class}}{N}$

We used it for earlier experiments with multiclass segmentation, where computing dice coefficient for multiple classes is far more time consuming, since it has to be computed separately for each class. However since we went with idea of training multiple expert networks for binary segmentation, and since our main metric of segmentation performance is dice, it only seems reasonable to use dice as loss function as well. We are using implementation provided by Tensorlayer[5].

4.6.4 Trainer

As a trainer we use *Adaptive Moment Estimation* (Adam), which yet another extension of traditional Gradient Descent optimization algorithm. It combines the advantages of algorithms Adadelta or RM-

[5] `https://tensorlayer.readthedocs.io/en/stable/_modules/tensorlayer/cost.html`

Sprop of storing exponentially decaying average of past squared gradients and, similar to momentum, also keeps exponentially decaying average of past gradients [28].

Using default Adam configuration from Tensorflow[3] works fine for most organs, however for training lungs, loss function whilst training, began jumping unpredictably. Some of this erratic behaviour can be explained by inaccuracies in provided annotations (see image 3.2). But this behaviour continued even afterwards annotations were fixed (see image 4.2).

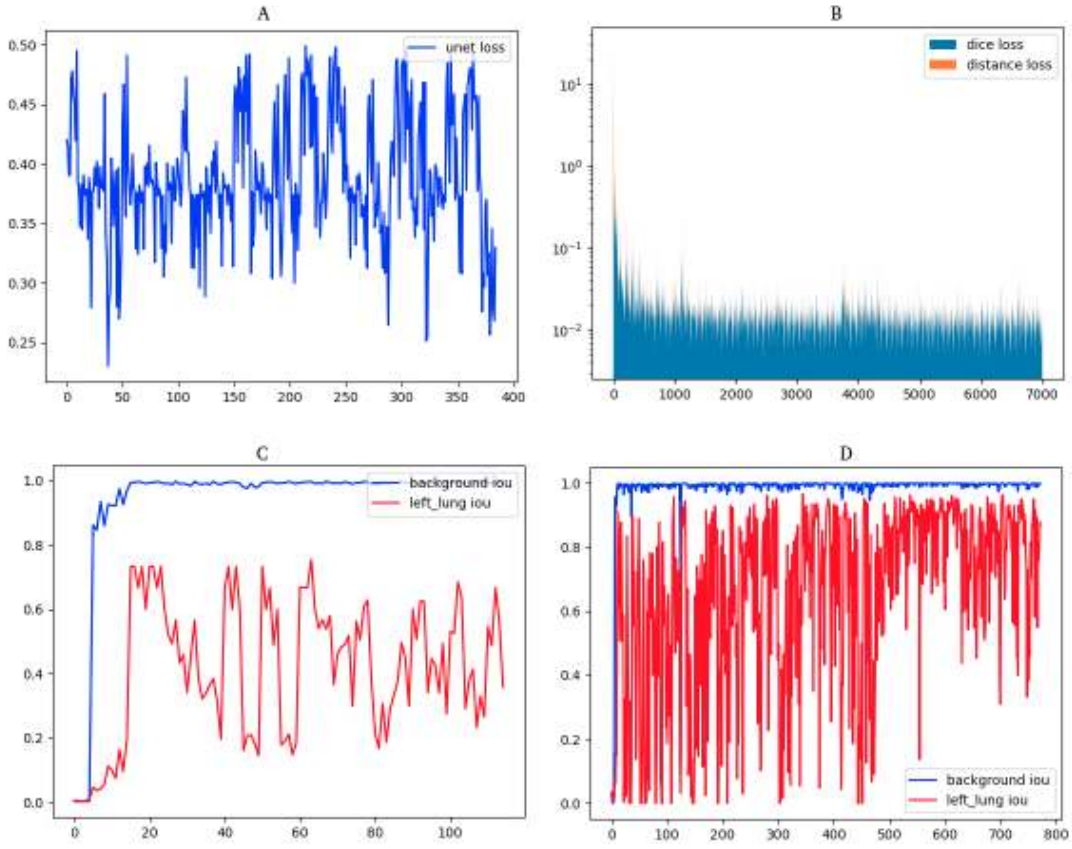


Figure 4.11: Jumpy loss function with default Adam parameters(A) and IOU metric(C) versus loss function with decreased β_1 and β_2 parameters(B) and it's IOU metrics(D).

We conducted multiple experiments, and conluded that best parameter settings for $\beta_1 = 0.8$ and for $\beta_2 = 0.9$ with learning rate $\alpha = 0.001$

4.6.5 Stop condition

We had a single criterion for stopping. Intersection over Union (IOU) metric on validation data was stable and stopped rising for several measurements in a row. If this criterion was not met, training was cut after reaching 10 000 iterations.

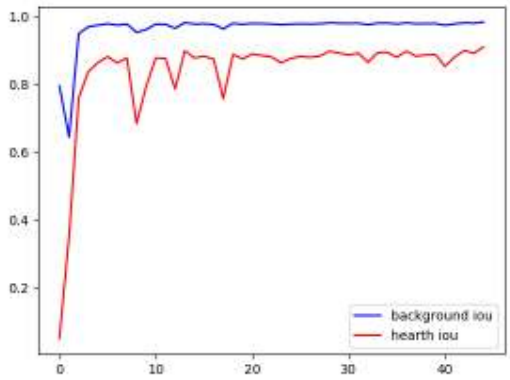


Figure 4.12: Intersection over Union metric is stable for multiple measurements on validation data, means time to stop training.

4.7 State of the art results on dataset

Comparison is done with currently best performing model on AAPM Thoracic Auto-segmentation Challenge, which currently is submission by user xuefeng. These results are used as reference model for all statistical evaluations.

Submission unfortunately does not include any details of implementation or methods used for segmentation, only Dice scores are present.

scan	heart Dice	left lung Dice	right lung Dice	esophagus Dice	spinal cord Dice
LCTSC-Test-S2-202	0.6026	0.7173	0.7741	0.3736	0.6132
LCTSC-Test-S2-201	0.3459	0.7418	0.7565	0.0	0.5846
LCTSC-Test-S1-201	0.6238	0.8007	0.7375	0.2417	0.6370
LCTSC-Test-S2-203	0.4517	0.6814	0.4551	0.2091	0.3240
LCTSC-Test-S1-203	0.6094	0.8426	0.8979	0.5624	0.7939
LCTSC-Test-S2-204	0.3753	0.6660	0.5597	0.5664	0.7391
LCTSC-Test-S3-203	0.4722	0.7602	0.6113	0.4408	0.5282
LCTSC-Test-S1-202	0.5100	0.8262	0.7362	0.3663	0.7728
LCTSC-Test-S3-201	0.3914	0.6198	0.5915	0.3409	0.6153
LCTSC-Test-S3-204	0.3155	0.7056	0.7401	0.0	0.7168
LCTSC-Test-S1-204	0.4151	0.8940	0.8861	0.2815	0.7136
LCTSC-Test-S3-202	0.4040	0.8989	0.8927	0.0	0.3001

Table 4.2: Table of Dice score for Challenge hidden test data of best submitted model at 2019/04/26 [6]

4.8 Proposed method single-stage 3D Unet evaluation

As mentioned before, we trained for each organ separately. Test dataset consists of 6 scans. For each organ we present Jaccard Dice score and confusion matrices for each test scan. We also provide statistical tests to determine if our results are statisticaly significantly better, same or worse for each organ.

[6] available at `http://aapmchallenges.cloudapp.net/my/competition/submission/271/detailed_results/`

All statistical testing of our model performance will be done against scores in table. Since we don't have ground truth labels for the same data, we will perform unpaired tests against our own test set, exluded from provided train data.

For heart and left lung proposed method shows statisticaly significantly better perfomance on Dice score metric, than reference method and not significantly different perfomance on rest of segmented organs.

4.8.1 Heart evaluation

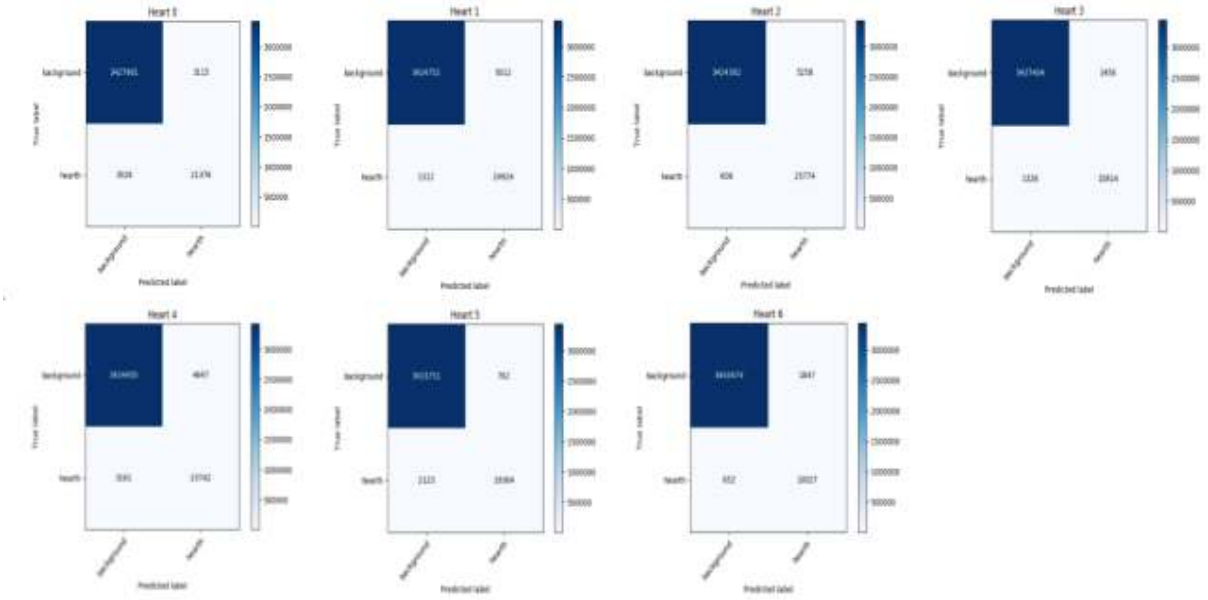


Figure 4.13: Confusion matrices for heart segmentation on test dataset.

Table 4.3: Table of results of Heart segmentation of proposed method.

scan	S1-010	S1-011	S1-012	S2-010	S2-011	S2-012	S3-010
heart dice	0.7629	0.7976	0.8146	0.8503	0.7525	0.8703	0.8004

```
Shapiro-Wilk normality test
proposed: W = 0.95225, p-value = 0.7502
reference: W = 0.91681, p-value = 0.2606
```

From the output of Shapiro-Wilk normality test, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution so parametric t-test preconditions are satisfied.

```
F test to compare two variances
F = 0.1637, num df = 6, denom df = 11, p-value = 0.03708
alternative hypothesis: true ratio of variances is not equal to 1
ratio of variances: 0.1636962
```

F test of variances equality shows us, that variances of our model and best model for heart, are statisticaly significantly not equal, and ratio of variances shows, that our model gives results with less variance, and therefore is more consistent.

Since value variances are statisticaly significantly better, we perform Welch Two Sample t-test to compare means of results.

```
Welch Two Sample t-test
t = 10.045, df = 15.764, p-value = 1.484e-08
alternative hypothesis: true difference in means is greater than 0
mean of proposed: 0.8069896
mean of reference: 0.4597971
```

P-value of Welch Two Sample t-test is lower than 0.05, therefore we won't dissmiss the alternative hypothesis, that proposed methods results are statisticaly significantly better with confidence level 0.95, than best results achieved so far on this data, for heart.

4.8.2 Left lung evaluation

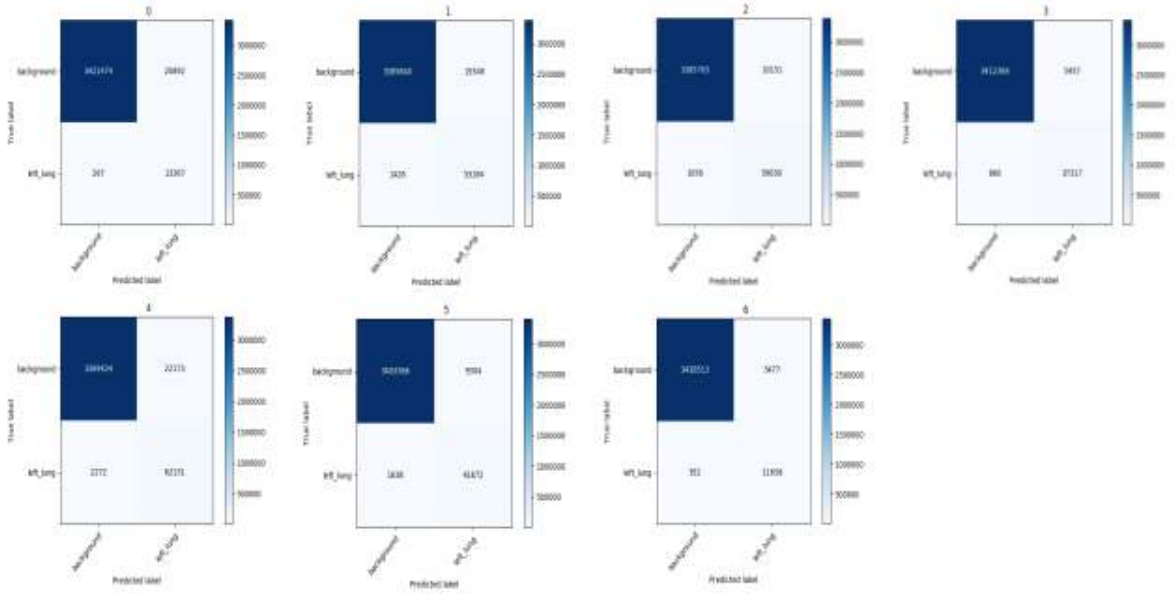


Figure 4.14: Confusion matrices for left lung segmentation on test dataset.

Table 4.4: Table of results of Left lung segmentation of proposed method.

scan	S1-010	S1-011	S1-012	S2-010	S2-011	S2-012	S3-010
left lung dice	0.3871	0.7580	0.8404	0.8552	0.7176	0.7920	0.6666

```
Shapiro-Wilk normality test
proposed: W = 0.82289, p-value = 0.06845
reference: W = 0.96152, p-value = 0.8052
```

From the output of Shapiro-Wilk normality test, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution so parametric t-test preconditions are satisfied.

```
F test to compare two variances
F = 3.1238, num df = 6, denom df = 11, p-value = 0.09732
alternative hypothesis: true ratio of variances is not equal to 1
ratio of variances: 3.12378
```

F test results conclude, that we can't dissmiss hypothesis that our model has the same variance of results, however p-value is not very high and also ratio of variances suggest, that our model may in fact be less stable.

```
F test to compare two variances
F = 3.1238, num df = 6, denom df = 11, p-value = 0.04866
alternative hypothesis: true ratio of variances is greater than 1
```

Another F test, this time with alternative hypothesis, that variance of our model is greater than of the currently best model in fact dismisses the equality of variances hypothesis.

```
Welch Two Sample t-test
t = -0.70241, df = 8.2924, p-value = 0.5017
alternative hypothesis: true difference in means is not equal to 0

t = -0.70241, df = 8.2924, p-value = 0.7492
alternative hypothesis: true difference in means is greater than 0

t = -0.70241, df = 8.2924, p-value = 0.2508
alternative hypothesis: true difference in means is less than 0
```

Welch Two Sample t-test results conclude, that with confidence level 0.95 we cannot dissmiss the hypothessis, that proposed method and reference method are performing with equal mean of IOU metric for left lung segmentation.

4.8.3 Right lung evaluation

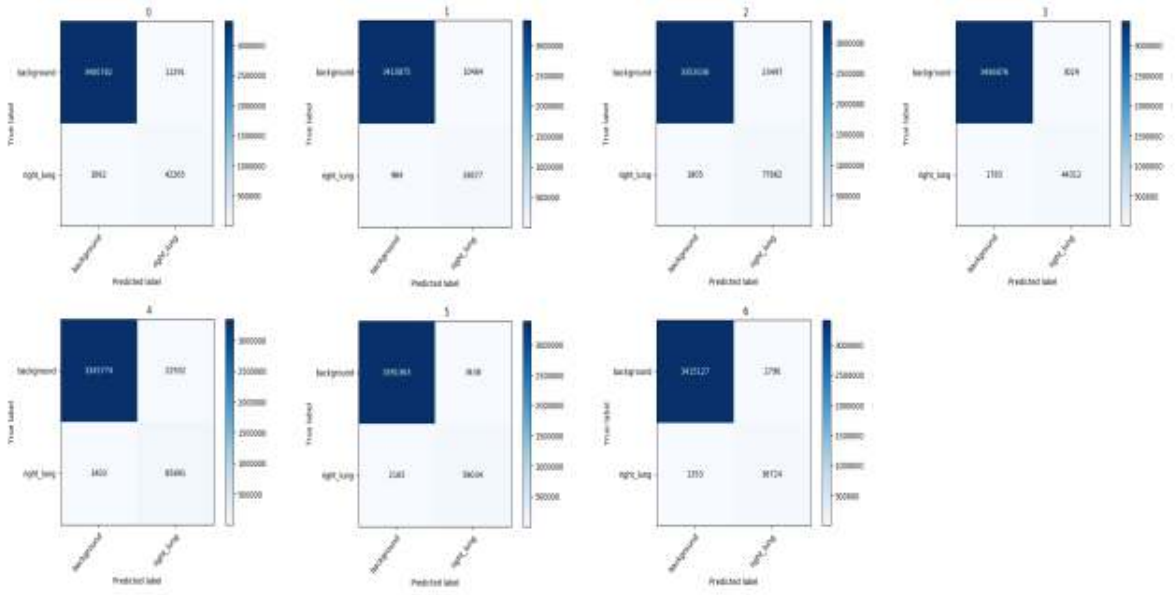


Figure 4.15: Confusion matrices for right lung segmentation on test dataset.

Table 4.5: Table of results of right lung segmentation of proposed method.

scan	S1-010	S1-011	S1-012	S2-010	S2-011	S2-012	S3-010
right lung dice	0.7654	0.7463	0.7542	0.9020	0.7792	0.9133	0.8984

```
Shapiro-Wilk normality test
W = 0.78178, p-value = 0.02693
```

Shapiro-Wilk normality test says that on confidence level 0.95 we cannot dissmiss the alternative that our results may not be from normal distribution, so we have to perform non-parametric t-test.

```
Wilcoxon rank sum test
W = 67, p-value = 0.01792
alternative hypothesis: true location shift is greater than 0
```

Wilcoxon rank sum test concludes, that with confidence level 0.95 we cannot dissmiss the hypothesis that proposed method performs statisticaly significantly better than reference method for right lung segmentation.

4.8.4 Spinal cord evaluation

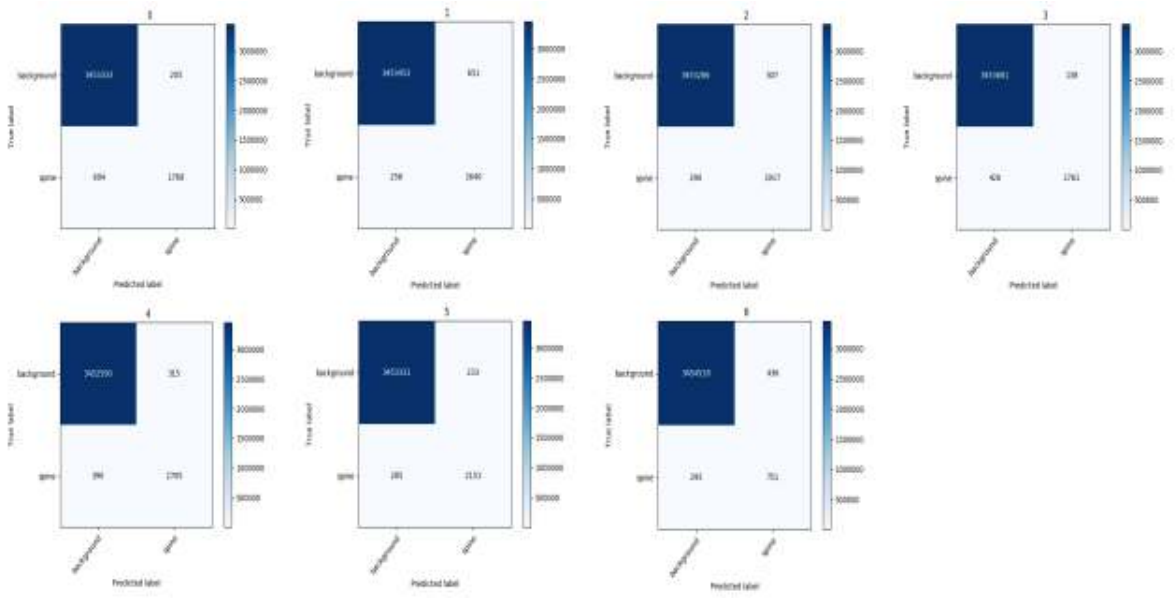


Figure 4.16: Confusion matrices for spinal cord segmentation on test dataset.

Table 4.6: Table of results of spinal cord segmentation of proposed method.

scan	S1-010	S1-011	S1-012	S2-010	S2-011	S2-012	S3-010
spinal cord IOU	0.6629	0.6438	0.7063	0.7593	0.7932	0.8059	0.5067

```
Shapiro-Wilk normality test
proposed: W = 0.9216, p-value = 0.482
reference: W = 0.87706, p-value = 0.08037
```

From the output of Shapiro-Wilk normality test, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution so parametric t-test preconditions are satisfied.

```
F test to compare two variances
F = 0.41998, num df = 6, denom df = 11, p-value = 0.2981
alternative hypothesis: true ratio of variances is not equal to 1
ratio of variances: 0.4199764
```

F test results imply that we on confindence interval 0.95 variances of results of proposed and reference method are significantly different. This means that both models are statisticaly significantly equally consistent with their prediction variances.

```
Two Sample t-test
t = 1.2497, df = 17, p-value = 0.2283
```

```
alternative hypothesis: true difference in means is not equal to 0

t = 1.2497, df = 17, p-value = 0.1142
alternative hypothesis: true difference in means is greater than 0

t = 1.2497, df = 17, p-value = 0.8858
alternative hypothesis: true difference in means is less than 0

mean of proposed: 0.6969216
mean of reference: 0.6116042
```

Two Sample t-test results conclude, that with confidence level 0.95 we cannot dissmiss the hypothesis, that proposed method and reference method are performing with significantly different mean of IOU metric for spinal cord segmentation.

4.8.5 Esophagus evaluation

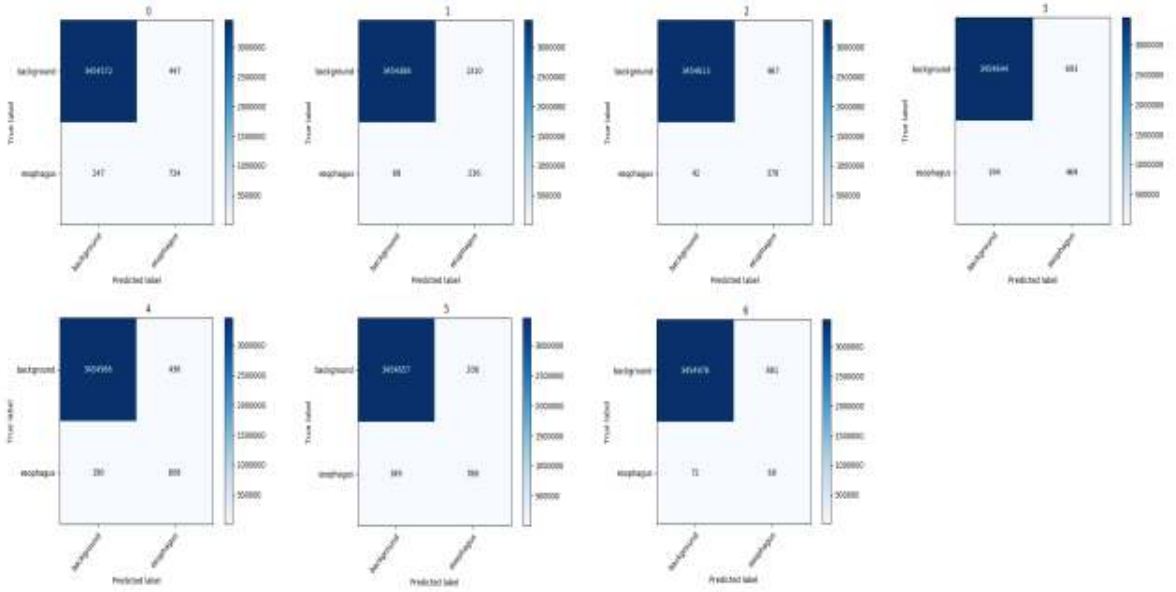


Figure 4.17: Confusion matrices for esophagus segmentation on test dataset.

Table 4.7: Table of results of esophagus segmentation of proposed method.

scan	S1-010	S1-011	S1-012	S2-010	S2-011	S2-012	S3-010
esophagus IOU	0.5140	0.1462	0.2725	0.3458	0.5634	0.5867	0.0675

```
Shapiro-Wilk normality test
proposed: W = 0.91686, p-value = 0.4454
reference: W = 0.91168, p-value = 0.2242
```

From the output of Shapiro-Wilk normality test, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution so parametric t-test preconditions are satisfied.

```
F test to compare two variances
F = 1.0427, num df = 6, denom df = 11, p-value = 0.8981
alternative hypothesis: true ratio of variances is not equal to 1
ratio of variances: 1.042678
```

F test results imply that we on confindence interval 0.95 variances of results of proposed and reference method are significantly different. This means that both models are statisticaly significantly equally consistent with their prediction variances.

```
Two Sample t-test
t = 0.77118, df = 17, p-value = 0.4512
alternative hypothesis: true difference in means is not equal to 0

t = 0.77118, df = 17, p-value = 0.2256
alternative hypothesis: true difference in means is greater than 0

t = 0.77118, df = 17, p-value = 0.7744
alternative hypothesis: true difference in means is less than 0

mean of proposed: 0.3566209
mean of reference: 0.2819377
```

Two Sample t-test results conclude, that with confidence level 0.95 we cannot dissmiss the hypothesis, that proposed method and reference method are performing with significantly different mean of IOU metric for esophagus segmentation.

4.9 Proposed method 2-stage 3D Unet evaluation

Training for only single organ, means saving some memory, which in return means we are able to serve network with larger patches, preserving more spatial context, or using patches with higher resolution, preserving more details.

To keep both advantages, namely spatial context - ideally all of it, and fine details, you either need huge amount of memory and extremely large network, or you can use 2 networks.

First network is the same, as described above in section 4.6. Output segmentation from this Unet is used to crop huge amount of irrelevant data from input data. We create 3D bounding box around the binary label map, and inflate it slightly (10% in each direction in each dimension) to compensate for inaccuracies of previous segmentations. This 3D bounding box is then applied to slice the input CT scan, and resulting slice is then rescaled to fit as input to second stage Unet (120x120x120).

Similar approach was proposed for multiclass segmentation by Wang [29], however their second stage network has different architecture from first stage, and we use the same Unet architecture with only size of input and output patches differing. Second Unet also takes as input binary label map from first stage unet, so number of input channels is doubled.

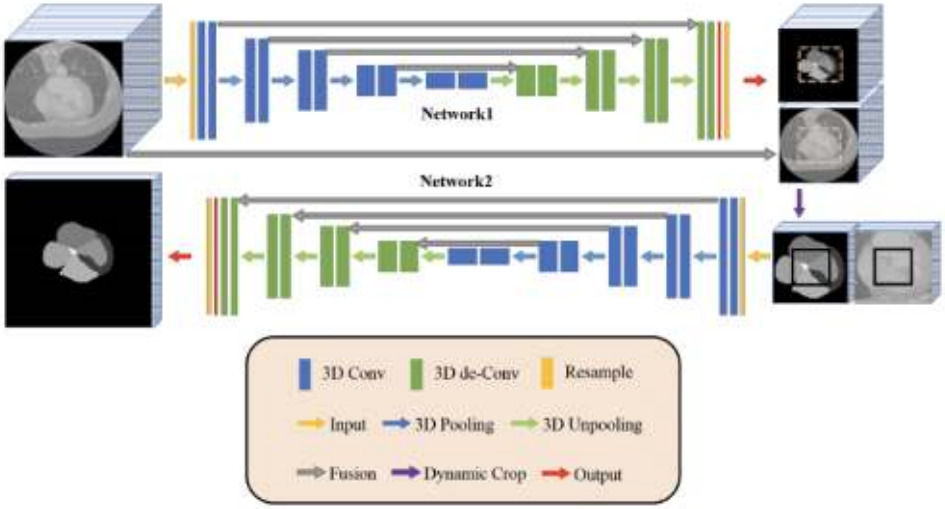


Figure 4.18: Schema of 2 Unets working in tandem. First Unet does rough estimate, second Unet takes segmentation from first Unet and fine tunes the results on cropped data with more details. (image adapted from [29])

Networks are trained independently. First unet is using the same model evaluated in section 4.8 and second is trained with outputs provided by the first one. It uses the same data augmentation framework and same evaluation framework. However loss function is now pure Dice score, without augmentations with distance transform maps.

Same as in section 4.8, model for each organ is trained separately. Test dataset is also the same 6 scans. This allows us to perform paired statistical tests against our own results, as well as unpaired testing against reference results of best performing model (in table 4.2).

2-stage 3D Unet was trained for heart, left lungs and right lungs. For all 3 organs we see statisticaly significantly better results compare to both single stage unet and reference model.

4.9.1 Heart evaluation

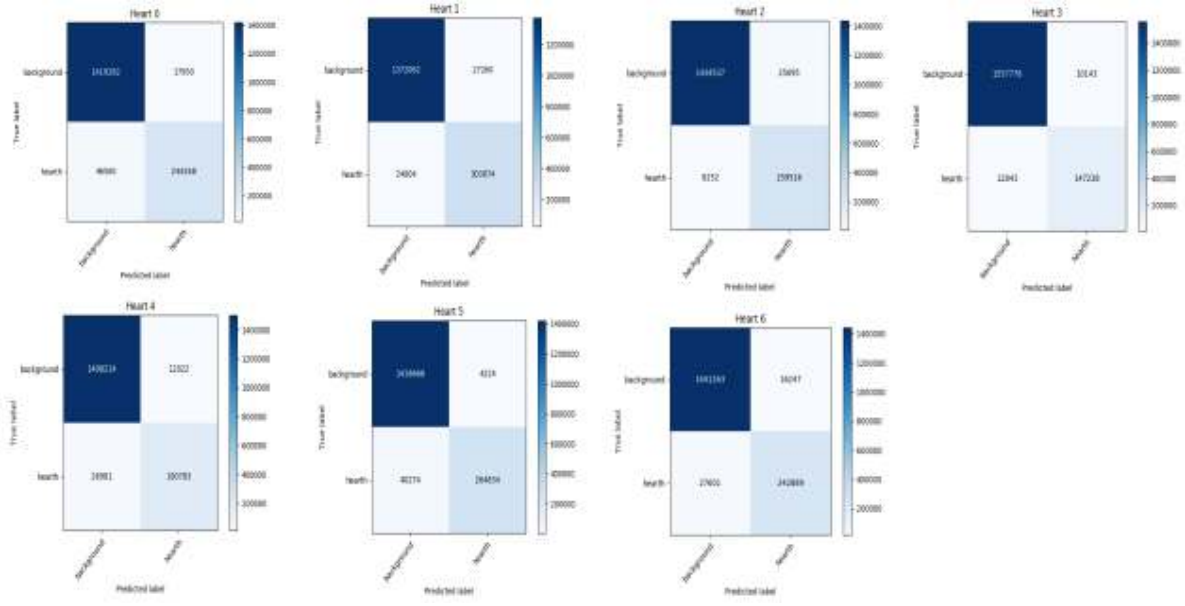


Figure 4.19: Confusion matrices for heart segmentation using dual Unet on test dataset.

Table 4.8: Table of results of Heart segmentation of proposed method.

scan	S1-010	S1-011	S1-012	S2-010	S2-011	S2-012	S3-010
single unet heart Dice	0.76291	0.79761	0.81465	0.85039	0.75251	0.87033	0.80049
2-stage unet heart Dice	0.79562	0.86224	0.88153	0.86368	0.83274	0.85877	0.84127

```
Shapiro-Wilk normality test
proposed old: W = 0.95225, p-value = 0.7502
proposed new: W = 0.9209, p-value = 0.4764
reference: W = 0.91681, p-value = 0.2606
```

From the output of Shapiro-Wilk normality test, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution so parametric t-test preconditions are satisfied.

```
F test to compare two variances
alternative hypothesis: true ratio of variances is not equal to 1
F = 0.1637, num df = 6, denom df = 11, p-value = 0.03708
ratio of variances: 0.1636962
```

F test of variances equality shows us, that variances of our model and best model for heart, are statisticaly significantly not different, and ratio of variances shows, that our model gives results with less variance, and therefore is more consistent.

It also shows, that variance is not statisticaly significantly different from previous single stage model.

Since value variances are statisticaly significantly better, we perform Welch Two Sample t-test to compare means of results with reference and paired t-test with single stage model.

```
Welch Two Sample t-test
t = 12.008, df = 13.448, p-value = 7.2e-09
alternative hypothesis: true difference in means is greater than 0
mean of proposed: 0.8479836
mean of reference: 0.4597971
```

P-value of Welch Two Sample t-test is lower than 0.05, therefore we won't dissmiss the alternative hypothesis, that proposed methods results are statisticaly significantly better with confidence level 0.95, than best results achieved so far on this data, for heart.

```
Paired t-test
t = 3.3299, df = 6, p-value = 0.007904
alternative hypothesis: true difference in means is greater than 0
mean of the differences: 0.040994
```

P-value of paired t-test is inferior to 0.05, so we can conclude, that results of new method are statisticaly significantly better then previous method, that was already better than reference model.

4.9.2 Left lungs evaluation

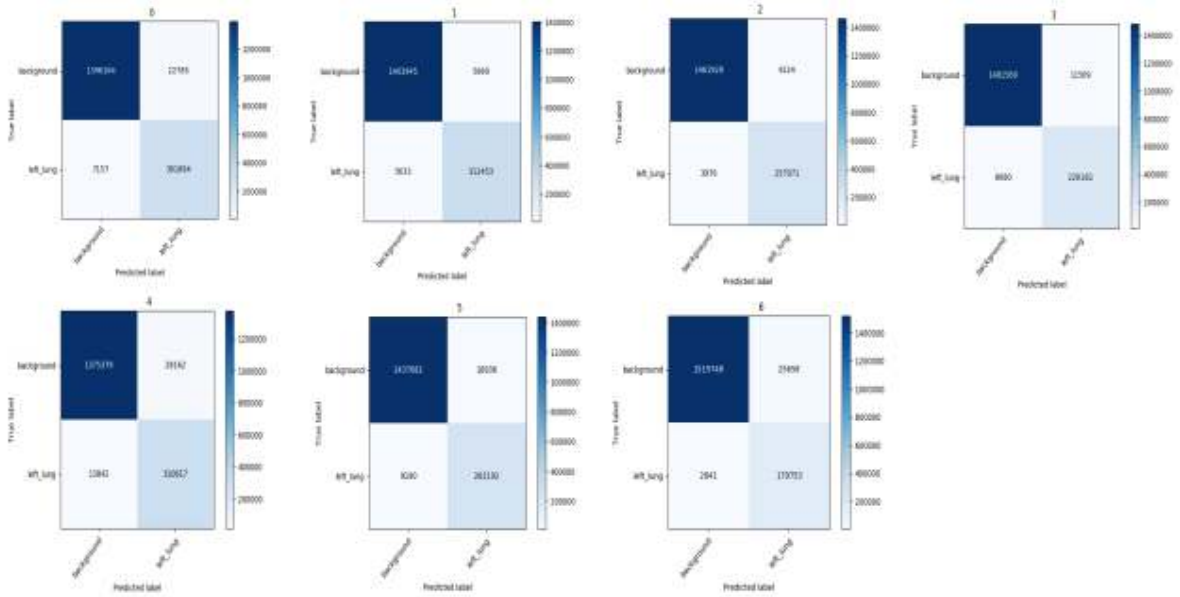


Figure 4.20: Confusion matrices for left lung using dual unet segmentation on test dataset.

Table 4.9: Table of results of Left lung segmentation of proposed method and dual unet.

scan	S1-010	S1-011	S1-012	S2-010	S2-011	S2-012	S3-010
single unet left lung Dice	0.38715	0.75803	0.84044	0.85522	0.71764	0.79203	0.66666
2-stage unet left lung Dice	0.90269	0.94704	0.96443	0.91581	0.87727	0.90094	0.863

```
Shapiro-Wilk normality test
proposed old: W = 0.82289, p-value = 0.06845
proposed new: W = 0.95862, p-value = 0.8068
reference: W = 0.96152, p-value = 0.8052
```

From the output of Shapiro-Wilk normality test, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution so parametric t-test preconditions are satisfied.

```
F test to compare two variances
F = 0.15783, num df = 6, denom df = 11, p-value = 0.03385
alternative hypothesis: true ratio of variances is not equal to 1
ratio of variances: 0.1578306
```

F test results conclude, that variances are statisticaly significantly different, and therefore Welch two sample test is required to compare new model with reference model.

```
Welch Two Sample t-test
t = 5.0116, df = 15.656, p-value = 6.817e-05
alternative hypothesis: true difference in means is greater than 0
mean of 2-stage unet: 0.9102580
mean of reference: 0.7629261

Paired t-test
alternative hypothesis: true difference in means is greater than 0
sample estimates:
mean of the differences between 2-stage and 1-stage unet: 0.1935137
```

Welch Two Sample t-test results conclude, that with confidence level 0.95 we cannot dissmiss the hypothessis, that proposed method is statisticaly significantly better than reference method, as opposed to single stage unet, which was not significantly better performing on Dice score metric. 2-stage unet also performs staticaly significantly better than 1-stage unet as shown by paired t-test.

4.9.3 Right lungs evaluation

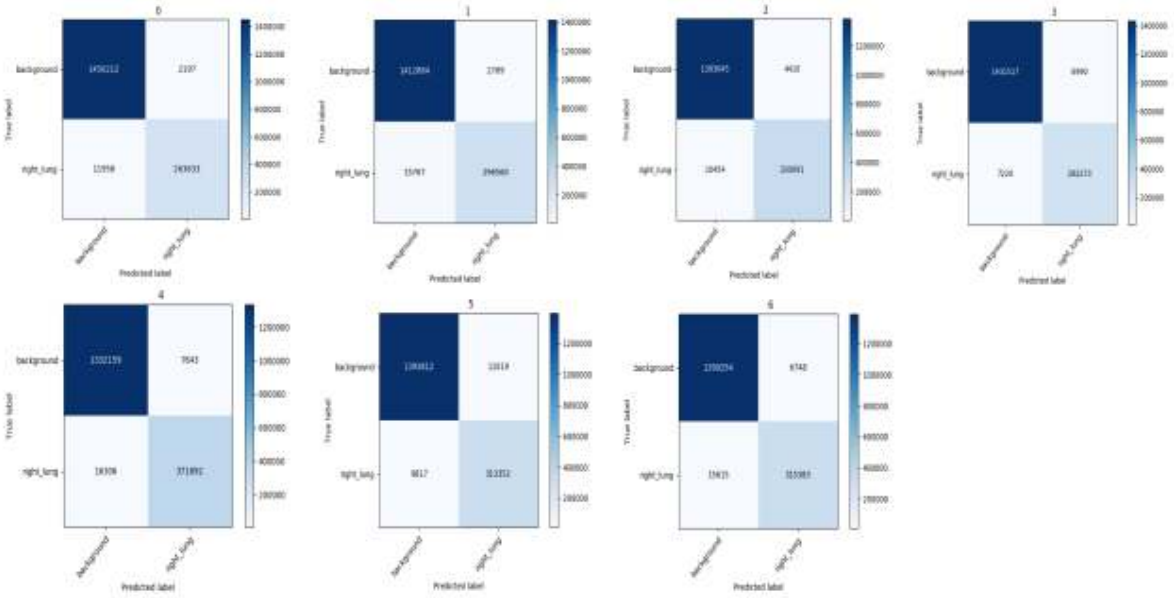


Figure 4.21: Confusion matrices for right lung using dual unet segmentation on test dataset.

Table 4.10: Table of results of right lung segmentation of proposed method and dual unet.

scan	S1-010	S1-011	S1-012	S2-010	S2-011	S2-012	S3-010
single unet right lung Dice	0.7654	0.74630	0.75426	0.90204	0.77922	0.9133	0.89849
2-stage unet right lung Dice	0.9496	0.93995	0.95466	0.94856	0.93575	0.93734	0.93239

```
Shapiro-Wilk normality test
proposed old: W = 0.78178, p-value = 0.02693
proposed new: W = 0.92904, p-value = 0.5428
reference: W = 0.927, p-value = 0.3493
```

From the output of Shapiro-Wilk normality test, the p-value > 0.05 implying that the distribution of the data are not significantly different from normal distribution so parametric t-test preconditions are satisfied for 2-stage unet and reference model. 1-stage unet model does not satisfy this condition.

```
F test to compare two variances
F = 0.0034944, num df = 6, denom df = 11, p-value = 6.114e-07
alternative hypothesis: true ratio of variances is not equal to 1
ratio of variances: 0.1578306
```

F test results conclude, that variances are statisticaly significantly different, and therefore Welch Two Sample t-test is required to compare new model with reference model.

```
Welch Two Sample t-test
t = 5.4567, df = 11.131, p-value = 9.532e-05
alternative hypothesis: true difference in means is greater than 0
mean of 2-stage unet: 0.9426213
```

```
mean of reference: 0.7199111

Wilcoxon rank sum test
alternative hypothesis: true location shift is greater than 0
2-stage vs 1-stage paired
V = 28, p-value = 0.007813
```

Statistical tests show, that 2-stage unet is, with confidence level 0.95, statisticaly significantly better than reference model, and nonparametric paired wilcoxon test shows, that 2-stage unet performs better than single stage, that was also proven to perform better than reference model on dice score measurement.

4.10 Unsuccessful experiments

4.10.1 Fine-tuning using Active Contour

To further improve satisfactory results of Unet segmentation, we tried to apply 2D active contour algorithm on slices using contours extracted from previous step. Even though this method leads to oversimplification, it can also improve results of other segmentation methods, as a fine-tuning approach(as demonstrated in subsection Active Contour initialized by thresholding).

For experiment we chose heart as segmented organ. Since Unet segmentation results, were not perfect, in some slices, heart was undersegmented. To ensure, that all contour points are on the outside of organ so that active contour altgorithm could encapsulate the organ (see subsection 2.3.3), we performed binary dilation on the mask, using diamond operator with size 10.

For contour extraction from binary mask we used *marching squares* algorithm implementation provided by scikit[7]. For *active countour* we used implemetation from scikit[1] aswell. It is configured to stop, when result doesn't change after next iteration.

Stretching contours

First configuration was, unlike any others done on eroded, instead of dilated mask, so we could see, if snake would actually encapsulate organ from the inside by stretching, instead of contracting.

Algorithm was configured:

- to be attracted to the dark areas (w_{line} = -0.5), since lungs, encompassing hearth have much lower brightness values on CT image.
- to be attracted to edges ($w_{edge} = 2$)
- to prefer smooth shape ($beta = 0.1$), to avoid taking every noice into segmentation
- not to contract ($alpha = 0.0$), because we need stretching, not contraction

[7] https://scikit-image.org/docs/0.8.0/api/skimage.measure.find_contours.html

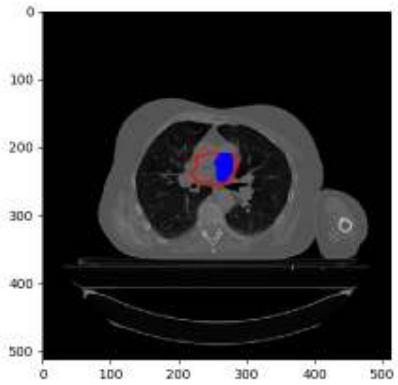


Figure 4.22: Results of stretching configuration of active contour postprocessing. Red is eroded contour from Unet segmentation, blue is results of active contour algorithm.

Results of this configuration did not prevent the active contour algorithm to contract, even with contraction disabled. It kept the number of points in contour, but put them very close one to each other. As can be observer on image 4.22, contours encapsulated one of the ventricles.

Contracting contours

Second configuration attempt was performed on dilated mask contours. Algorithm configuration:

- be attracted to light areas ($w_{line} = 0.5$), since we are moving from dark lungs, to lighter heart.
- be attracted to edges ($w_{edge} = 2$)
- prefer edges over smoothness of line ($beta = 0.01$)
- contract very slowly ($alpha = 0.1$)

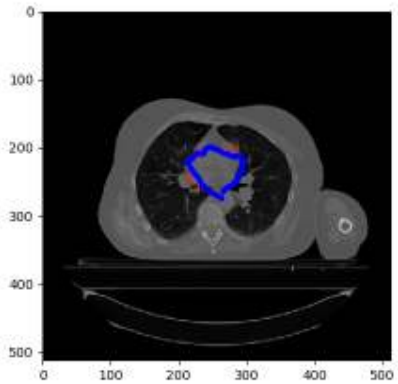


Figure 4.23: Results of contracting configuration of active contour postprocessing. Red is eroded contour from Unet segmentation, blue is results of active contour algorithm.

Results of this configuration are observably superior to previous configuration, however, as can also be seen by naked eye, do not encapsulate hearth satisfyingly and results are worse after applying the method.

4.10.2 Fine-tuning results using Fully Connected Layer

Proposed solution so far only worked with image, not using any prior anatomical or positional knowledge. Since dicom standard contains metadata as image position and image orientation[8], it seemed rational, to use this data to further augment the results of unet, e.g completely eliminate false positive matches in anatomicaly wrong areas, such as heart in abdominal cavity, based on positional information.

Each slice of dicom scan contains coordinate in cranio-caudal direction, coordinate of top-left voxel in slice in transversal plane, and we know how many voxels there are in each row and column. Voxel spacing in each direction is also provided. With this information, there is no problem to augment each voxel with positional information in form of x, y, z coordinate, even after rescaling, since only first and last voxel coordinates are necessary to extrapolate all others.

We therefore *concatenated* positional information for each voxel output with last feature map of unet. This meant having 5D vector for each voxel.

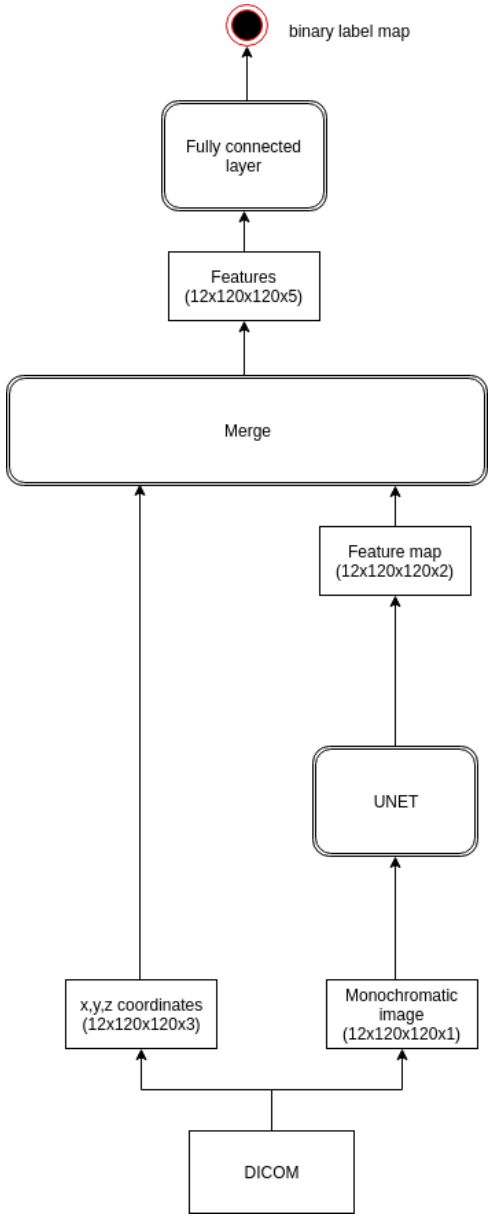


Figure 4.24: Schema of processing data with positional data concatenated to results of Unet.

[8] http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_C.7.6.2.html#sect_C.7.6.2.1.2

Concatenad data had to be flattened and was fed to Fully Connected layer(see image 4.24) with size of *patch size * patch size * patch depth* and each neuron took vector of 5 values - each took single voxel. Output is the single value for each voxel. Activation function is sigmoid, since it is between zero and one - perfect for binary segmentation, and opposed to step function, that was also considered, it is differentiable.

As loss function we used Mean Squared Error (MSE) of predictions against binary label mask desired. As trainer we used AdamOptimizer implemented in tensorflow [3].

We trained both Unet and Fully Connected Layer(FCL) together and separately - first Unet and second FCL, but both approaches yielded same results - completely random noise as segmentation (see image 4.25).

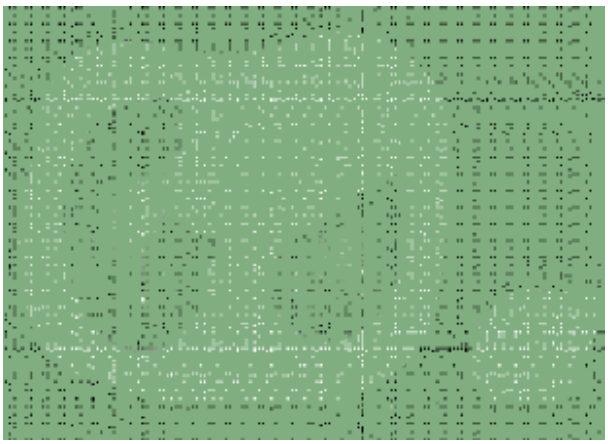


Figure 4.25: Results of segmentation using tandem Unet-Fully Connected Layer. Completely random noice on the output.

Even though this approach sound as reasonable, we later discovered, that DICOM metada are not reliable, because cranio-caudal positioning can be reversed, and top left slice coordinate just means first voxel transmitted from scanner, not the top left hand position of image of patient laying on back. There should be another field of metada that defines patient rotation, that could allow to normalise the coordinates for each patient, independently of scanner, however not every scan has one.

Chapter 5

Summary

We proposed an automated segmentation method for organs in thoracic cavity based on 3D Unet, which is a type of Convolutional Neural Network, that uses forward passes and concatenation of features from previous convolution layers to preserve much higher amount of details, because features from previous layers are guaranteed to be less downscaled, in which process details are lost.

Presented method uses preprocessing, that should allow segmentation of thoracic organs scanned by scanners of different manufacturers, with different brightness configurations, because all input data is normalized based on mandatory metadata provided by manufacturers, and rescaled to supported shape.

Multiple configurations were tested, each ceiled by the limits of the available hardware. We tried training network using multiple smaller patches of data, with higher details and full 3D scan with downscaled resolution but with more context, and it appears that for convolutional neural network, the latter approach seems much more feasible, as it yields significantly better segmentation resuls.

We evaluated proposed single 3D Unet method and for heart and left lung it shows statisticaly significantly better performance than reference model (current best on AAPM Thoracic autosegmentation challenge) using Dice score as a metric. For the rest of the organs in thoracic cavity (right lung, spinal cord and esophagus) it has proven not to be statisticaly significantly better nor worse than reference model.

The loss function we use for training is custom built, composed of 2 parts. First part is regular 1 - Dice score metric, trainer minimizes, and another part is punishment based on distance of false positive results from nearest ground truth positive value. Since organs are compact and each is present only once (left and right lungs are treated as separate organs) this approach motivates network not to similar organ patterns outside correct anatomical region.

Afterwards we used segmentation results from 3D Unet as crude contours for thoracic organs, and crop out the relevant section from scan as detailed input for another 3D Unet, that provides more precise segmentation around the borders of organs. This method has proven to be *statisticaly significantly better than both reference model and previously proposed method* for every tested organ (heart, left lung, right lung) compared on Dice score.

2-stage unet requires further evaluation for spine and esophagus to be fully comparable to both reference model and single stage unet. We would also like to make submission to AAPM Thoracic autosegmentation challenge to receive further feedback for proposed method, from independent arbiter.

Chapter 6

Resumé

Motivácia

Detekcia a určenie presného ohraničenia orgánov ľudského tela z medicínskych snímkov z CT prístrojov, je dôležitým nástrojom pre diagnostiku, ako aj na predoperačnú prípravu. Jednou z motivácií pre automatickú segmentáciu orgánov je úspora času kognitívnej záťaže expertov.

Algoritmy na detekciu primárnych, alebo sekundárnych nádorov, napríklad v pečeni, alebo pľúcach, potrebujú, aby bol najskôr vysegmentovaný samotný orgán, na ktorom majú detekciu vykonať. Podobne aj namierenie lúča ionizujúceho žiarenia pri rádioterapií vyžaduje presnú segmentáciu, aby sa dokázalo minimalizovať poškodenie zdravého tkaniva. Túto segmentáciu musia rádiológovia robiť ručne, rez za rezom. Aj keď plne automatické metódy už dosahujú *state of the art* výkonnosť, stále niesú dostatočne presné a spoľahlivé, na využitie v klinickej praxi, a používajú sa najmä interaktívne metódy [3].

Cieľom je teda poskytnúť nástroj na automatickú segmentáciu orgánov na neoznačených snímkach z CT prístroja, ktorého výsledky by sa následne použiť na ďalšie spracovanie špecifickými algoritmami.

Analýza

Počítačová tomografia

Pri tradičnom Rontgene sa stráca veľké množstvo informácií, kvôli tomu, že sa trojrozmerné ľudské telo premieta na dvojrozmerný foto-papier. Preto pán Hounsfield prišiel s myšlienkou, že vykonaním viacero meraní z rôznych uhlov by sa dali zrekonštruovať rôzne formácie (orgány) mäkkých tkanív [6].

Moderné prístroje fungujú na princípe helikálneho skenu, kde oblúk detektora je vlastne stacionárny kruh detektorov s rontgenovou lampou, ktorá krúži okolo detektora a pacient je plynule preťahovaný skrz na pohyblivej podložke. Tento prístup umožňuje ovládať hrúbku rezov, a to reguláciou rýchlosti pohybu podložky s ležiacim pacientom, ako aj rýchlosťou rotácie rontgenky (viz obr. 2.2).

Hodnota, ktorú skener nameria, je absorbcia žiarenia tkanivami tela, ktorá je rozdielom medzi intenzitou žiarenia rontgenky a hodnotou zachytenou na snímačoch. Táto hodnota sa dá rozpočítať medzi jednotlivé voxely ako suma parciálnych absorbcií, čím dostaneme N rovníc o N neznámych, čo sa dá jednoducho vypočítať.

Výzvy pri segmentácií orgánov

Orgány v dutinách ako sú hrudná, alebo brušná, prípadne panvová, môžu byť deformované, alebo poposúvané v náhodných smeroch. Tieto zmeny sú spôsobené množstvom faktorov, ako dýchanie, činnosť srdca, naplnenosťou mechúra, alebo žalúdka, prípadne mechanickým poškodením (pneumothorax, ascites).

Výzvou je aj to, že orgány, ako mäkké tkanivá, majú veľmi podobné, až totožné hodnoty jasu meraného skenermi. Problémom teda je, že textúra orgánov sa bez znalosti tvaru (ktorý je deformovateľný

pri mäkkých orgánoch), prípadne anatomickej polohy, sa dá len veľmi ťažko, alebo nedá vôbec rozoznať.

Najväčšou výzvou je však nedostatok dostupných anotovaných dát s orgánmi z nejakej telesnej dutiny. Ručné anotovanie expertom je totiž časovo veľmi náročné a tým pádom aj nákladné.

Tradičné metódy segmentácie

Práhovanie je metóda, ktorá funguje na predpoklade, že objekty v popredí sa dajú oddeliť od pozadia na základe hodnoty jasu. Tkanivá ako kosti, alebo svaly, majú vyššie hodnoty hustoty a teda jasu ako mäkké tkanivá, alebo vzduch. S použitím dvojitého prahovania, teda orezania hodnôt z hora aj z dola, sa dajú vysegmentovať útvary tvorené špecifickými hodnotami hustôt. Jednotlivé orgány majú hodnoty týchto hustôt popísané v tabuľke 2.2. Na segmentáciu samotnú sa veľmi nehodí, ale dá sa použiť ako inicializácia pre iné metódy.

Metóda rastúcich regionónov je založená na predpoklade, že voxely jedného orgánu sa nachádzajú pri sebe sú si podobné. Vychádza z jedného zadaného vodu a snaží sa v okolí hladať podobné voxely a postupne okolie rozširovať. Nevýhodou je, že vyžaduje inicializáciu a teda je to poloautomatická metóda.

Metóda aktívnych kontúr funguje na pricípe minimalizácie dvojzložkovej energie, ktorá je tvorená tvarom kontúr a rozdielom intenzít. Metóda sa inicializuje jednoduchou kontúrou, ktorá sa snaží postupnou iteráciou priľnúť ku hranám na obrázku. Nevýhodou je, že vyžaduje inicializáciu.

Metódy založené na strojovom učení

Neurónové siete tvoria základ pre väčšinu metód strojového učenia. Skladajú sa z veľkého množstva neurónov s horizontálnou (v rámci jednej vrstvy) a vertikálnou (medzi viacero vrstvami) štruktúrou. Každý neurón má svoju aktivačnú funkciu, sadu váh a bias.

Konvolučné neurónové siete majú vrámci celej siete zdieľané váhy. To znamená, že model sa nepotrebuje dookola učiť tie isté detektory pre rovnaký objekt, ktorý sa vrámci jedného obrázka vyskytuje viackrát. Na každej vrstve sa robí sada konvolúcií s množinou kernelov, a každá konvolúcia vygeneruje novú množinu príznakov. Tento proces sa opakuje v každej ďalšej vrstve.

U-Net

Unet je typ plne konvolučnej siete, ktorá ale namiesto toho, aby robila neustále podvzorkúvanie, robí aj spätné škálovanie nahor, čo znamená, že výstup zo siete ma oveľa vyššie rozlíšenie.

Sieť sa skladá z 2 hlavných častí:

- kontraktívna časť: vykonáva konvolúciu, nasledovanú ReLu a podvzorkovaním, podobne ako klasická konvolučná neurónová sieť.
- expanzívna časť: vykonáva škálovanie nahor a up-konvolúciu, ktorá znižuje počet máp príznakov.

Tento typ siete sa dá jednoducho previesť na sieť, ktorá vie pracovať s 3D snímkami, nahradením operácií ich 3D ekvivalentami [25].

Predspracovanie dát

Orezávanie objemu snímok, pokiaľ ho je možné robiť automatizovane, sa veľmi oplatí, pretože dokážeme znížiť pamäťové požiadavky a použiť vyššie rozlíšenie snímok, alebo značne skrátiť požiadavky na dobu trénovania a zvýšiť presnosť segmentácie, keď sa odstránia nepotrebné dáta.

Nie všetky CT prístroje vracajú snímky v normalizovaných tabuľkových hodnotách, tzv. Hounsfieldových jednotkách. Aby sme sa vyvarovali tohoto problému, je nutné všetky dáta preškálovať na Hounsfieldove jednotky a to prostredníctvom metadát uvedených pri každej snímke.

Opis datasetu

Dáta pochádzajú z výzvy AAPM Thoracic Auto-segmentation a obsahujú CT skeny z 3 štúdií, každá s 12 pacientami. Dáta sú vo formáte DICOM a sú ručne anotované rádiológom. Tieto anotácie predstavujú základnú pravdu pre vyhodnocovanie úspešnosti navrhnutej metódy.

Snímky majú rozlíšenie 512x512x140 voxelov a zväčša sú zachytené regióny od stavca C2 až po stavec L2. V transverzálnom reze je vzdialenosť voxelov 0.976mm a šírka rezu je 3mm.

Rozšírenie dátovej množiny

Na zmiernenie problémov, ktoré vyplývajú z malej dátovej vzorky, ako je preučenie neurónovej siete, je potrebné pôvodný dataset rozšíriť o umelé dáta. Takéto rozšírenie slúži ako forma regularizácie a bráni sieti, aby sa naučila požadované výstupy naspamäť.

Používame 3 typy augmentácií, kde každá ma svoju vlastnú pravdepodobnosť, že bude aplikovaná na práve spracúvanej snímke. Samostatná pravdepodobnosť pre každý typ rozšírenia znamená, že sa ďalej zvyšuje variancia možných vygenerovaných nových snímok, pretože nie každý typ sa musí aplikovať všade zároveň. Každá a z augmentácií, okrem manipulácie odtieňov šedej, sa musí 1:1 vykonať aj na anotáciách.

Typy použitých rozšírení:

- Rotácia: Pravdepodobnosť vykonania rotácie je 0.6. Rotácia sa vykonáva pozdĺž prvých dvoch osí, čo znamená rotácie v sagitálnej a transverzálnej rovine. Uhol rotácie je náhodné celé číslo v rozpätí -10 a 10 stupňov.
- Priblíženie: Pradepodobnosť vykonania priblíženia je 0.5. Jedná sa o rovnakú funkciu priblíženia, aká sa používa aj pri predspracúvaní údajov. Faktor mierky je náhodne vygenerovaný v rozpätí 1.0 až 1.2. Snímku je po priblížení nutné orezať na požadovaný tvar.
- Manipulácia odtieňov šedej: Pravdepodobnosť vykonania manipulácie je 0.6. Manipulácia odtieňov šedej znamená mierne zmeny hodnôt jasu voxelov.

Navrhnuté riešenie: adaptácia siete 3D Unet

V niekoľkých posledných výzvach na segmentáciu, bol 3D Unet úspešne použitý. Či už šlo o úlohy ako segmentácia pankreasu [27], prostaty [20], obličiek [25], štruktúry nervových buniek v elektrónovom mikroskope [24], a mnohé ďalšie.

Naše riešenie vychádza z implementácie Imrana Ahmeda, ktorá je dostupná na githube[1]. Toto riešenie je pôvodne použité na segmentáciu prostaty, ale s malými úpravami sa dá na ňom ďalej stavať.

Architektúra UNetu sa skladá zo 4 vrstiev, označených 0-3, kde každá, okrem poslednej, sa skladá z kontraktívnej a expanzívnej časti. Pred každým vstup do novej vrstvy sa vykonáva *dropout*, ktorý slúži ako forma regularizácie a zabraňuje preučeniu.

Príznaky z poslednej vrstvy sa dajú previesť na binárnu masku orgánu, s použitím funkcie argmax a následne s pomocou *one-hot* kódovania preložiť na samotnú segmentáciu. Prípadne sa tieto príznaky dajú ešte použiť na ďalšie spracovanie, napríklad v ďalšej neurónovej sieti.

Na trénovanie používame dáta s fixnou veľkosťou 240x240x60 voxelov. Z každej trénovacej masky sa ešte vypočítava mapa vzdialeností (distance transform map), ktorá sa ďalej používa pri výpočte funkcie ceny. Ako optimalizačnú funkciu používame Adam-a ktorý je implementovaný v knižnici tensorflow[3].

Metrika na vyhodnocovanie je Dice skóre, ktoré je vyjadrené ako podiel prieniku a zjednotenia segmentácie a základnej pravdy, poskytnutej v anotáciách.

Funkcia ceny je tiež Dice skóre, ale je obohatené o tresty za nepravdivo pozitívne nálezy v závislosti od vzdialenosti správnej pozície orgánu. Na tento výpočet sa používajú informácie z mapy vzdialeností.

Podmienkou zastavenia tréningu je, aby sa už výrazne nemenila vyhodnocovacia metrika, alebo sa dosiahlo 10 000 behov. Dosiahnutie druhej podmienky trvá približne 30 hodín.

Vyhodnotenie 2-úrovňovej siete Unet

Výstupy z Unet-u, tak ako je popísaný vyššie, sa dajú použiť na orezanie vstupných dát, a takto orezané dáta sa dajú vložiť do ďalšieho Unet-u na spresnenie výsledku.

Výsledky takéhoto postupu sme postupne štatisticky vyhodnocovali, a zisťovali sme, či sú štatisticky významne lepšie ako pôvodná jednoúrovňová sieť, resp. či sú významne lepšie ako referenčný model, opísaný v tabuľke 4.2.

1-úrovňový Unet sme vyhodnotili na orgánoch: srdce, ľavé pľúca, pravé pľúca, miecha, pažerák. Pre srdce a ľavé pľúca vychádza jednoúrovňový unet, na hladine významnosti 0.95 dosiahnuté Dice skóre ako štatisticky významne lepšie. Pre ostatné orgány nevyšlo dosiahnuté skóre ani ako lepšie ani ako horšie.

2-úrovňový Unet sme vyhodnotili na orgánoch: srdce, ľavé pľúca, pravé pľúca. Pre všetky orgány nám vyšlo, že pre 2-úrovňový unet je dosiahnuté dice skóre na hladine významnosti 0.95 štatisticky významne lepší ako referenčný najlepší model na výzve, ako aj ako 1-úrovňový Unet.

Bibliography

[1] P. Hu, F. Wu, J. Peng, Y. Bao, F. Chen, and D. Kong, "Automatic abdominal multi-organ segmentation using deep convolutional neural network and time-implicit level sets," *International Journal of Computer Assisted Radiology and Surgery*, vol. 12, no. 3, pp. 399–411, 2017.

[2] E. Goceri and E. Martinez, "Artificial neural network based abdominal organ segmentations: A review," *Proceedings - 2015 IEEE 14th International Conference on Machine Learning and Applications, ICMLA 2015*, pp. 1191–1194, 2016.

[3] G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, and T. Vercauteren, "Interactive Medical Image Segmentation using Deep Learning with Image-specific Fine-tuning," *IEEE Transactions on Medical Imaging*, pp. 1–12, 2018.

[4] Y. Li, J. Yao, and D. Yao, "Automatic beam angle selection in IMRT planning using genetic algorithm Related content Genetic algorithm based deliverable segments optimization for static IMRT," *Physics in Medicine & Biology Yongjie Li et al Phys. Med. Biol*, vol. 49, 2004.

[5] D. Bartušek, *Diagnostické zobrazovací metody.* Masarykova Universita v Brně Lékařská fakulta, 2004.

[6] L. W. Goldman, "Principles of CT and CT Technology," *Journal of Nuclear Medicine Technology*, no. September, pp. 115–129, 2009.

[7] L. Navrátil, J. Rosina, *et al.*, "Medicínská biofyzika. 1. vyd. praha: Grada, 2005," tech. rep., ISBN 80-247-1152-4.

[8] M. Moghbel, S. Mashohor, R. Mahmud, and M. I. B. Saripan, "Review of liver segmentation and computer assisted detection/diagnosis methods in computed tomography," *Artificial Intelligence Review*, pp. 1–41, 2017.

[9] F. Itwm, "Survey of 3d image segmentation methods," *Keywords image processing 3d image segmentation binarization*, vol. 123, no. 123, pp. Kaiserslautern, Germany, 2007.

[10] R. Molteni, "Prospects and challenges of rendering tissue density in Hounsfield units for cone beam computed tomography," *Oral Surg Oral Med Oral Pathol Oral Radiol*, vol. 116, pp. 105–119, 2013.

[11] R. Pohle and K. D. Toennies, "Segmentation of medical images using adaptive region growing,"

[12] E. Smistad, T. L. Falch, M. Bozorgi, A. C. Elster, and F. Lindseth, "Medical image segmentation on GPUs A comprehensive review," *Medical Image Analysis*, vol. 20, pp. 1–18, 2015.

[13] M. Debakla, K. Djemal, and M. Benyettou, "Influence of noise distribution on active contour models: Medical images segmentation," *Asian Journal of Applied Sciences*, vol. 4, no. 2, pp. 101–111, 2011.

[14] T. Okada, M. G. Linguraru, Y. Yoshida, M. Hori, R. M. Summers, Y.-w. Chen, and N. Tomiyama, "Abdominal multi-organ segmentation of CT images based on hierarchical spatial modeling of organ interrelations," *Proceedings of Abdominal Imaging*, pp. 173–180, 2011.

[15] T. Heimann and H.-P. Meinzer, "Statistical shape models for 3D medical image segmentation: A review," 2009.

[16] Y. Boykov and O. Veksler, *Graph Cuts in Vision and Graphics: Theories and Applications*, pp. 79–96. Boston, MA: Springer US, 2006.

[17] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A Survey on Deep Learning in Medical Image Analysis," no. 1995, 2017.

[18] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber, "Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation," in *Advances in Neural Information Processing Systems*, pp. 2998–3006, 2015.

[19] H. R. Roth, H. Oda, Y. Hayashi, M. Oda, N. Shimizu, M. Fujiwara, K. Misawa, K. Mori, and H. R. Roth, "Hierarchical 3D fully convolutional networks for multi-organ segmentation Hierarchical 3D fully convolutional networks," 2017.

[20] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation,"

[21] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks,"

[22] H. Chen, Q. Dou, L. Yu, and P.-A. Heng, "VoxResNet: Deep Voxelwise Residual Networks for Volumetric Brain Segmentation,"

[23] S. Andermatt, S. Pezold, and P. Cattin, "Multi-dimensional Gated Recurrent Units for the Segmentation of Biomedical 3D-Data," in *Deep Learning and Data Labeling for Medical Applications* (G. Carneiro, D. Mateus, L. Peter, A. Bradley, J. M. R. S. Tavares, V. Belagiannis, J. P. Papa, J. C. Nascimento, M. Loog, Z. Lu, J. S. Cardoso, and J. Cornebise, eds.), (Cham), pp. 142–151, Springer International Publishing, 2016.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, 2015.

[25] O. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation,"

[26] A. Canziani, E. Culurciello, and A. Paszke, "An analysis of deep neural network models for practical applications," tech. rep.

[27] O. Oktay, J. Schlemper, L. Le Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. Mcdonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, "Attention U-Net: Learning Where to Look for the Pancreas," tech. rep.

[28] S. Ruder, "An overview of gradient descent optimization algorithms," 2016.

[29] C. Wang, T. MacGillivray, G. Macnaught, G. Yang, and D. Newby, "A two-stage 3D Unet framework for multi-class segmentation on full resolution image," 2018.

[30] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," tech. rep.

[31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC Superpixels," tech. rep.

Attachments

Contouring Guidelines

Esophagus

- Standard name: Esophagus
- RTOG Atlas description: The esophagus should be contoured from the beginning at the level just below the cricoid to its entrance to the stomach at GE junction. The esophagus will be contoured using mediastinal window/level on CT to correspond to the mucosal, submucosa, and all muscular layers out to the fatty adventitia.
- Additional notes: The superior-most slice of the esophagus is the slice below the first slice where the lamina of the cricoid cartilage is visible (+/- 1 slice). The inferior-most slice of the esophagus is the first slice (+/- 1 slice) where the esophagus and stomach are joined, and at least 10 square cm of stomach cross section is visible.

Heart

- Standard name: Heart
- RTOG Atlas description: The heart will be contoured along with the pericardial sac. The superior aspect (or base) will begin at the level of the inferior aspect of the pulmonary artery passing the midline and extend inferiorly to the apex of the heart.
- Additional notes: Inferior vena cava is excluded or partly excluded starting at slice where at least half of the circumference is separated from the right atrium.

Lungs

- Standard names: Lung_L, Lung_R
- RTOG Atlas description: Both lungs should be contoured using pulmonary windows. The right and left lungs can be contoured separately, but they should be considered as one structure for lung dosimetry. All inflated and collapsed, fibrotic and emphysematic lungs should be contoured, small vessels extending beyond the hilar regions should be included; however, pre GTV, hilars and trachea/main bronchus should not be included in this structure.
- Additional notes: Tumor is excluded in most data, but size and extent of excluded region are not guaranteed. Hilar airways and vessels greater than 5 mm (+/- 2 mm) diameter are excluded. Main bronchi are always excluded, secondary bronchi may be included or excluded. Small vessels near hilum are not guaranteed to be excluded. Collapsed lung may be excluded in some scans. Regions of tumor or collapsed lung that are excluded from training and test data will be masked out during evaluation, such that scores are affected by segmentation choices in those regions.

Spinal cord

- Standard name: SpinalCord
- RTOG Atlas description: The spinal cord will be contoured based on the bony limits of the spinal canal. The spinal cord should be contoured starting at the level just below cricoid (base of skull for apex tumors) and continuing on every CT slice to the bottom of L2. Neuroformanines should not be included.

- Additional notes: Spinal cord may be contoured beyond cricoid superiorly, and beyond L2 inferiorly. Contouring to base of skull is not guaranteed for apical tumors.

Traditional Computer Vision Methods

To further understand classical segmentation methods, several experiments with described dataset were conducted. Primary focus of these experiments was to segment lungs as whole.

Thresholding lungs using Hounsfield units

Hounsfield units(HU), as previously mentioned, are standardized units of tissue density as it measured by Computer Tomography. Depending on type of examination and the needs of the radiologist, it may be useful not to use Hounsfield units for measurement, but rather some custom scale, if using some contrast solution, etc. Data in DICOM format has stored in metadata values of *intercept* and *slope*, that can be used to rescale values into Hounsfield units (see listing 1).

```
image = slope * image
image = image + intercept
```

Listing 1: Rescaling image slice (pixel array) to HU. Each pixel has to be at least 16bit integer, because CT has much greater dynamic range, than traditional cameras.

As much as 70% of volume of image is occupied by air, most of it beeing around patient and some of it beeing in the airways and lungs. Air is followed by water with approximately 14% and the rest is actual human tissue (see image 6.1).

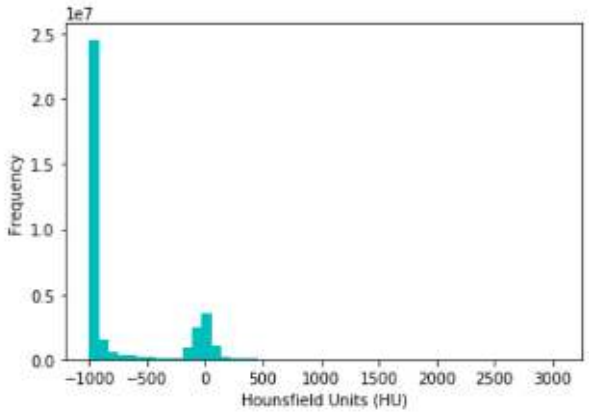


Figure 6.1: Histogram of volume density in CT image. -1000 is air, 0 is water, 700-3000 are various bone tissues

After conversion to HU, simple demonstration of thresholding is cutting off every voxel density below 350, effectively removing all soft tissue, with only bones, and parts of CT are visible (see image 6.2).

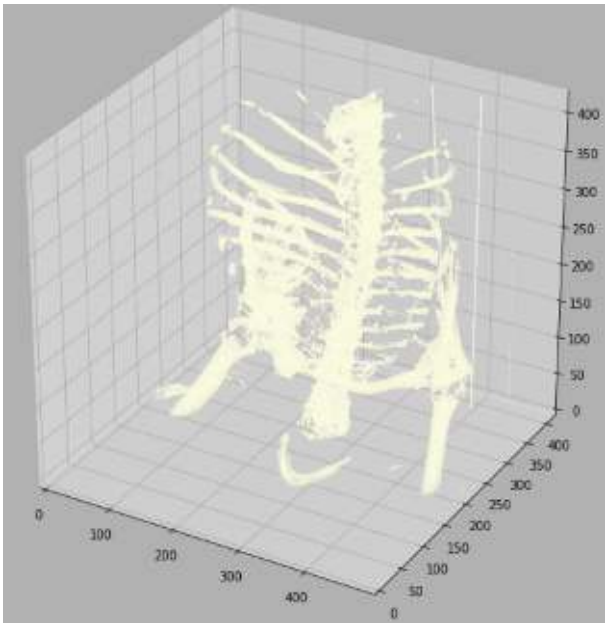


Figure 6.2: Simple thresholding with value 350, removing all soft tissues.

This rather crude method can be refined to produce much better results. Since lungs, are highly saturated with air, their HU value is much lower, than the average tissue, around -700 (see table 2.2). With classifier, such as KMeans, we can split voxels into 2 separate groups:

1. **very light** mostly air and lungs

2. **very dense** rest of the tissues

Calculating mean of the centers of those 2 groups, yields fine threshold value, since there is huge gap between dense and solid tissues (see image 6.1).

Figure 6.3: Thresholding using mean of KMeans centers.

Since lungs consist of some dense tissue as well (bronchi, blood vessels, etc.), there is huge amount of artifacts in resulting segmentation (see image 6.3). Removing them can be achieved by using simple morphological operations of erosion and dilation in series (see image 6.4).

Figure 6.4: Lung mask after erosion and dilation.

This method is very much lung specific and would be almost impossible to apply on another organ then lungs or bones, because of distribution of density of tissues, so simple classifier as KMeans cannot be used. However results looks very promising, scoring 0.6 Dice score. Much of the loss of the score can be explained by segmenting also trachea and bronchi as lungs, since they are full of air as well (see image 6.5).

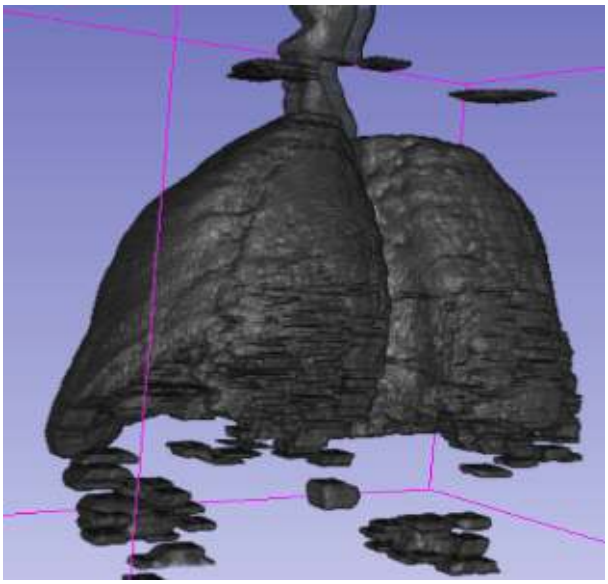


Figure 6.5: Raycasting visualisation of thresholding segmentation of lungs.

Active Contour initialized by thresholding

As seen on image 6.4, mask obtained by thresholding can be very rough around the edges. To counter this issue, it is possible to smoothen the edges by employing active contour method. This method, as described above, is minimizing energy. Total energy is composed of two parts:

1. **internal energy** or smoothness - perfect tool to counter rough edges.
2. **external energy** or metric of interest - the more interesting the are is, the lower energy it has.

This method requires initialization. It can be initialized with mask produced by thresholding described in previous section, however not directly, since active contour works with poly-lines. Conversion can be done by simple detection of bit-inversion on binary mask and ordering coordinates of these inversions into sequece of points (see image 6.6).

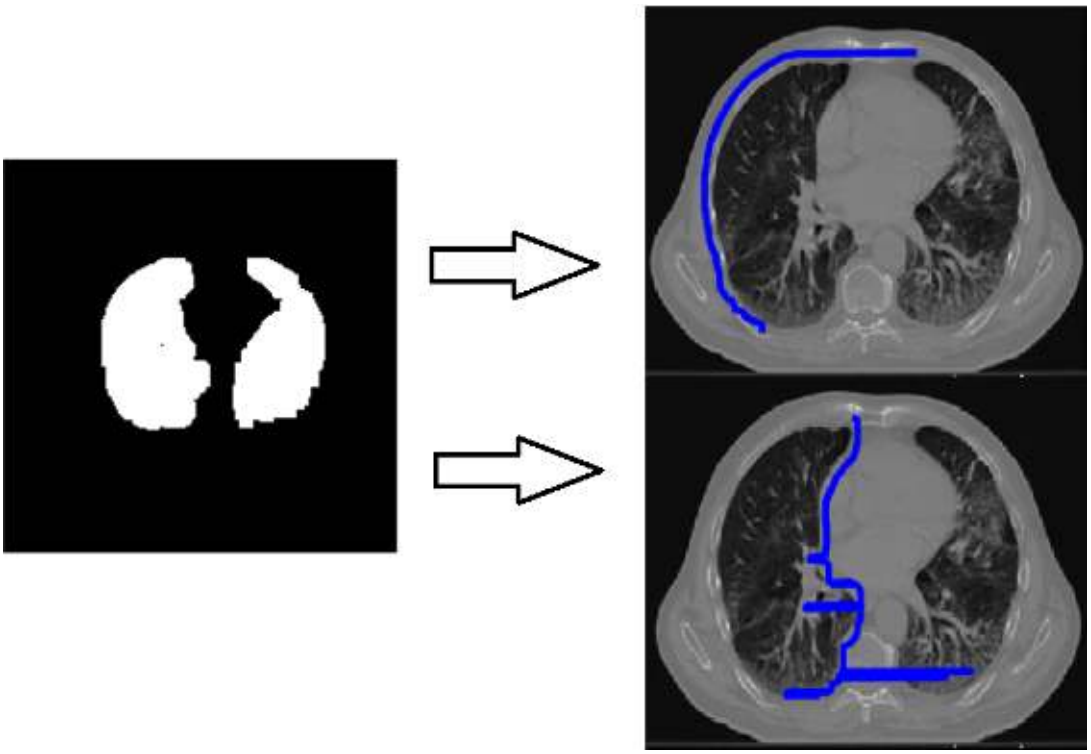


Figure 6.6: Extraction of poly lines from binary mask produced by thresholding.

As seen in picture 6.6, this method is not entirely reliable, and can produce artifacts, that highly deform the polyline. This imperfection can be mitigated by performing statistical analysis on polyline points, and removing extremes in process. Simple use of sliding window with reasonable size (e.g 10 previous and 10 next points) scanning through polyline is sufficient (see image 6.7).

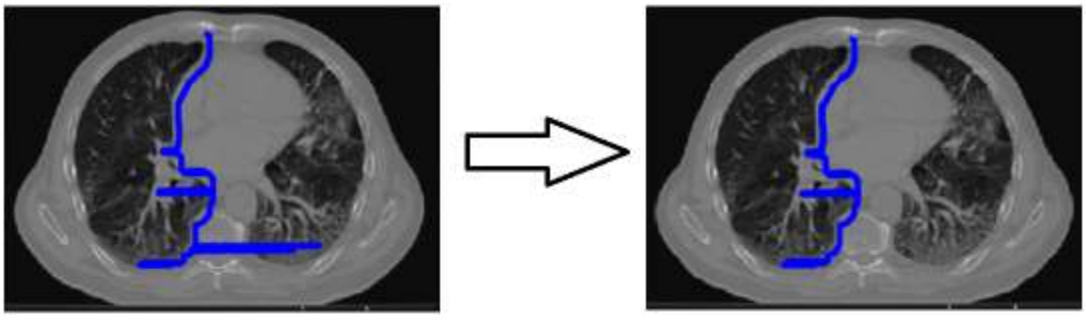


Figure 6.7: Removal of extremes using sliding window.

Extracted polylines can then be joined together to initialize the active contour algorithm (eg. `active_contour` function found in `skimage.segmentation` library). Algorith has been parametrized to prefer darker edges over lighter ones, and to prefer smoothness over precision. Preference to darkness means, that snake will more probably adhere to dark lung, than to light ribs surrounding it. Prefering smoothness further mitigates edgyness of thresholding (see image 6.8).

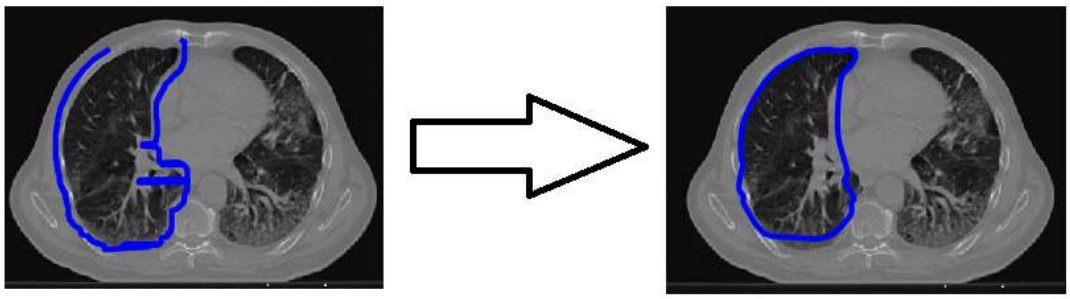


Figure 6.8: Initialization of active contour (left) and final result (right).

Active countour is fine method for segmenting object, that are not very ragged. However lungs, when dealing with 2D slices very much are. Since smoothness is a considerable energy factor, it is a problem to deal with almost fractal-like surface (orifices where bronchi penetrates lungs). This leads to oversimplification of segmentation (see image 6.9), resulting in score deficiency.

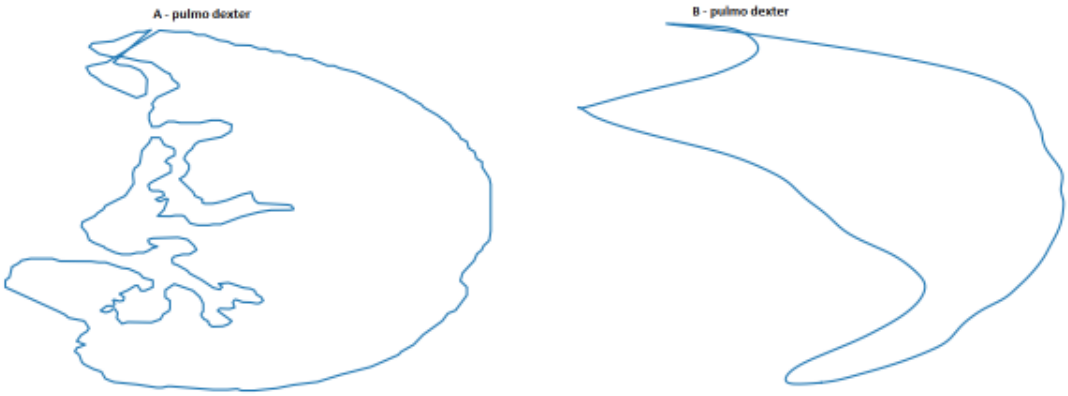


Figure 6.9: Ground truth (A) and extreme segmentation oversimplification of snake algorithm (B).

Using active contour did actually improve Dice score, on average by 5% on 130 slices of single patient and this method proves, that it is possible to seed semi-automatic segmentation method, using simpler, but automatic one.

Graph cut and SLIC

Another approach to segmentation is to construct a fully connected graph and try to disect it. Classical graph cut, using min cut algorithm requires 2 seeds. One for background and one for foreground. Since goal of this paper is to automatize organ segmentation, another approach for graph disection needs to chosen.

Instead of constructing graph of similarity of each voxel to seed points provided by user as described in previous chapeters, it is possible to construct Region Adjacency Graph, where neighbouring voxels are connected and these connections are weighted as similarity between connected nodes and then perform normalized cut, that, based on threshold configuration, cuts connections of dissimilar regions. However computational requirements are very high, as this is NP problem [30].

As there is too many nodes, even in sigle CT slice, reduction of their number can be achieved using some superpixelation algorithm. Among multiple options SLIC is considerably faster and is generally prefered for medical applications [31]. Creating 500 superpixels from single slice with dimensions 500x500 reduces number of graph nodes to square root (see image 6.10).

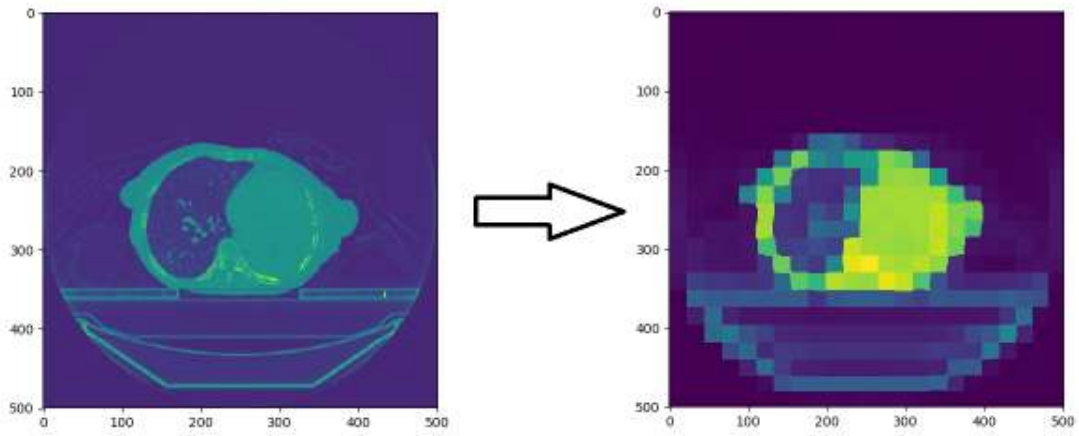


Figure 6.10: Superpixelation of input image using SLIC.

Having superpixels, graph constructions is fairly simple. Python library skimage provides good implementation, that uses region color as similarity metric [1]. Since amount of nodes is exponencialy reduced, it is really fast and it is possible to visualise these relationships, since regions are on average 20x20 px (see image 6.11).

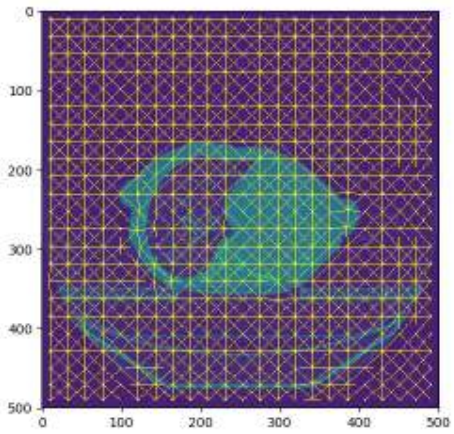


Figure 6.11: Region Adjacency Graph (RAG) based on superpixels. Thicker the edge, higher similarity between the connected regions.

Using several thresholds yields different results, however none of the results is capable of appropriate automatic organ segmentaion, because, the results are either oversegmented or multiple organs, or even surroundings are merged together (see image 6.12). However it could be useful as preprocessing method, to remove the most part of the background and therefore reduce the input image size for AI method. But using simple thresholding with knn for this use case yields same if-not-better results as well as being considerably less resource demanding to compute.

[1] `http://scikit-image.org/docs/dev/api/skimage.future.graph.html#rag-mean-color`

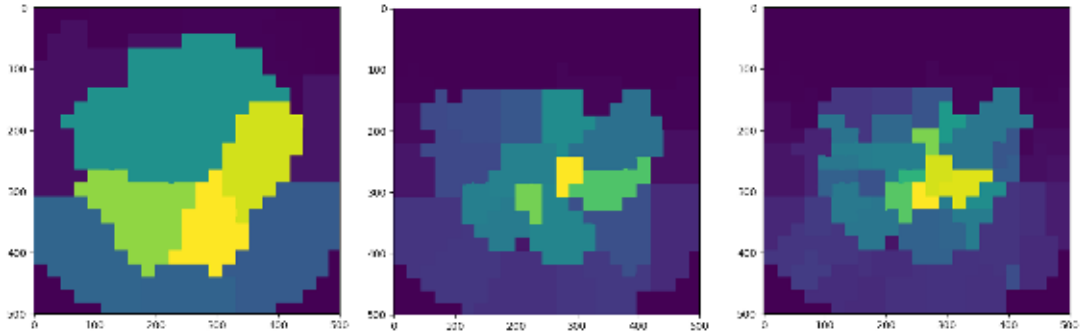


Figure 6.12: Graph cut using normal cut algorithm results with different threshold applied. From left to right the similarity criterion threshold is being leveled.

Technical documentation 3D-Unet reimplementation

U-Net is implemented in tesorflow framework, and is capable of training on GPU. It was trained on Nvidia Titan V with 12GB of VRAM. 10 000 iterations of training were performed in under 30 hours.

Defined Pipelines

Convolution-BatchNormalization-ReLu

does convolution with input volume, with 3D kernel of size 3 and stride 1 in each dimension and variable filter size (dimensionality of output space). Then batch normalization is performed. This has been proved to dramatically accelerate training of deep networks. After that, ReLu is performed. Described steps are repeated across all the layers of the network, therefore it makes sense to declare them as single step.

Compressing path

From level 0 to 2 all levels perform the same operation pattern. They all call **Convolution-BatchNormalization-Relu** operation twice. Each time this operation is called with doubled filter size, initialy starting at 8. Then dropout with rate 0.2 is perfomed and then max-pooling per slices with 2x2 pool size and the same stride.

Level 3 does not perform max-pooling on output but upconvolution - it is a transition to Expanding path.

Expansion path

From level 2 to level 0, input from lower layer is concatenated with features from complementary compressing layer and then Convolution-BatchNormalization-ReLu pipeline is performed twice, each with the same filter size as complementary compressing layer in desceding order.

Zero level does one extra final 3D convolution, with 2 filter, each for one class. These features are used for final label prediction, using argmax function.

To compute Loss Function, Dice score is computed for each softmaxed feature map, where each feature map is treated as binary segmentation of single class. As ground truth, one-hot encoded labelmaps are used.

Content of DVD media

Attached media contains one zipped file, that needs to copied to hard drive and unzipped, in order to be accesed. After unzipping, there are 3 folders:

- source - contains source code for proposed method
  - 1-stage - contains source code for 1-stage unet
    - `prepare_pickles.py` - preprocesses dicom data into `.hickle` format used in training, also does rescaling and hounsfield normalisation
    - `train.py` - starts network training. Model is stored in `../tf` folder, metrics in `../results` and intermediate results in `../training_results`. It takes 2 arguments:
      1. organ name: heart, left_lung, right_lung, spine, esophagus
      2. restore model iteration: number of model you want use for restoration, for fresh start input 0
    - `predict.py` - does prediction on test data and produces confusion matrices and IOU metrics
  - 2-stage - contains source code for 2-stage unet
    - same as 1-stage, byt with suffix 2
  - label_fixer - one time script, that fixed broken encodings
  - postprocessing - contains scripts from failed experiments
  - `upscale_results.py` - scales results from 2-stage unet back to original size, same size as corresponding DICOM images
  - `final_eval.py` - does final evaluation of upscaled results and prints metrics to `results3` folder
  - data - contains raw DICOM and nrrd data with CT scans and ground truth segmentations
  - all other folders are only placeholders for results. Missing folder may cause script failure. If that happens, please create folder metioned in error message. Zipping may remove empty folders
- experiment-data - containes results and logs of training, serialized models and segmentation results
  - [organ name] - esophagus and spine contain only first 3 folders
    - model - trained model from 1-stage unet
    - results - test data segmentation results and test metrics
    - training_results - validation data segmentation results and training metrics
    - model2 - trained model from 2-stage unet
    - results2 - test data segmentation results and test metrics
    - training_results2 - validation data segmentation results and training metrics
- thesis - source code of this thesis in org with R scripts used for results evaluation, and pdf of this document