EAST WEST UNIVERSITY

B. Sc. ENGINEERING THESIS

Liver Segmentation Using Deep Learning

Authors:

Tonmoy Mondol

(2015-2-50-030)

Syeda Nurun Nahar

(2016-1-50-020)

Ankhi Chakroborty

(2016-1-50-023)

Supervisor:

Dr. Mohammad Arifuzzaman

Assistant Professor

Department of Electronics and Communications Engineering

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science in Information and Communication Engineering

December 2019

Letter of Acceptance

This thesis entitled "Liver Segmentation using Deep Learning" submitted by Tonmoy Mondol (ID: 2015-2-50-030), Syeda Nurun Nahar (ID: 2016-1-50-020) and Ankhi Chakroborty (ID: 2016-1-50-023) to the Electronics and Communication Engineering Department, East West University, Dhaka-1212, Bangladesh is accepted as satisfactory for partial fulfilment of requirements for the degree of Bachelor of Science (B.Sc.) Information and Communications Engineering.

____

Dr. Mohammed Moseeur Rahman

Assistant Professor Chairperson

Department of Electronics and Communications Engineering

East West University

____

Dr. Mohammad Arifuzzaman

Assistant Professor

Department of Electronics and Communications Engineering

East West University

Declaration of Authorship

We, Tonmoy Mondol, Syeda Nurun Nahar and Ankhi Chakroborty, declare that this thesis titled, "Liver Segmentation using Deep Learning" supervised by Dr. Mohammad Arifuzzaman and the work presented in it are our own. We confirm that:

This work has done wholly while in candidate for Bachelor of Science in Information and Communication Engineering degree at this University. Where any part of this thesis has done previously by others have clearly cited. Where we have consulted the published work of others, this is always clearly attributed. Where we have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely our own work. We have acknowledged all main sources of help. Where the thesis is based on work done by ourselves jointly with others, we have made clear exactly what was done by others and what we have contributed ourselves.

Tonmoy Mondol
2015-2-50-030

Syeda Nurun Nahar
2016-1-50-020

Ankhi Chakroborty
2016-1-50-023

Abstract

Throughout the most recent couple of years, major breakthroughs were achieved in many computer visions tasks, such as image classification and segmentation by using the application of deep learning. The programmed liver division from Computed Tomography (CT) pictures has become a significant territory in clinical research, including radiotherapy, liver volume estimation, and liver transplant medical procedures. This research introduces a framework for the automated segmentation of liver and lesions in images of the CT and Magnetic Resonance Images (MRI) abdomen using Cascaded Fully Convolutional Neural (CFCN) networks for the segmentation of large-scale medical trials and quantitative image analysis. We train and cascade two Fully Convolutional Neural (FCN) networks for the combination of liver segmentation and lesions. We train an FCN as a first step to segment the liver as an input of Region of Interest (ROI) for a second FCN. The second FCN segments only lesions inside phase one's estimated liver ROIs. CFCN models have been trained on a 100-volume abdominal CT dataset. Validation tests on additional data sets show that linguistic liver and lesion segmentation based on CFCN achieves Dice scores above 94% for the liver with computation times below 100s per volume. On 38 MRI liver tumor volumes and the public 3DIRCAD dataset, we further experimentally demonstrate the robustness of the proposed method.

Acknowledgements

We would like to express our most sincere gratitude to our supervisor Dr. Mohammad Arifuzzaman for his patient guidance, continuous support and advice. He has always been very helpful and inspired us a lot. We would also like to express our gratitude to our honorable Chairperson and other faculty members and staff of our department. Last but not the least we are grateful to our parents who always support and inspired us.

Contents

Dedicated to Our Parents …

Chapter 1

Introduction

1.1 Motivation

Since the liver is a solid organ radiological method can identify the position and size of the liver [1]. The segmentation of medical image and separating the pixels of organs or scratches from the medical background images like CT and MRI, is one of the most challenging parts for providing critical information about the shapes and volumes of these organs. CT and MRI are increasingly significant for introducing the finding and improvement of essential and auxiliary tumor malady from the abnormality in the shape and surface of the liver and likely scratch[2].
Key cancers like breast, colon, pancreatic cancer spread conversion to the liver across the course of the disease. So, in primary tumor treatment liver and its scratches are routinely analyzed. The liver tumor is also a site of Hepatocellular carcinoma (HCC) which is the most common cause of death of people in recent years [3].

Manual and semi-manual segmentation methods are used in medical activity to view the CT and MRI images to obtain the liver diagnosis. However, this method is time consuming, subjective and operator-dependent. But the patient's health was at risk. Then, for a long time, deep learning techniques to remove hand-crafted features became a dominant technique. The design and production of these features has always been the primary concern for the development of such a program, and the complexity of these solutions have been seen as a major weakness for their implementation [4].

1.2 Organization of thesis

For the task we had very small dataset of liver. To extend our dataset and to train the machine more effectively we have augmented our dataset by using various augmentation techniques. Also, to get more accurate data we have preprocessed images. For our task we have modified the U-Net into “mini U-Net” to train the machine for smaller resolution image. After training the result had 0.87 dice coefficient, which means it has given us almost accurate result. We have organized the thesis by following manner:

Chapter 2: In this chapter we have discussed the literature review of this thesis. Then we have discussed about the history and development of CCN, FCN and U-net.

Chapter 3: This chapter contains necessary information about image segmentation and Masking.

Chapter 4: This chapter contains the method of our segmentation process.

Chapter 5: This chapter contains the results and discussions about the result

Chapter 6: This chapter contains conclusion and future work.

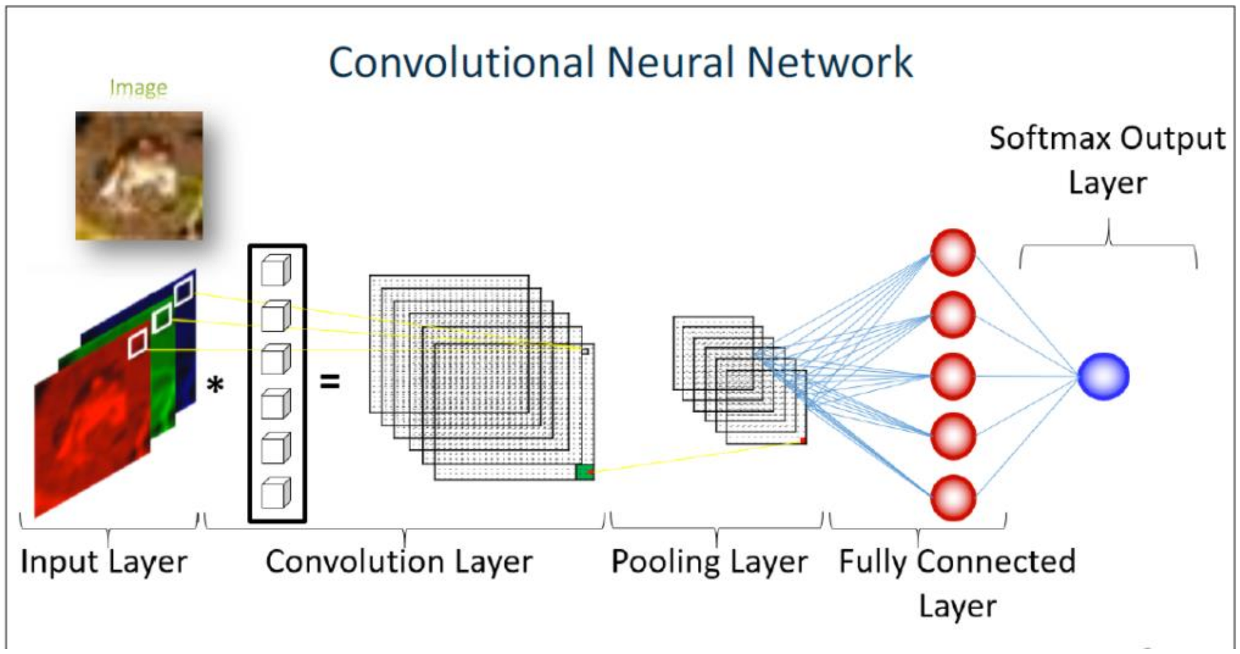
Chapter 2

Literature Review

This review highlights the tremendous achievement of Deep convolutional network, machine learning algorithms applied to medical image analysis and clinical characteristics of this field. Deep convolutional network is a pattern recognition instrument, where a neural network can automatically learn characteristics rather than traditionally hand designed characteristics, and selecting these characteristics which was a difficult task.

2.1 Convolutional neural networks (CNN)

CNNs have been used for clinical image processing since 1996 in the work of Sahiner et al [5]. A CNN is a branch of neural networks and consists of a stack of layers each appearing a specific operation, e.g., convolution, pooling, loss calculation, and so on. Each intermediate layer receives the output of the preceding layer as it enters (see Figure 2.1). The starting layer is an enter layer, that is without delay related to an enter picture with the variety of neurons identical to the range of pixels in the enter image. The subsequent set of layers are convolutional layers that gift the results of convolving a sure number of filters with the enter information and carry out as a feature extractor. The filters, normally known as kernels, are of arbitrary sizes, defined via designers, and relying on the kernel size. Each neuron responds most effective to a specific location of the preceding layer, called receptive difficulty. The output of each convolution layer is considered as an activation map, which highlights the impact of applying a particular clear out on the input. Convolutional layers are usually followed through activation layers to use non-linearity to the activation maps. The next layer can be a pooling layer relying on the layout and it helps to lessen the dimensionality of the convolution's output. To carry out the pooling, there are some strategies, which consist of max pooling and common pooling. Lastly, excessive-degree abstractions are extracted by using really connected layers. The weights of neural connections and the kernels are constantly optimized at a few degrees in the method of an again propagation within the education phase.

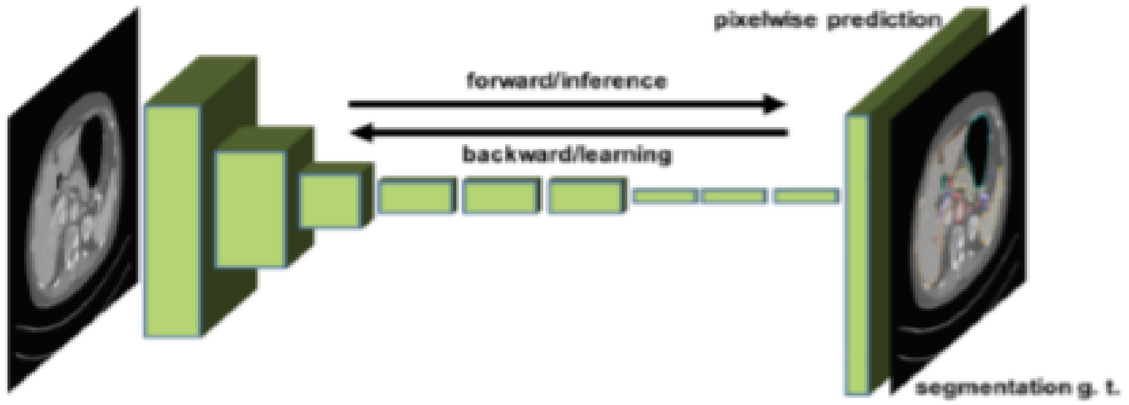


[Adapted from (S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997)]

Figure 2.1. The basic structure of CNN

2.2 Fully convolutional networks (FCN)

Fully convolutional network means that the neural network consists of convolutional layers that are usually found at the end of the network without any fully connected layers or MLP. A CNN with fully connected layers can be learned from the end to the end just as it is fully convolutional. The main difference is that learning filters everywhere is the complete coevolutionary net. Even the layers of decision making at the end. When the coevolutionary features are inserted finally into the network’s fully connected layers One downside of CNNs is that the image’s spatial information is lost. Long et al[6]. Suggested the Fully Convolutional Network (FCNs) and showed that trained end-to-end, pixel to-pixel semantic segmentation exceeded prior best outcomes without further equipment. The typical FCN setup is shown in Figure.2.2.



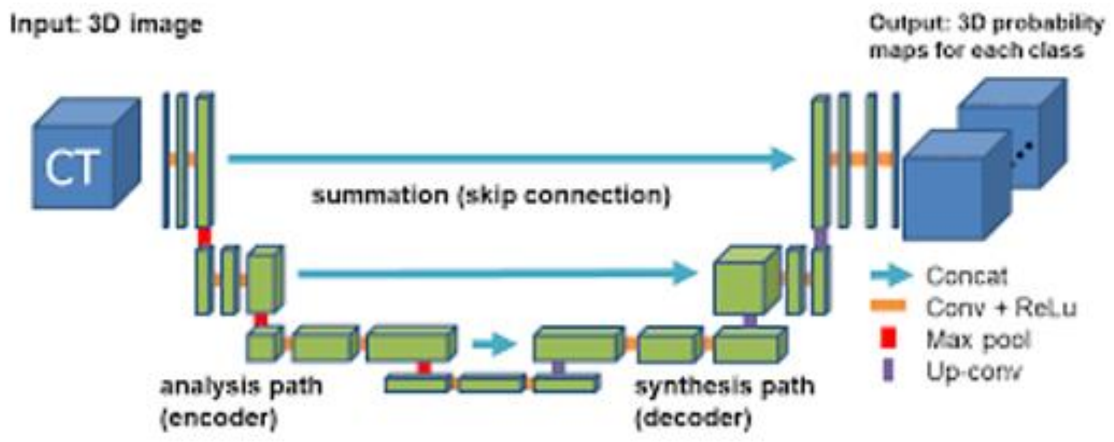
[Adapted from (H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1285–1298, 2016)]

Figure 2.2. Fully Convolutional Network (FCN)

2.3 3D Fully convolutional networks (3DU-Net)

In computer-assisted liver surgery planning, detection and delineation of the liver from abdominal 3D computed tomography (CT) images are key tasks. Nevertheless, automated and precise segmentation, in particular liver identification, remains difficult due to complex contexts, unclear borders, heterogeneous presentation and highly varied types of the liver. We propose an automated segmentation model based on 3D convolutional neural network (CNN) and globally optimized surface evolution to overcome these challenges.

In 2015, Olaf Ranneberger, Philipp Fischer and Thomas Brux proposed U-Net architecture, which is an updated version of a completely coevolutionary network which operates effectively with a small size of training samples and provides accurate results. Because of its structure's characteristic fit is commonly used in the analysis of medical images, but has recently been successfully used in non-medical areas. The main principle is to introduce successive layers to the contracting network where pooling operators are replaced by up sampling operators.rm having a downscaling path on the left and an upscaling path on the right, it is called U-net.



[Adopted from H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1285–1298, 2016.]

Figure 2.3. Architecture of U-Net.

Chapter 3

Segmentation Basics

3.1 Image Segmentation

Image segmentation is a digital image process which divides into different sections (pixel sets, also called ultra-pixels).data-reach-world[9].Image segmentation is a procedure of digital image dividing into various section(pixel set, also known as ultra-pixel).The purpose of the segmentation is to simplify simply by making the representation of a picture more meaningful and easier to understand to evaluate. Image segmentation has two objects. The first objective is to decompose the image into parts for further analysis and the second objective of segmentation is to perform a change of representation [10].

3.2 Need for Image Segmentation

Cancer has already been a fatal disease. If we don't acknowledge it early on, cancer can be fatal even in today's age of technological advances. He earliest possible detection of cancer cells can possibly save millions of lives. Cancer cell shape plays a crucial role in determining cancer seriousness [11]. You might have put the pieces together – it won't be very useful here to detect objects. Only bounding boxes will be generated that will not assist us to identify the shape of cells. The methods of image segmentation have a MASSIVE effect here. They assist us get to grips with this issue in a more granular way and get more significant outcomes. A win - win in the healthcare industry for everyone. The main goal is to find meaning from a picture whether an object is to be identified, experiences, etc. The object segmentation makes it easier to access meaning from an image than finding meaning from pixels.

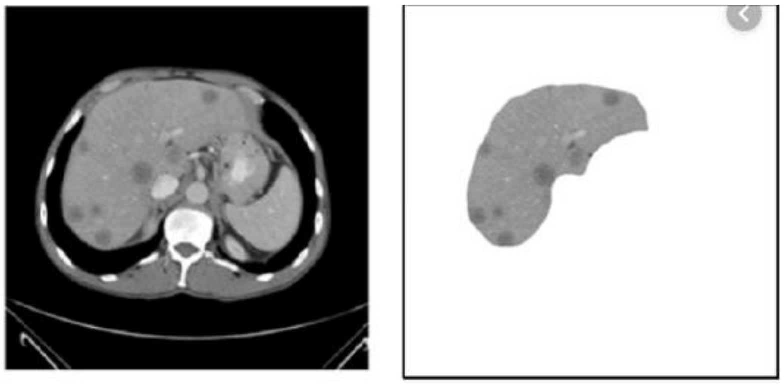
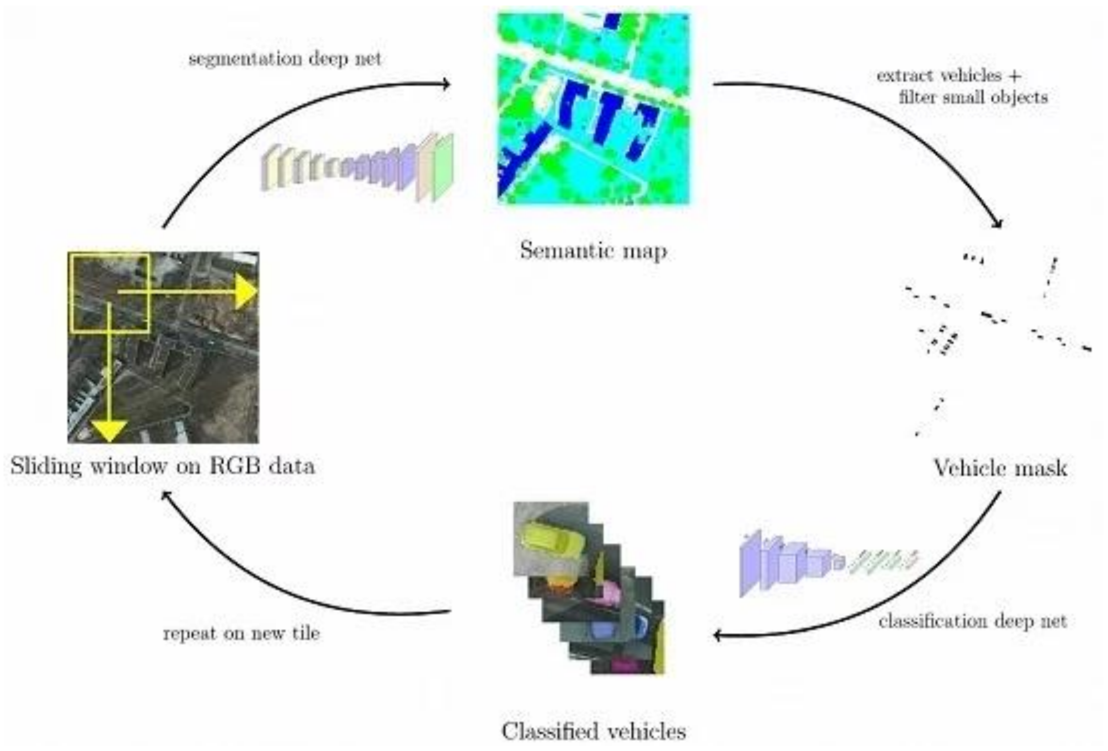


Figure 3.1. Model of liver segmentation

[Adapted from (Nader H. Abdel-massieh ; Mohiy M. Hadhoud ; Khalid M. Amin,” Fully automatic liver tumor segmentation from abdominal CT scans, The 2010 International Conference on Computer Engineering & Systems, IEEE, Cairo, Egypt,**doi:**10.1109/ICCES.2010.5674853)]

3.3 Working procedure of image segmentation

We can split the picture into different components called segments or partition it. Processing the whole image at the same time as there will be regions in the image that do not contain any information is not a great idea. We can use the significant sections to process the picture by splitting the picture into sections [11]. That’s how picture segmentation operates, in a nutshell. A picture is a multi-pixel collection or set. We group together the pixels that use picture segmentation to have comparable characteristics. Take a moment to go through the visual below (the picture segmentation will offer you a practical concept):



[Adapted from (Norman Kerle, Markus Gerke and Prasad S. Thenkabail," Received: 28 December 2016 / Revised: 5 April 2017 / Accepted: 7 April 2017 / Published: 13 April 2017 *Remote Sens.* 2017, *9*(4), 368; https://doi.org/10.3390/rs9040368")]

Figure 3.2. Vehicles segmentation

Detection of objects generates a bounding box corresponding to each class in the image. But it tells us nothing about the form of the object. We only get the set of bounding boxes for the coordinates. We want more information – this is too unclear for our purposes. Image segmentation for each object in the image creates a pixel-wise mask. This method allows us to understand the object(s) in the picture much more granularly. Why are we going to have to go deep? Is it not possible to solve all image processing functions with easy bounding box coordinates? To answer this relevant question, let's take a real-world instance. There are three general approaches to segmentation, called thresholding, methods based on edge, and techniques based on region.

3.4 Threshold Segmentation

The simplest method of image segmentation is thresholding [12]. Thresholding can be used to construct binary images from a grayscale image. Binary images are produced by segmentation from color images. Based on their size, the pixels are separated. Segmentation is the method of assigning two or more groups to each pixel in the source image [13]. If there are more than two groups, multiple binary images are the usual result. The binary map contains two possibly disjoint regions, one of them containing pixels with input data values smaller than a threshold and another relating to the input values that are at or above the threshold. The former and latter areas are generally labeled respectively with zero (0) and non-zero (1) labels. The segmentation relies on the threshold picture property and how the threshold is selected

3.4.1 Simple Thresholding

The most common threshold image property is the gray pixel level: g (x, y) =0 if the threshold is f (x, y) T2 and g (x, y) = 1 if T1= f (x, y) = T2.

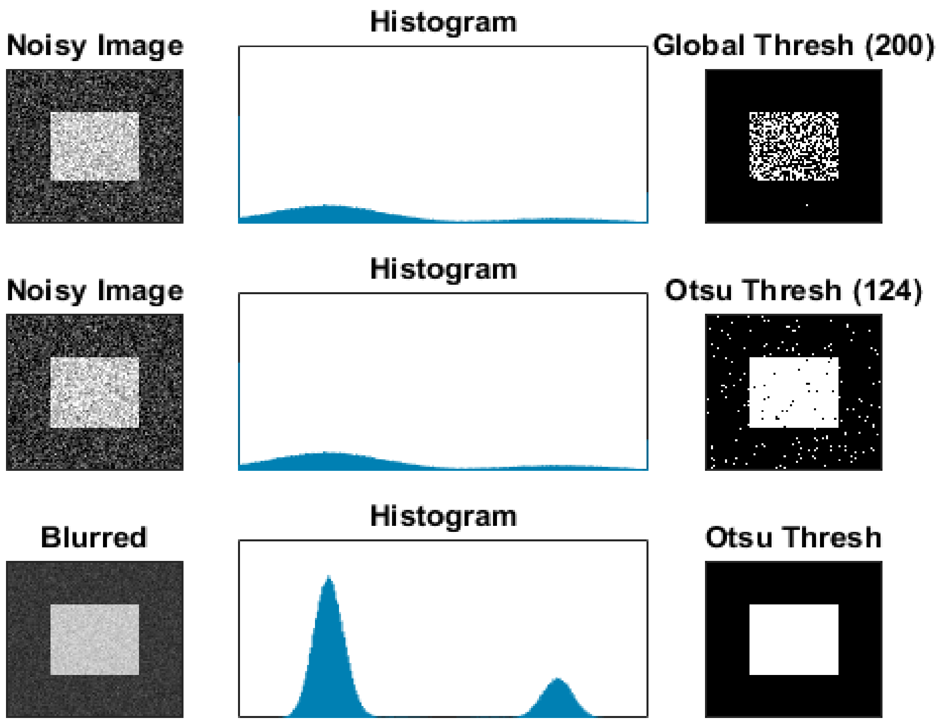


Figure 3.3. Image thresholding

[Adapted from (Digital Image Processing, Rafael C. Gonzalez)]

The main problems are whether it is possible and, if yes, how to choose an adequate threshold or a number of thresholds to separate one or more desired objects from their background [13]. In many practical cases the simple thresholding is unable to segment objects of interest, as shown in the above images. A general thresholding method is based on the premise that pictures are multimodal, i.e. different items of concern are associated with separate peaks (or modes) of the 1D signal histogram. Despite typical overlaps between the signal ranges corresponding to individual peaks, the thresholds must distinguish these peaks optimally. A threshold in the valley between two overlapping peaks distinguishes their primary bodies, but some pixels with intermediate signals are inevitably detected or mistakenly rejected. The ideal threshold that minimizes the anticipated number of false detections and rejections may not coincide between two overlapping peaks with the smallest point in the valley:

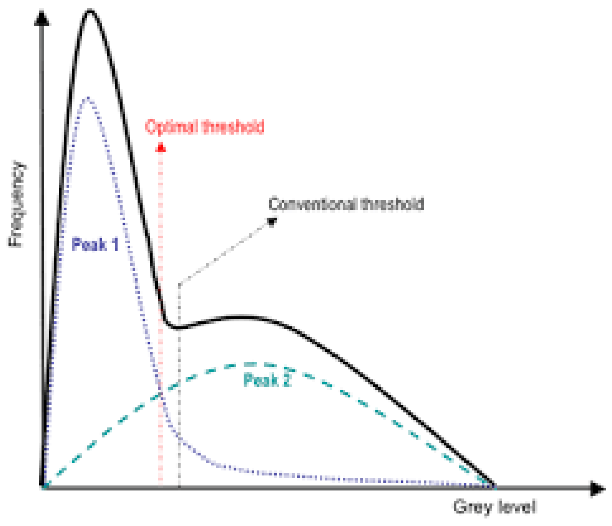


Figure 3.4. Graphical representation of thresholding

[Adapted from (V. Rajinikanth, N. Sri Madhava Raja, K. Latha, Optimal Multilevel Image Thresholding: An Analysis with PSO and BFO Algorithms. Aust. J. Basic & Appl. Sci., 8(9): 443-454, 2014)]

3.4.2 Adaptive thresholding

Because the threshold distinguishes the background from the object, the adaptive separation may take into account object (e.g. dark) and background (bright) pixel distributions of empirical probability [13]. Such a threshold must match two types of anticipated mistakes: assign a background pixel to the item and assign a background pixel to the item. More complicated adaptive thresholding methods use a spatially variable threshold to offset local spatial context effects (such a spatially variable threshold can be considered as background normalization). The classification of the object and background pixels is done at each iteration j by using the threshold Tj found at previous iteration. Thus, at iteration j, each grey level f (x, y) is assigned first to the object or background class (region) if f(x,y) Tj or f(x,y) > Tj, respectively. Then, the new threshold, Tj+1 = 0.5(j,ob + j,bg) where j,ob and j,bg denote the mean grey level at iteration j for the found object and background pixels, respectively.

3.4.3 Color thresholding

Color segmentation may be more precise due to more pixel-level data compared to gray-level pictures. The normal color depiction of Red-Green-Blue (RGB) has strong color components and a number of other color schemes (e.g. HSI Hue-Saturation-Intensity) have been designed in order to exclude redundancy, determine actual object / background colors irrespectively of illumination, and obtain more stable segmentation. An example below [13] shows that color thresholding can focus on an object of interest much better than its greyscale analogue.

Segmentation of color images involve a partitioning of the color space, i.e. RGB or HSI space. One simple approach is based on some reference (or dominant) color (R0, G0, B0) and thresholding of Cartesian distances to it from every pixel color f(x,y) = (R(x,y),G(x,y),B(x,y)).

$$g(x,y) = \left\{\begin{matrix} 1\ if\ d(x,y) \leq d_{max} \\ \\ 0\ if\ d(x,y) > d_{max} \end{matrix}\right. ; d(x,y) = \sqrt{(R(x,y) - R0)^2 + (G(x,y) - G_0)^2 + (B(x,y) - B_0)^2} \left.\vphantom{\begin{matrix} 1 \\ \\ 0 \end{matrix}}\right\} \tag{1}$$

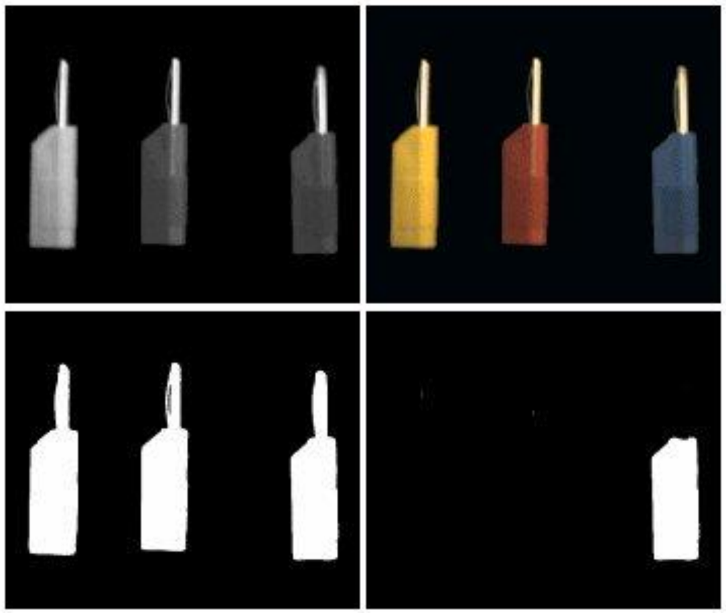


Figure 3.5. Color thresholding

[Adapted from (MATRIX VISION GmbH Talstrasse 16 DE - 71570 Oppenweiler)] where g(x,y) is the binary region map after thresholding. This thresholding rule defines a sphere in RGB space, centered on the reference color. All pixels inside or on the sphere belong to the region indexed with 1 and all other pixels are in the region 0. Also, there can be an ellipsoidal decision surface if independent distance thresholds are specified for the R, G, and B components. Generally, color segmentation, just as the greyscale one, may be based on the analysis of 3D color histograms or their more convenient 2D projections. A color histogram is built by partitioning of the color space onto a fixed number of bins such that the colors within each bin are considered as the same color. An example below of the partitioned 11×11×11 RGB color space is from.

3.5 Region-based Segmentation

Regional segmentation algorithms operate iteratively by grouping neighboring pixels with similar values and dividing pixel groups that differ in value. Growing seeded region is a semi-automatic merging type technique. We're going to clarify it as an instance. In Figure.3.5.- (a) displays a collection of radius 3 white seeds positioned inside all muscle fibers using a computer mouse-controlled on-screen cursor. In Figure.3.5.- (b) demonstrates again the edge filter output from Prewitt. The seeds and limits of a segmentation generated by a watershed algorithm type are superimposed on it in white. In Figure3.5.- (c) also shows the limits superimposed on the initial muscle fiber picture [13]. The watershed algorithm (we will explain the name later) works as follows:

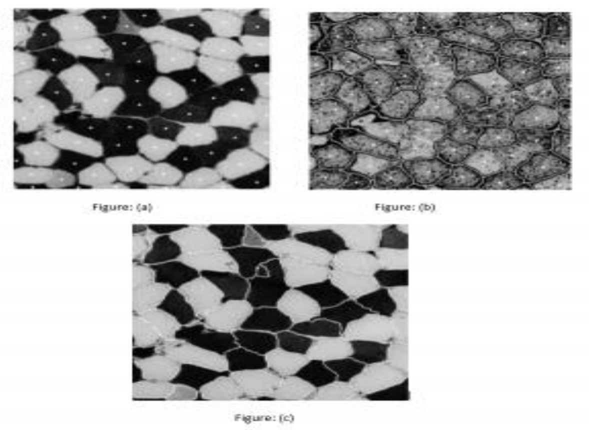


Figure 3.6. Manual segmentation of muscle fibers image by use of watersheds algorithm [13]

The normal use of the watershed algorithm is completely automatic. Again, by illustration, we will show this. Figure demonstrates the production of a Gaussian weight's variance filter (3.6) applied to the picture of muscle fibers after histogram-equalization.

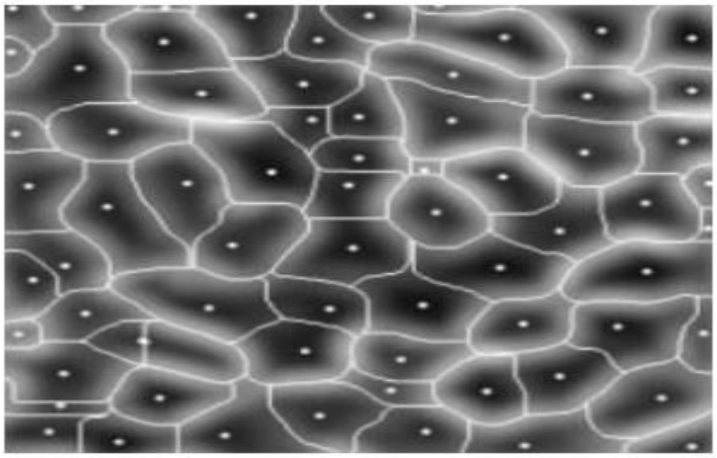


Figure 3. 7.Output of variance filter with Gaussian weights (σ2 = 96) produced by watershed algorithm [13]

There are many other algorithms based on the region, but most of them are quite complex. We will consider only one more in this chapter, namely an elegant split-and-merge algorithm suggested by Horowitz and Pavlidis (1976). To segment the log-transformed SAR picture, we will present it in a mildly altered form, basing our segmentation choices on variances, whereas Horowitz and Pavlidis are based on their pixel range. The algorithm works in two phases and needs a maximum variance limit to be specified in pixel values in a region. The first phase is the one that splits. Initially, the entire image's variance is calculated. The picture is split into four quadrants if this variance exceeds the designated limit. Likewise, if the variance exceeds the threshold in any of these four quadrants, it is further subdivided into four. This goes on until the entire picture consists of a collection of squares of variable dimensions, all of which vary below the limit. (Note that this must be achieved by the algorithm because the variances are taken to be zero at the finest resolution of each square consisting of a single pixel.) The indicates the resulting white borders overlapping the log-transformed SAR picture with the variance limit set at 0.60. Note that:

• In non-uniform areas of the picture, squares are narrower

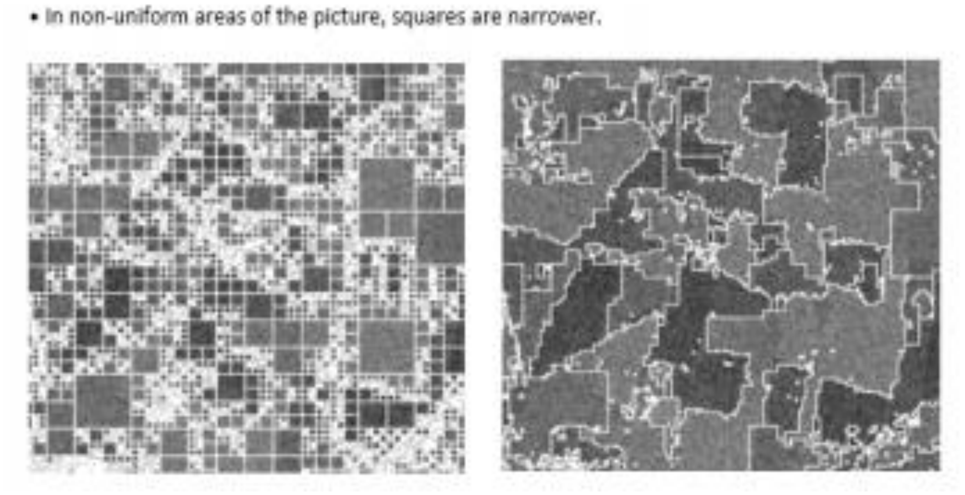

• In non-uniform areas of the picture, squares are narrower.

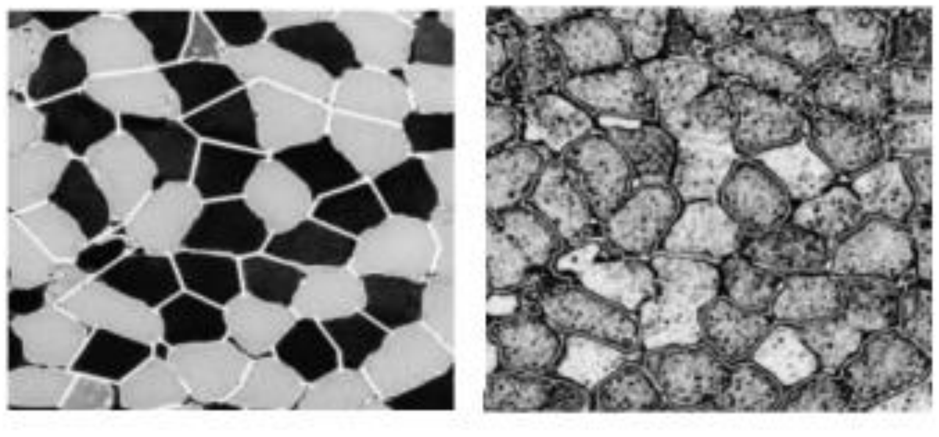
Figure 3.8. Region-growing segmentation of log-transformed SAR image [13]

• The strain limit was set at 0.60 instead of the speckle variance value at 0.41 (Hogan, 1994), as the resulting areas were very low in the latter case. The algorithm needs the size of the picture, n, to be a 2 force. The 250 x 250 SAR picture was therefore filled out to 256 x 256 by adding width 3 boundaries.

The second stage of the algorithm, the merging one, involves merging squares with a common edge, provided that this does not exceed the limit of the new region's variance. Upon completion of all amalgamations, the outcome is a segmentation in which each region has a variance below the specified threshold. However, although the outcome of the algorithm's first phase is unique, that from the second phase is not — it depends on the order of which squares are regarded. Fig: 2(b) shows the boundaries produced by the algorithm, superimposed on the SAR image. Dark and light fields appear to have been successfully distinguished between, although the boundaries are rough and retain some of the artefacts of the squares in Fig 2:(a). Pavlidis and Liow (1990) suggested by merging the outcomes with those from an edge-based segmentation to overcome the deficiencies in the limits generated by the Horowitz and Pavlidis algorithm. Many other concepts have been suggested for region-based segmentation (see, for instance, Haralick and Shapiro review, 1985), and it is still an active rese region.

3.6 Edge-based segmentation

Usually the findings of threshold-based segmentation are less than ideal, as we have seen[13]. A scientist will often have to change the automatic segmentation outcomes. One easy way to do this is to manage a screen cursor with a computer mouse and draw border lines between areas. In Figure.3.6.(a) shows the boundaries obtained by thresholding the image of muscle fibers (as shown in Figure :3(a), superimposed on the output from the edge filter of Prewitt (§ 3.4.2), with the contrast stretched so that values between 0 and 5 are displayed as shades of gray ranging from white to black and values above 5 are all displayed as black. This display can be used to help determine where to insert additional limits to fully segment all muscle fibers. Figure: 3(b) indicates the outcome after 71 straight lines have been added manually.



(a) (b)

Figure: 3.9.(a) Boundaries acquired by thresholding the picture of muscle fibers (b) Manual picture segmentation by adding additional lines to threshold limits [13]

Algorithms are accessible for semi-automatic drawing of edges, smoothing and disturbing the rough lines of the scientist to maximize some criterion of matching with the picture (see, for instance, Samadani and Han, 1993). Alternatively, it is possible to make edge finding fully automatic, though not necessarily successful. The outcome of applying Prewitt's edge filter to the muscle fiber picture is shown in Figure.3.9.-(a). The filter output was thresholder at a value of 5 in this screen: all pixels above 5 are labeled as edge pixels and shown as black. Connected edge pixel chains split the picture into areas.

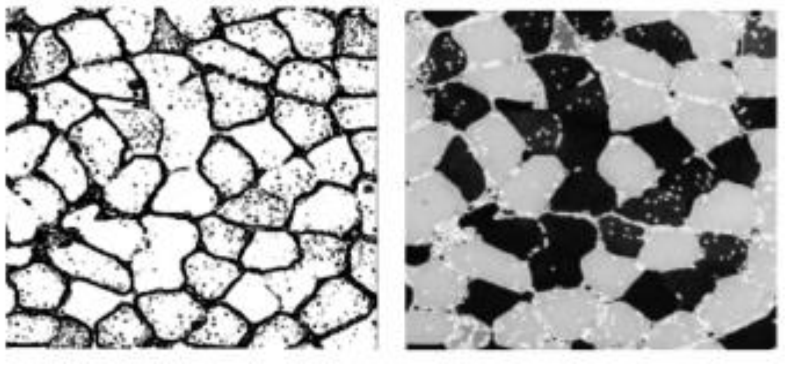


Figure 3.10. (a) Thresholder output of Prewitt’s edge filter (b) boundaries generated in (a) from linked areas [13]

Segmentation can be accomplished by assigning all non-edge pixels not separated by an edge to a single category. Rosenfeld and for 4- and 8-connected areas, Pfalz (1966) provided an effective algorithm, called a connected component algorithm. This algorithm will be described in words, then mathematically.[14]

3.7 Using Deep Learning

Deep learning is a form of artificial intelligence which simulates the functioning of the human brain in information processing and generating patterns for decision making use. Deep learning is a subset of artificial intelligence (AI) machine learning it has networks capable of understanding from unorganized or unidentified information without supervision. Deep learning might have developed hand in hand with the computer age, resulting in an increase of information in all kinds that from all areas of the universe. Along with other sources like social media, internet search engines, e-commerce platforms, and digital films, this information, known as big data. This massive amount of information is easily available [15].

How deep learning generates of this kind outstanding results: in a phrase, precision. Deep learning generates identification effectiveness at higher levels than before. This allows consumer electronics to satisfy user expectations and seems to be essential for safety-critical applications like driverless vehicles. Recent developments in deep learning have risen to the extent that certain functions, like cl, have been implemented. such as classifying objects in pictures, deep learning outperforms people. While deep learning was first theorized in the 1980s, there are two main reasons it has only recently become useful:

- Deep learning needs big quantities of information marked. For example, it takes millions of images and thousands of hours of video to develop driverless cars.
- Deep learning needs considerable electricity in computing. High-performance GPUs have an effective parallel architecture for profound learning.

This allows development teams to decrease training time for a deep learning network from weeks to hours or less when coupled with clusters or cloud computing. Examples of Deep Learning at Work: Automated Driving: Automobiles research teams use deep learning to automatically understand goods such as stop signs and traffic lights. In addition, deep learning is used to identify pedestrians, which helps reduce accidents. Aerospace and Defense: Deep learning is being used to classify areas of research for satellites and to detect safe or unsafe areas for soldiers. Medical Research: Deep learning is also used by cancer researchers to classify cancer cells instantly. UCLA teams have developed a unique microscope that gives a high-dimensional collection of data used it to train a deep learning application for specific cancer cell detection. Industrial Automation: Deep learning serves to ensure the safety of workers around heavy equipment by automatically recognizing people or products within an unsafe range of devices. Electronics: Home support tools that relate to our voice and recognize our preferences are driven. Deep learning is used in automatic hearing and speech translation.

3.7.1 Neural Networks

Deep learning becomes part of a bigger family of machine learning methods based on artificial neural networks. Artificial neural networks were inspired by the processing of information and the production of communication nodes in biological systems. Neural networks, a magnificent programming framework that encourages a computer to learn from experimental data Deep learning, a solid set of neural network learning techniques [16].

Neurons

Neuron: An element in a neural network comprising of a prematurely quantity of neurons. A neural network has various layers. A neuron is a function (x1, x2, xn), a femoral function that uses f as input and delivers a binary output and a weight factor multiplied by the operate of sigmoid and calculates how often this neuron is considered for the output of the layer.

Neurons are the basic unit of a neural network. Basically, the neurons involve multiple axons (inputs), a cell nucleus (processor) and an axon (output). When the neuron activates, all of its incoming inputs are gathered, and a message passes through axon when it exceeds a certain threshold. sort of. The key thing for neurons is that they can learn. Artificial neurons look more like this:

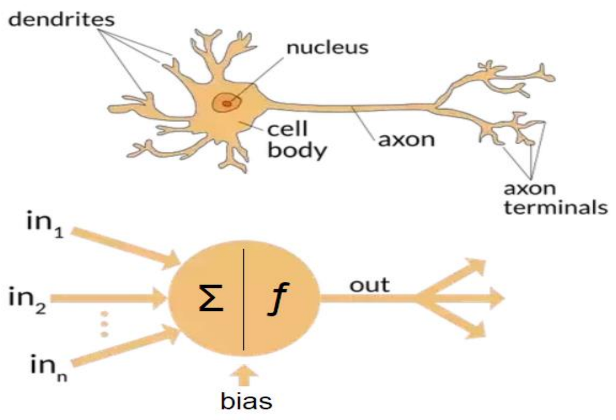


Figure 3.11. Neuron Neural Network

[Adapted from (Nagyfi Richárd, “The differences between Artificial and Biological Neural Networks”, Towards Data Science, Sep 4, 2018)]

There is a weight for each input as you can see that they have multiple outputs. When the artificial neuron activates, it computes its condition by adding all incoming inputs multiplied by their linked attachment weight. Neurons, however, always have one extra input, the distortion, which is always 1 and is associated with its own weight. This ensures that there would be activation in the neuron even though all inputs are none (all 0s). Where Guam is all inputs (such as bias) After measuring its situation, the neuron passes it through its activation function, which normalizes the outcome (usually between 0-1).

Activations

Functions for activation are statistical equations that evaluate the output of a neural network. The function is attached to each neuron of the network and determines whether it should be activated ("fired") or not depending on whether each neuron's input is relevant to the model's prediction. Activation functions often attempt to normalize each neuron's output to a range of 1 to 0 or -1 to 1. Another element of the activation features is that they are computationally demanding because thousands or even millions of neurons are calculated for each information sample the model, which places the activation function and its derivative function with an enhanced computational strain:

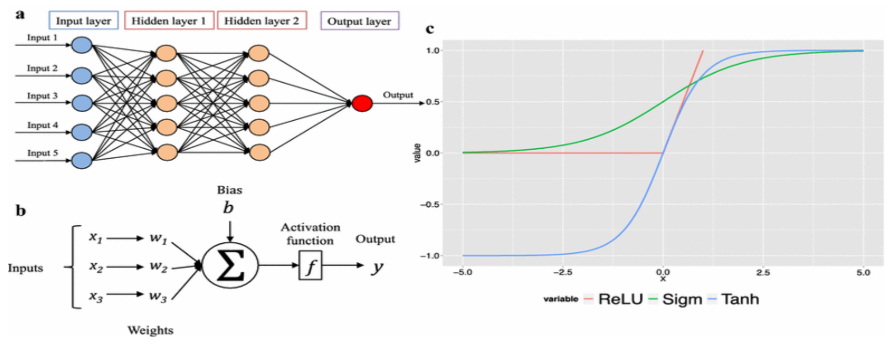


Figure 3.12. Activation

[Medium,@krishnakalyan3/introduction-to-exponential-linear-unit-d3e2904b366c]

Elements of a Neural Network

Layer of input: -This layer recognizes characteristics of input. All of this offers the network with data from the outside world, no computation is done on this layer, networks here just pass the information(features) on to the hidden layer. Hidden Layer: - Nodes of this layer are still not presented in the outside world, they are a function of any neural network's abstraction. Hidden layer calculates all kinds of characteristics entered through the input layer and transfers the outcome to the output layer. Output Layer: - This layer introduces for the outside world the data the network has gained.

What is an activation function and why to use them?

Definition of activation function: -

Activation function chooses whether or not to activate a neuron by calculating weighted sum and adding bias with it further. The activation function's goal is to bring nonlinearity into a neuron's output.

Explanation: - We understand that the neural network has neurons that operate in weight, prejudice and their corresponding activation function accordingly. In a neural network, based on the output mistake, we would update the weights and biases of the neurons. This method is called back-propagation. Activation functions make the back-propagation possible since the gradients are supplied along with the error to update the weights and biases.

3.7.2 Convolutional Neural Networks

Convolution

They are basically just neural networks using Convolutional layers, a.k.a. Conv layers, depending on convolution's mathematical operation. Conv layers are a collection of filters that you can only think of as 2d numbers matrices. Here's an example 3x3 filter:

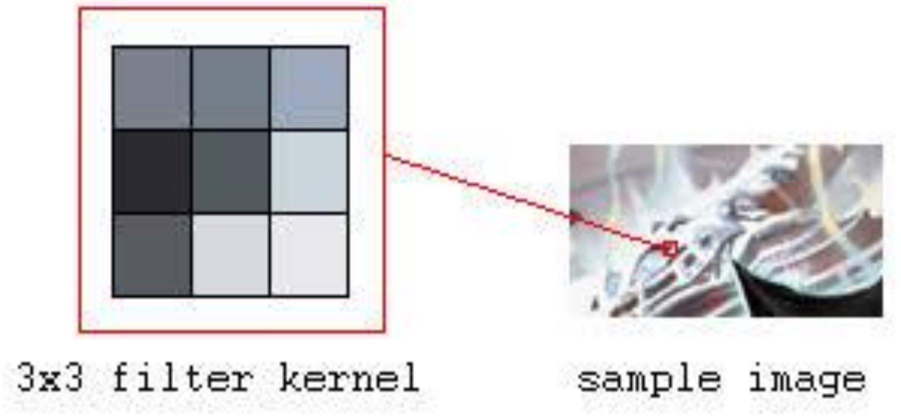
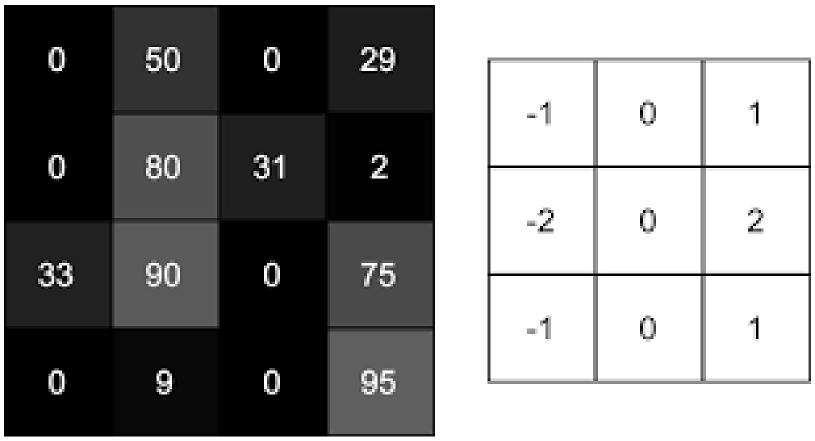


Figure 3.13. 3x3 filter

Fig: 3x3 filter use for an input image and a filter to produce an output image by convolving the filter with the input image.

- Overlay the filter at some place on top of the picture.
- Multiplication of the element-wise values in the filter and the respective values in the picture.
- Summing up all the products that are element-wise. This amount is the output value in the output picture for the target pixel.
- Repeat for all sites.

That description of 4 steps was a little abstract, so let’s take an example. Consider this small image of 4x4 grayscale and this filter of 3x3:



0	50	0	29
0	80	31	2
33	90	0	75
0	9	0	95

-1	0	1
-2	0	2
-1	0	1

Figure 3.14. 4x4 image (left) and 3x3 filter (right).

The picture figures depict the levels of intensity of pixels, where 0 is black and 255 is white. To generate a 2x2 output image, we will convert the input image and the filter:

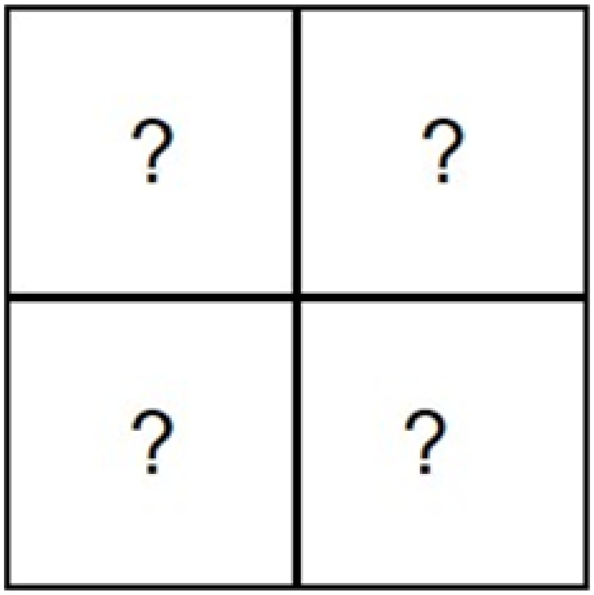


Figure 3.15. 2x2 filter

output image

To start, lets overlay our filter in the top left corner of the image:

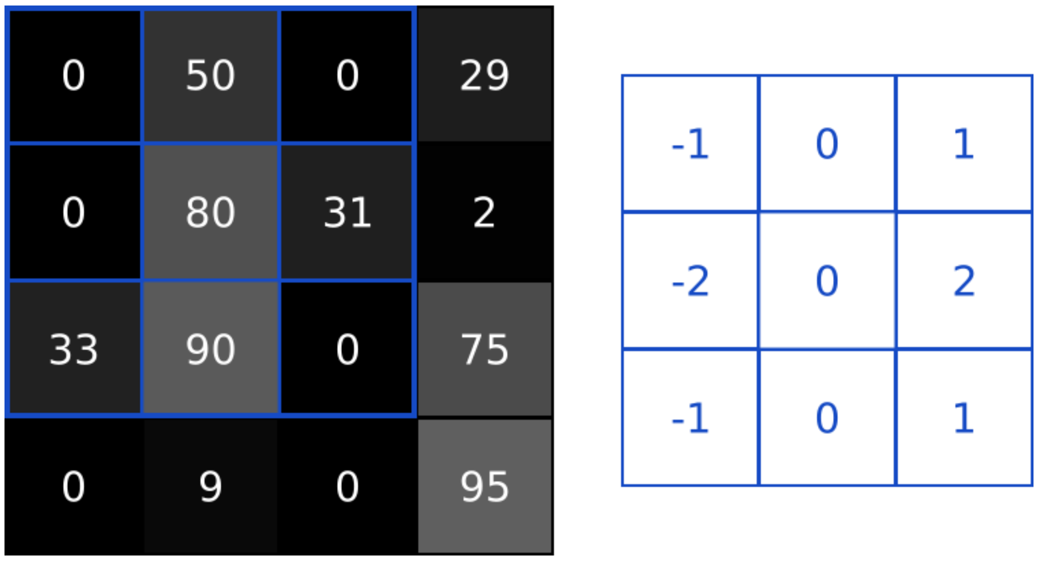


Figure 3.16. Step 1: Overlay the filter (right) on top of the image (left)

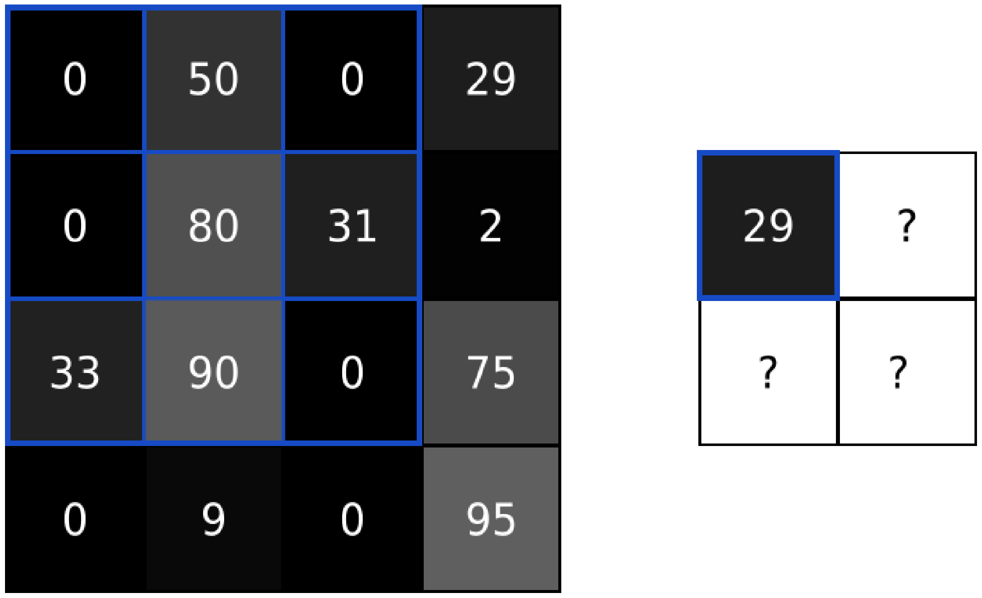
Next, we multiply element-wise between the overlapping values of the image and the values of the filter. Here are the results, from the top left corner to the right, then down:

Image Value	Filter Value	Result
0	-1	0
50	0	0
0	1	0
0	-2	0
80	0	0
31	2	62
33	-1	-33
90	0	0
0	1	0

Figure 3.17. Step 2: Performing element-wise multiplication.

Next, we sum up all the results. That's easy enough: 62–33=29.

Finally, we position our outcome in our output image's destination pixel. As our filter is overlaid in the top left corner of the input picture, the top left pixel of the output picture is our target pixel:



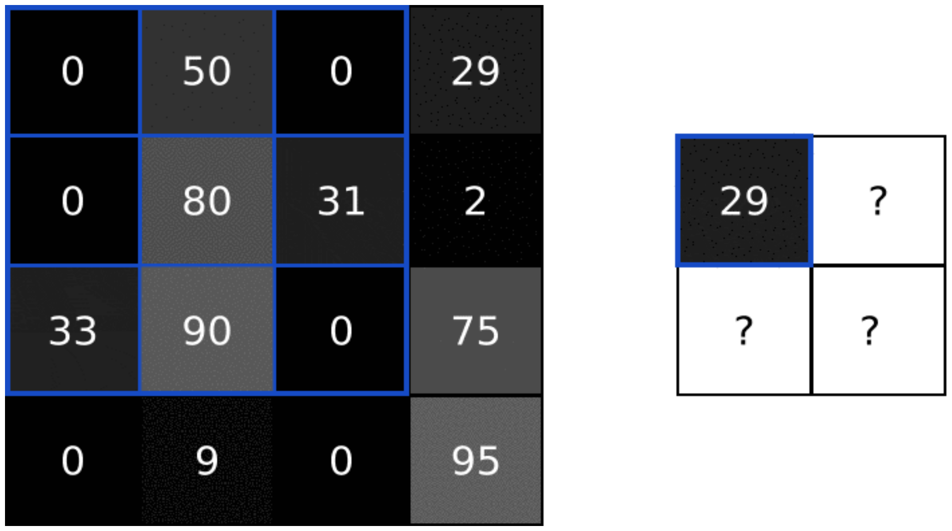


Figure 3.18. Do the same thing to generate the rest of the output image

Padding: Remember to convert a 4x4 input picture to generate a 2x2 output picture with a 3x3 filter previously? We will often prefer the output picture to be the same size as the input picture. To do this, we are adding zeros around the picture so that in more locations we can overlay the filter. 3x3 filter needs 1-pixel padding:

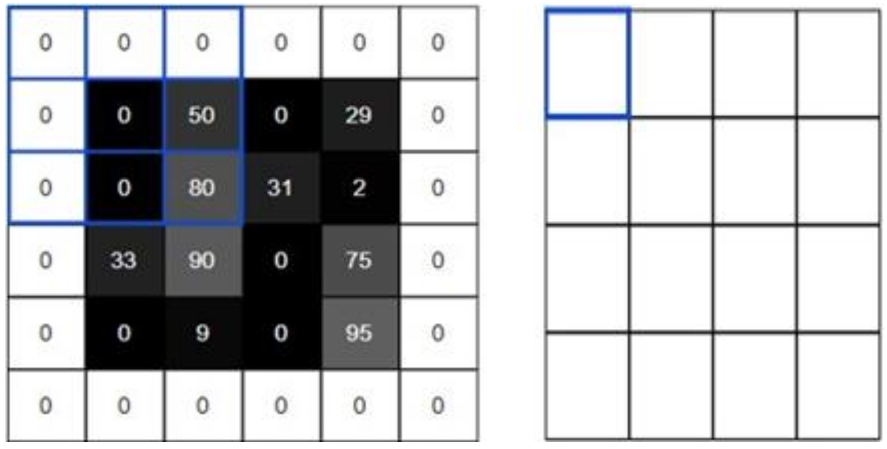


Figure 3.19. 4x4 input convolved with 3x3 filter to produce a 4x4 output using same padding

This is called "same" padding because the sizes of input and output are the same. It is sometimes referred to as "valid" padding not to use any padding, which is what we have done and will continue to do for this article.

Conv Layers: Since know how much the transformation of images works and why it is useful, let’s see how it is effectively used in CNNs. As mentioned previously, CNNs include converting layers using a set of filters to convert input images into output images. A conv layer’s primary parameter is the quantity of filters it has. For here MNIST CNN, will use as the previous layer in our network a relatively small conv layer with 8 filters. This implies that the 28x28 input picture will become a volume of output 26x26x8.

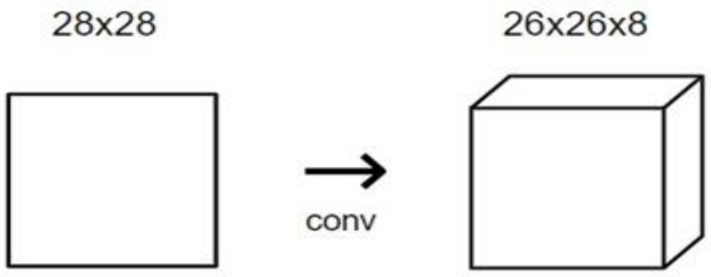


Figure 3.20. Conv Layers.

Each of the 8 filters in the conv layer produces a 26x26 output, so stacked together they make up a 26x26x8 volume. All of this happens because of 3.

3.7.3 Pooling

General pooling

The pooling technologies may also behave other tasks as well as max pooling, such as average pooling or even L2-norm pooling. Historically, average pooling is often used, but it has recently fallen by the wayside relative to the max pooling method, which has been shown to work better.

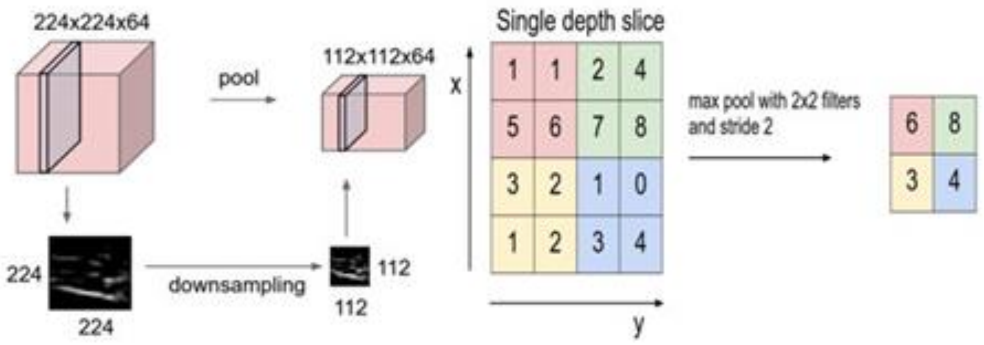


Figure 3.21. Pooling
[Adapted from (Leonardo Araujo Santos , Gitbooks , Artificial Inteligence , content , pooling)]

Pooling layer down samples the number in space, differently in each feedback volume shape slice. Left: In this example, the length input volume [224x224x64] is pooled into the size output volume [112x112x64] with filter size 2, step 2. Note that the depth of the quantity is maintained. Right: The most widespread down sampling procedure is max, resulting in max pooling, with a step of 2 shown here. That is, 4 numbers (little 2x2 square) are drawn over each max.

3.7.4 U-NET Architecture

U-NET: U-Net is regarded as one of the regular CNN architectures for image identification tasks where it is necessary mostly to define this whole image by class, as well as to segment the image area by class, i.e. to create a mask that separates the image into several classes. The architecture comprised of a hiring route for context capturing and a symmetrical extension approach for exact localization.

The network is instructed in end-to-end operation from very few images and reaches the past perfect method on the ISBI challenge of segmenting neuronal structures in electron microscopic stacks (a coevolutionary sliding window network).That use the same network hired on transmitted light microscopy pictures (stage contrast and DIC), U-Net gained a big margin in these categories in just the 2015 ISBI cell tracking challenge. The network is also quick. Segmentation of an image of 512 range512 on a contemporary GPU requires less than a second.

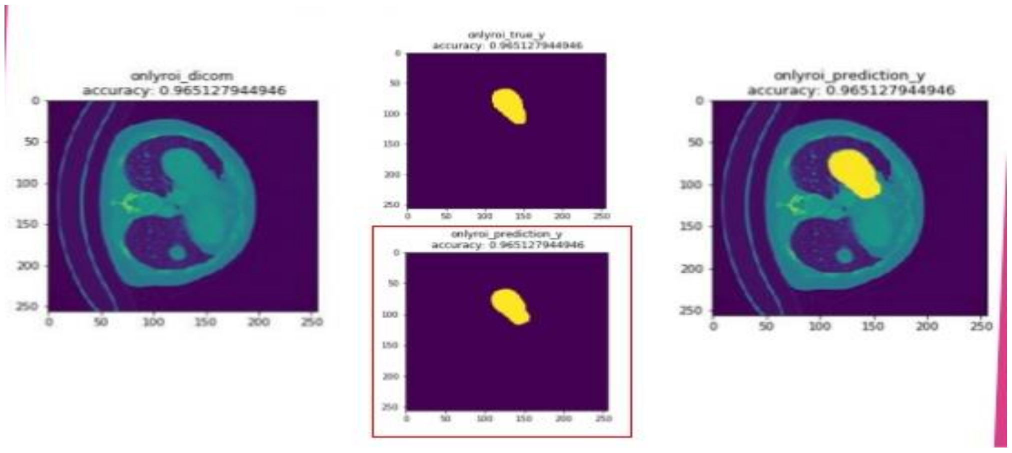


Figure 3.22. Liver segmentation using U-net.

[Adapted from (WonjoongCheon, Liver segmentation using U-net: Practical issues @ SNU-TF, Published on Jul 25, 2018, and Published in: Health & Medicine)]

The U-net Architecture

Developed on the Fully Convolutional Network, the U-Net architecture is modified to improve segmentation in medical imaging. Compared to FCN-8, the two key differences are:

- U-net is symmetric
- The links between the down sampling route and the up-sampling route apply a concatenation operator rather than a sum.

The aim of these skip connections is to provide local information when sampling global information. Because of its simplicity, the network has a large number of feature maps in the up-sampling path, which allows information to be transmitted. By comparison, the basic FCN architecture contained only a number of class feature maps in its sampling path.[16]. The U-Net owes its name to its symmetric shape, which is different from other FCN variants. U-Net architecture is separated into 3 parts:

- The contracting/down sampling path

- Bottleneck
- The expanding/up sampling path

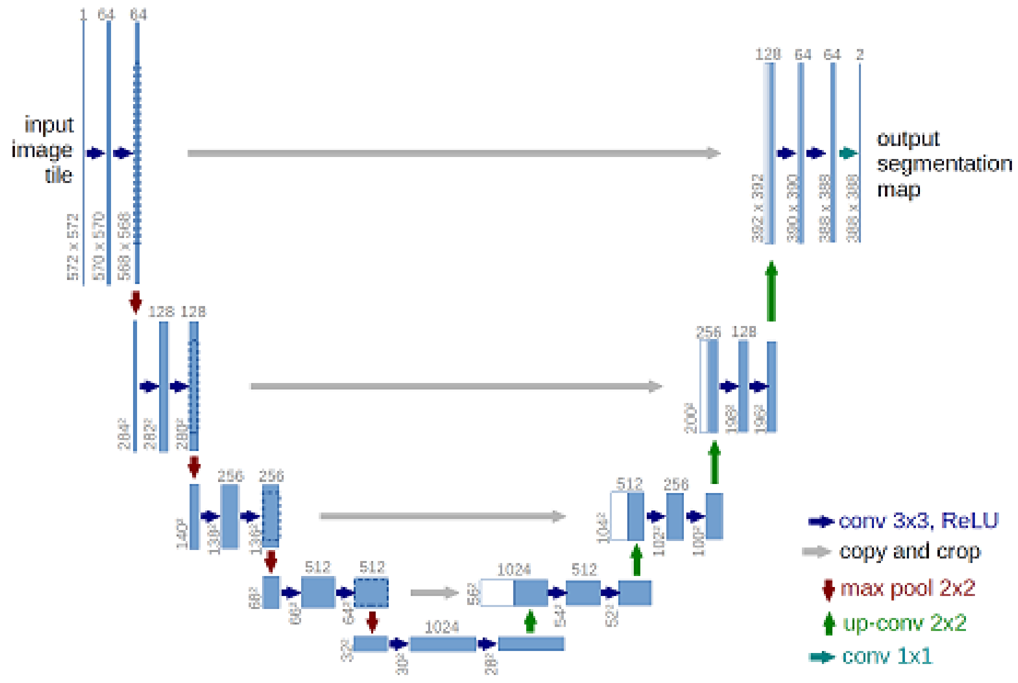


Figure 3.23. U-net architecture (display at smallest resolution for 32×32 pixels) [4]

In Figure.3.23.- Each blue box connects to a function map of different channels. The set of channels on top of the box is stated. The x-y-size is displayed at the box's upper left edge. White boxes are duplicate maps of features. The arrows represent the various actions. Here the figure explains the network architecture. It contains an agreement route (left side) and an extension path (right side). The standard network architecture follows the contracting path. It includes repeated application of two 33 covenants (unpadded covenants), each accompanied by a linear rectified unit (ReLU) and a 2×2 max pooling procedure with step 2 down sampling. New app channels are matched to each step of down sampling. Each stage in the evolving route comprises of a sampling of the function border followed by an up-convolution of 2×2 that halves the number of function segments, a concatenation with the correspondingly cropped function map from the contracting route, and two convolutions of 3×3, merged by a ReLU. The shading is necessary in each and every convolution due to the current loss of boundary pixels.

Advantages

- High consistency of adequate training, appropriate data sets and time for training.
- No heavy layer, then we can use images of various sizes as input (as the only dimensions to learn on convolution layers are the kernel, as well as the kernel size is independent of the size of the input image).
- The use of huge information gains is significant in fields such as biomedical segmentation, as the percentage of samples annotated is generally restricted

Chapter 4

Methodology

4.1 Overview of our proposed segmentation:

Our proposed work on division is appeared in Figure 4.1. The work process is comprised of three significant advances. In the initial step we manage information handling and groundwork for the neural system division. In the second step, two completely convolutional neural systems initially portioned the liver and afterward sores inside the locale of enthusiasm for the liver (ROI). The determined probabilities of CFCN will be refined in the last third step utilizing a thick 3D restrictive irregular field to produce the consequence of conclusive division.

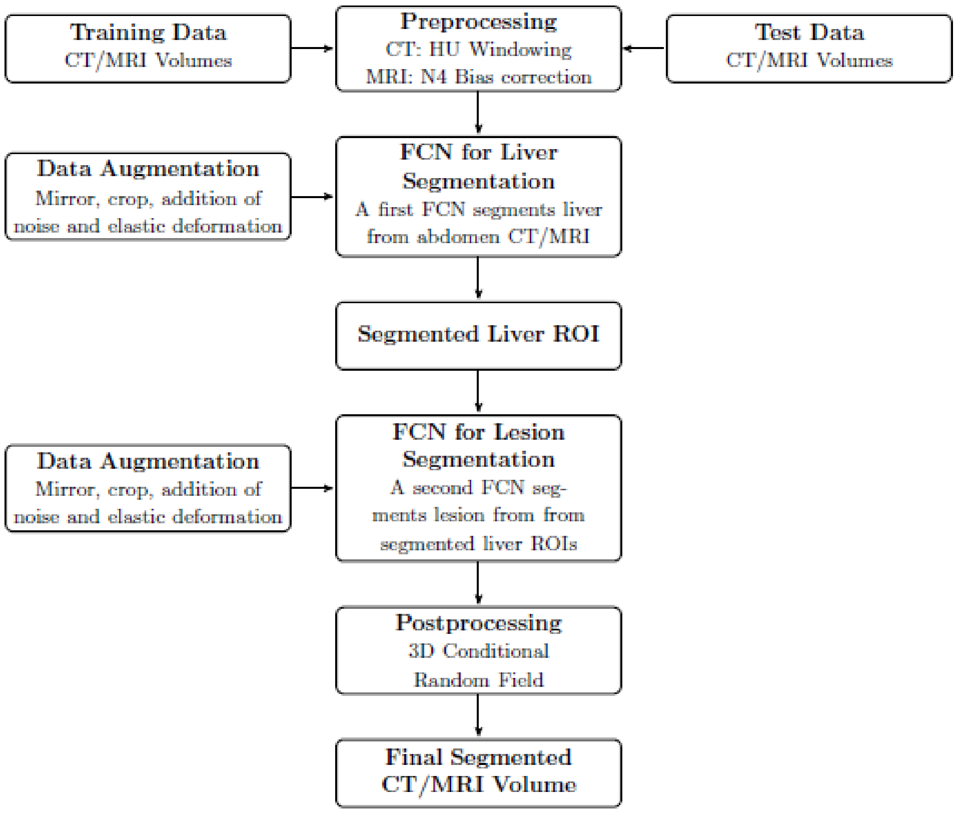

Training Data
CT/MRI Volumes
Preprocessing
CT: HU Windowing
MRI: N4 Bias correction
Test Data
CT/MRI Volumes
Data Augmentation
Mirror, crop, addition of noise and elastic deformation
FCN for Liver Segmentation
A first FCN segments liver from abdomen CT/MRI
Segmented Liver ROI
Data Augmentation
Mirror, crop, addition of noise and elastic deformation
FCN for Lesion Segmentation
A second FCN segments lesion from from segmented liver ROIs
Postprocessing
3D Conditional Random Field
Final Segmented CT/MRI Volume

Figure 4.1. Diagram of the proposed picture division work process for preparing and testing [Adapted from (O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: MICCAI, Vol. 9351, 2015, pp. 234–241)]

4.2 Data Preparation

This area manages the pre-handling and expansion for CT data. In a cut astute manner, pre-preparing was performed. In the first place, so as to bar unessential organs and properties, the Hounsfield unit esteems are windowed in the range [- 100; 400]. Figure 4.2. shows the impact of our applied preprocessing to a crude restorative cut. We improved the contrast by equalizing the histogram is also shown in Figure.4.2. After HU-windowing and contrast enhancement, Figure 4.2. also shows the final slice. For improved detection of irregular liver tissue, the contrast within the liver has been enhanced. The data preparation scheme is similar for DW-MRI and differs in the normalization of results, which also performs correction of N4bias.The data preparation scheme is similar for DW-MRI and differs in the normalization of results, which also performs correction of N4bias [18].In order to teach the network the desired invariance properties, several steps of data increases such as elastic deformation, translation, rotation and addition of Gaussian noise with standard deviation of the current slice were used to increase the CFCN training data.[17,19].

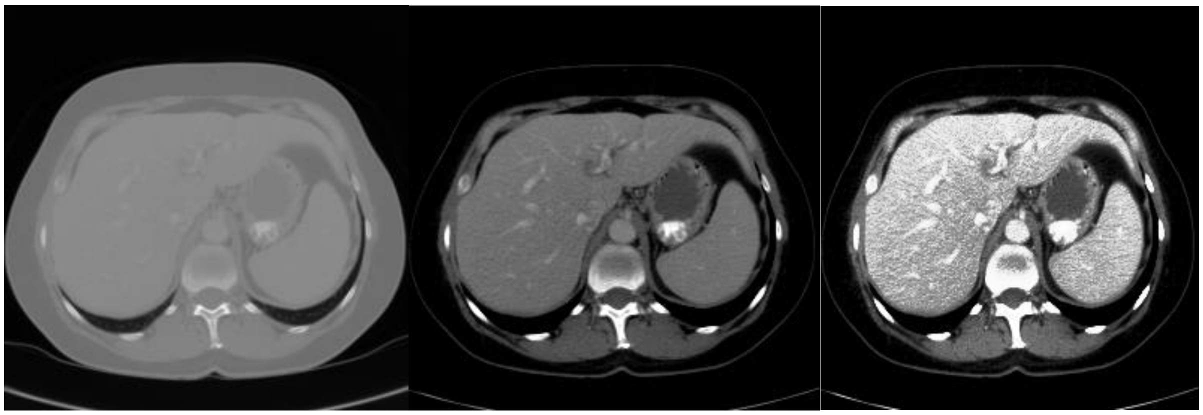


Figure 4.2. Overview of the applied preprocessing steps [18]

4.2.1 Augmentation

Data augmentation technique significantly increases the diversity of data available for training models, without actually collecting new data. Data augmentation is a way of creating new 'data' from prior information with distinct orientations. It is done by applying domain specific techniques to examples from the training data that create new and different training examples. Image augmentation includes more variety to the training dataset and in the event that it is done right, reflects the variety within the genuine data and thus makes a difference in the model to generalize better [26].

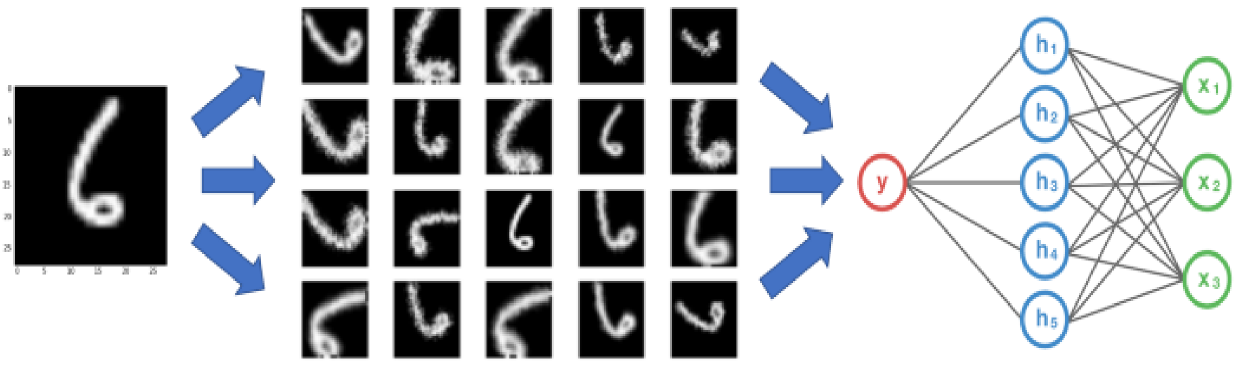


Figure 4.3. Data Augmentation [26]

4.2.2 Augmentation Techniques used

Shift

It is not possible to center objects in the images within the outline. In a variety of ways, they may be off-center. A shift to an image means that all pixels of the image are moved in one course. It can be horizontal or vertical and it holds the proportions of the picture the same. This means that some of the pixels are clipped off the image and it is necessary to specify new pixel values in an image region. There are two forms of shift; vertical and horizontal. The horizontal change shifts the picture along the horizontal axis (left or right) in a certain direction. Through vertical change, the object is shifted (up or down) along the vertical axis. The arguments for the Image Data Generator width_shift_range and height_shift_range regulate the amount of horizontal and vertical shift respectively. In particular, for each image and the shift performed, e.g.[0, value], a value in the range between no shift and the percentage or pixel value will be sampled. Alternatively, we can specify a min and max variety tuple or array from which to sample the change; e.g.: [-100, 100] or [-0.5, 0.5].

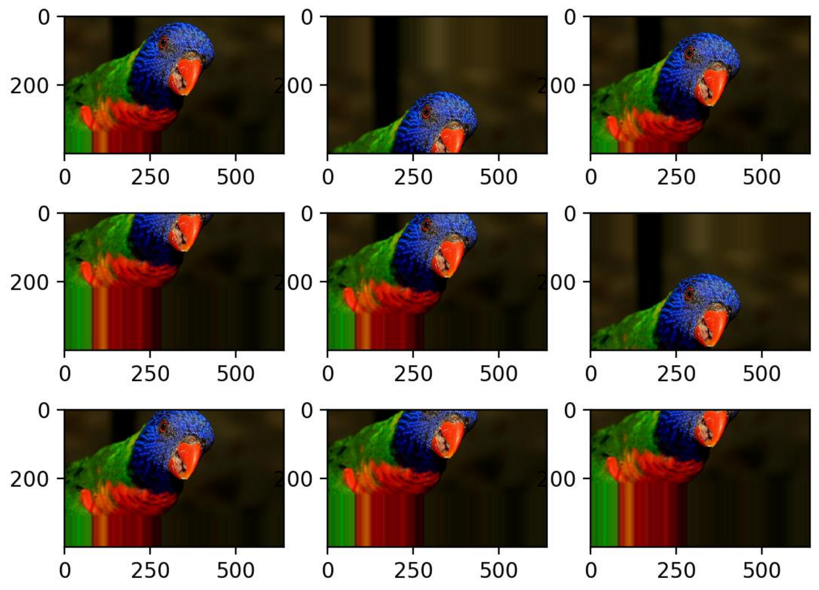


Figure 4.4. Augmented Images with a Random Vertical Shift [26]

Flip

You can horizontally and vertically flip images. Some frameworks do not provide vertical flips feature. Nonetheless, a vertical flip is equivalent to a 180-degree rotation of an object and then a horizontal flip. Below are examples of flipping images. In Figure 4.2.3. From the left, we have the original image, followed by the image flipped horizontally, and then the image flipped vertically [26].

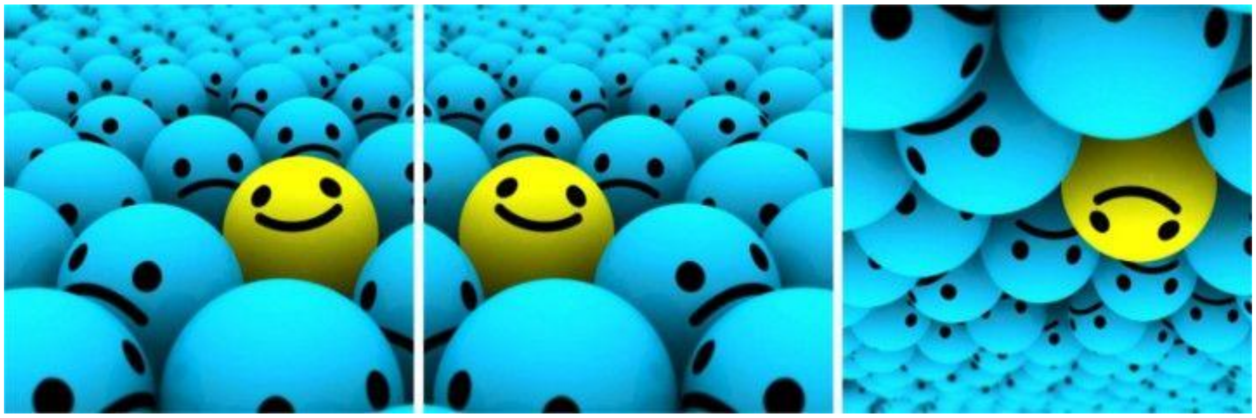


Figure 4.5. Augmented flipped image [26]

Rotation

A rotation augmentation rotates the picture randomly from 0 to 360 degrees in the clockwise direction. Image dimensions after rotation might not be maintained. The rotation will rotate pixels out of the picture outline and take-off zones of the frame with no pixel data that must be filled in. If the image is square, the image size will be preserved by rotating it at right angles. If it is a rectangle, it would maintain the size by rotating it by 180 degrees. The final image size will also be changed by rotating the image with finer angles [26]. In Figure 4.6.the images are rotated by 90 degrees clockwise with respect to the previous one, as we move from left to right.

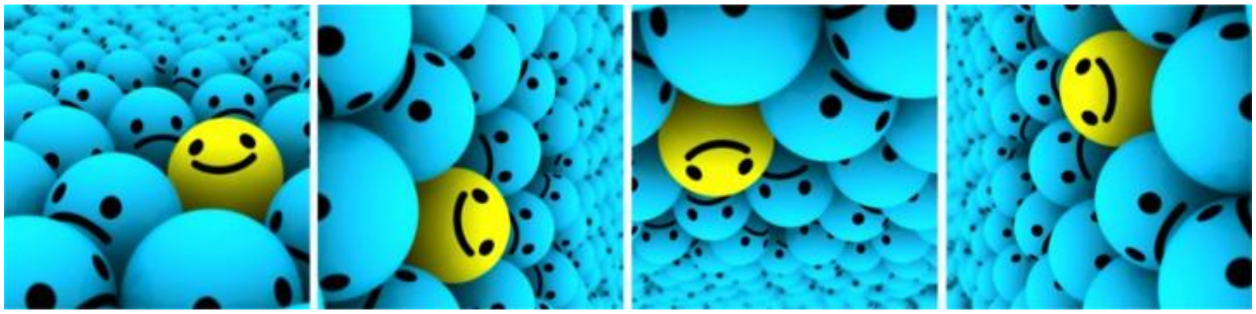


Figure 4.6. Augmented rotated image [26]

4.3 Cascaded Fully Convolutional Neural Networks

This paper presents a method for automatically segmenting liver and lesions in CT and MRI images using cascaded, fully convolutional neural networks (CFCNs) for the segmentation of a large-scale medical or quantitative image analysis[4].We describe various state-of - the-art deep learning architecture and design choices we evaluated for use in our segmentation tasks in the following section. We signify the volume of 3D image as I , the total number of voxels as N,and the number of potential labels as L = f0; 1;::; lg.We define a variable **xi** ∈ **L** for each voxel I which denotes the assigned label. The probability of a voxel i belonging to label k given the image I is described by P (xi = kjI) and will be modelled by the FCN. In our particular study, we use L = {0, 1, 2} for background, liver and lesion, respectively.[17]. Our results show that CFCN-based semantic liver and lesion segmentation achieves Dice scores over 94% for liver with computation times below 100s per volume.

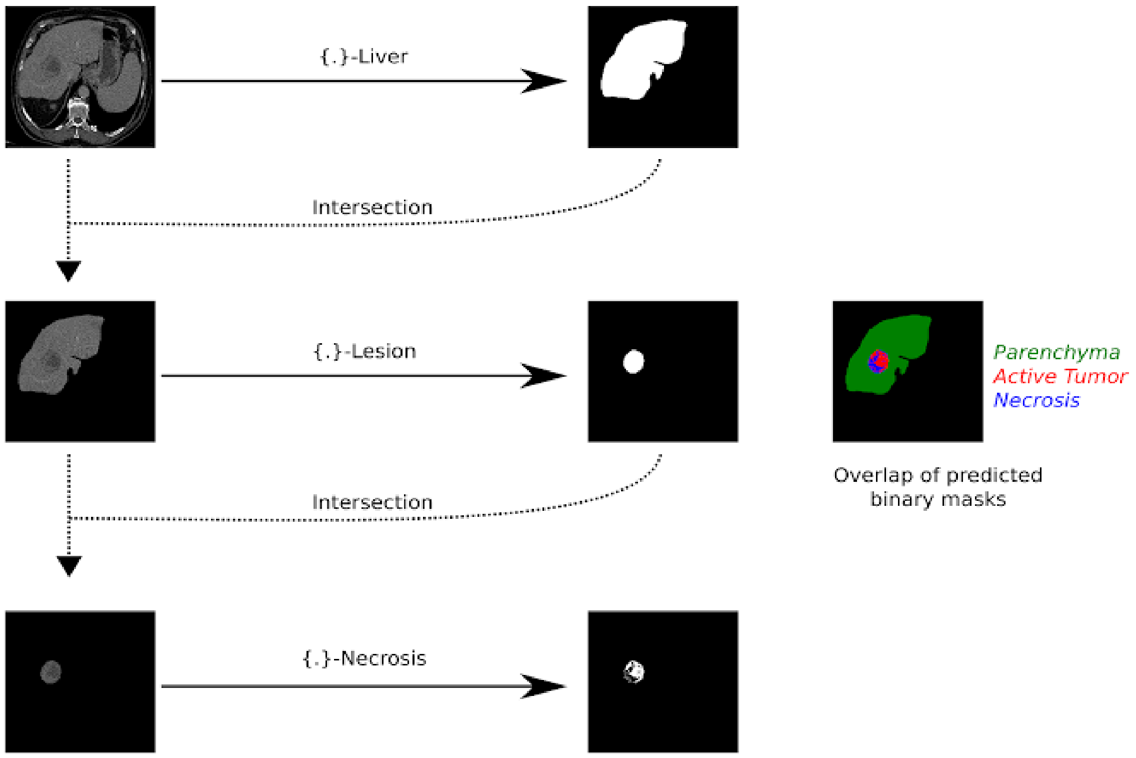


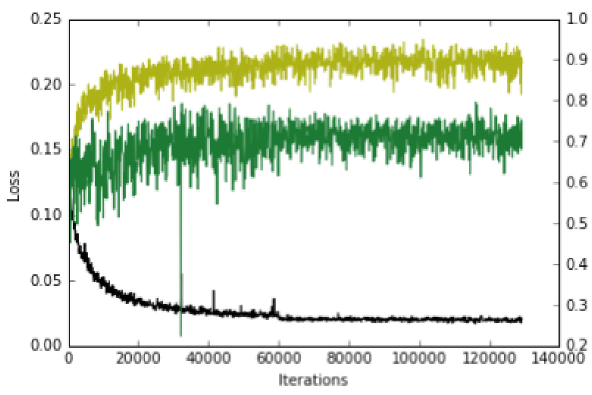
Figure 4.7. Automatic liver segmentation using Cascade-fully-Neural-network.

[Adapted from (J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, CVPR)]

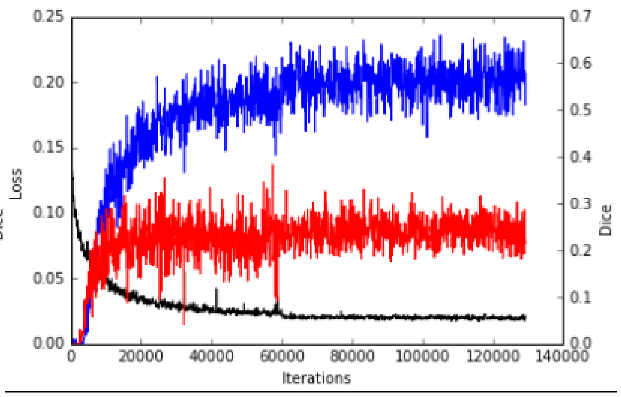
4.3.1 From Alex Net to U-Net

The first completely convolutional network architecture for semantic segmentation was proposed by Long et al. (2015) [20]. The main idea in their research is to replace the last fully connected layers of a classification network like the Alex Net[21] with fully convolutional layers so that dense pixel-wise predictions are possible.

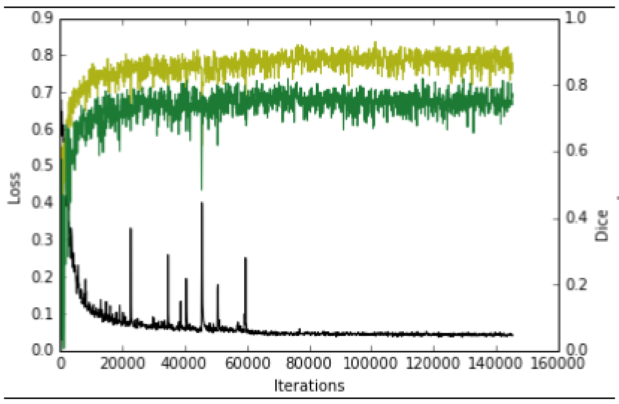
To suit the dimensions of the source, the last completely convolutional layers need to be upscaled. Compared to previous work, the AlexFCN allows full-sized clinical slices to be projected pixel wise rather than patch-wise. In Figures 4.4(a) and (b) display the AlexFCN (without class balancing) training curves on the 3DIRCAD dataset. In training and testing Dice overlap, all training curves converged rapidly to a steady state. All learning curves show a large overfitting of the AlexFCN without class balance, with Dice overlaps of 71%/90% in liver test / training data and 24%/60% in lesions. Overall, the 24 percent check time lesion dice is comparably small. Big, and so on. Long et al. (2015) explicitly stated that their question of natural object segmentation did not need to be extended to class balancing. One explanation is that they used Alex Net pretrained weights trained on natural objects, i.e. Data from ImageNet. However, for many medical applications it is mandatory to apply class balancing since pre-trained networks from natural images cannot be used properly and the class of interest occurs more seldom in the dataset. Figure 4.4 (c) and (d) show the importance of matching classes in the segmentation of clinical objects. Training and screening Dice for both liver and lesions significantly increase to 78% for liver and 38% for lesions. Another significant change can be accomplished by implementing Ranneberger et al. (2015)'s revised U-Net Architecture [17].In addition to the increased depth of 19 layers and learnable upscaling (up-convolution), the U-Net offers a superior pattern of skipping connections between various neural network stages. Spatial knowledge is present in the activations of the current stage in the early stages of the neural network. In later stages of the neural network, at the expense of specific knowledge on the location of these constructs, spatial information is converted to semantine information. For example, in the U-Net bottleneck, the original U-Net architecture reduces an input image of 6 size 388x388 to 28x28. Skip-connections were introduced by Ranneberger et al. to allow spatial and semantic data to be used at a later stage as spatial information from an earlier stage can be merged later in the neural network. Therefore, later stages of the neural network may use semanthropic and spatial information to infer data.



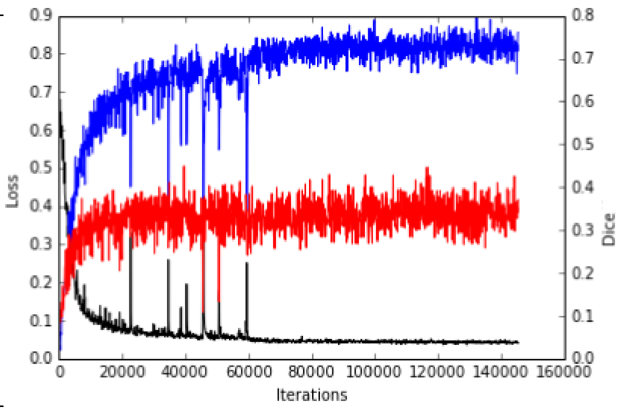
(a) AlexFCN architecture without class balancing.



(b) AlexFCN architecture without class balancing of Lesion



(c) AlexFCN architecture with class balancing.



(b) AlexFCN architecture with class balancing of Lesion

Figure 4.8. Training curves of different network architectures and training procedures of liver and lesion on 3DIRCAD dataset.

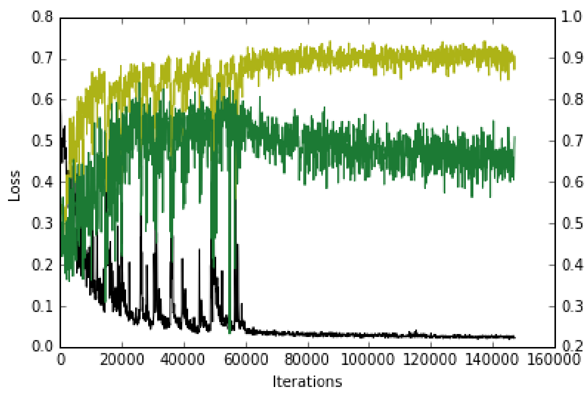
[Adapted from (A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, M. Nielsen, Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network, in: MICCAI, Vol. 16, 2013, pp. 246–253.)]

4.3.2 From FCN to CFCN

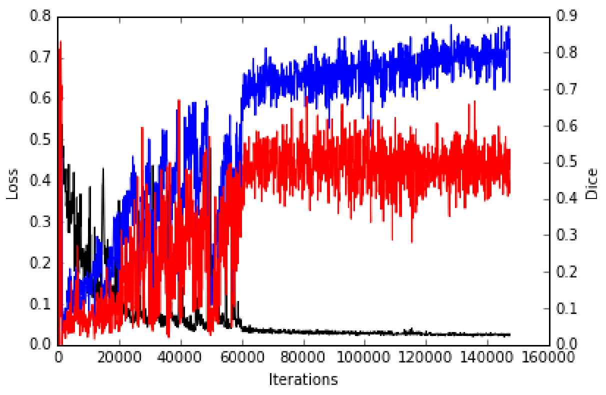
U-Net architecture [17] was used to measure the probability maps of the soft tag P(xi |I). The U-Net architecture allows for accurate pixel-wise prediction by integrating spatial and contextual data in a 19-layer network architecture. Figures 4e and 4f display on 3DIRCAD data set the training curves for the U-Net. The lesion segmentation's overall performance is further increased to 53 percent Dice test. The U-Net has acquired features that simultaneously discriminate against liver and lesion. As one of our key contributions, we deliver a cascaded FCN training to learn specific features to solve a segmentation task once per week, resulting in higher quality in segmentation.

The motivation behind the cascade approach is that U-Nets and other forms of CNNs have been shown to learn a hierarchical representation of the data provided. Instead of designing hand-crafted apps for separating various tissue types, the stacked layers of coevolutionary filters are customized to the desired category in a data-driven manner. By cascading two U-Nets, we ensure that the U-Net in step 1 learns filters that are specific for the detection and segmentation of the liver from an overall abdominal CT scan, while the U-Net in step 2 arranges a set of filters for separation of lesions from the liver tissue. In addition, liver ROI helps to reduce false lesion positives. Our suggested approach is demonstrated in Figures 5 and 6. They train a network to divide the liver into slices of the abdomen (step 1). This network will focus solely on learning biased features for segmentation of the liver versus the context, e.g. Figure 5. After that we train another network, given a liver image (step 2), to segment the lesions. The segmented liver from step 1 is harvested and re-sampled for the cascaded U-Net in step 2 to the required input size.

All non-liver regions are masked and the second U-Net can focus on learning discriminative characteristics for segmentation of lesion vs. liver background.



(a) U-Net architecture with class balancing of Liver



(b) U-Net architecture with class balancing of Lesion

Figure 4.9. Training curves of different network architectures and training procedures of liver and lesion on 3DIRCAD dataset.

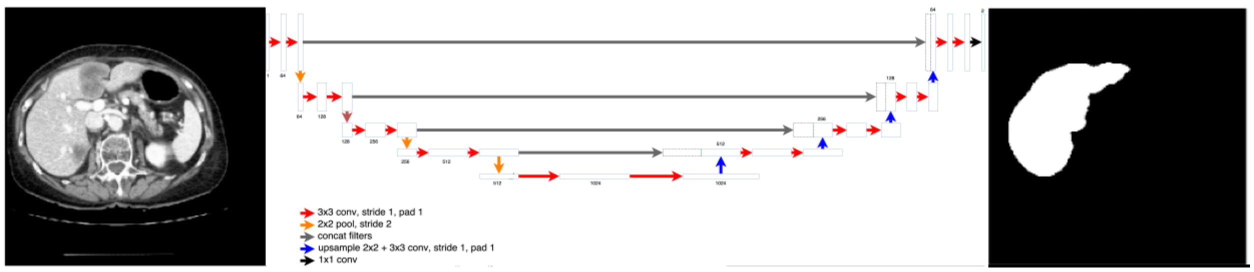

3x3 conv, stride 1, pad 1
2x2 pool, stride 2
concat filters
upsample 2x2 + 3x3 conv, stride 1, pad 1
1x1 conv

Figure 4.10. Step 1 of Cascaded FCN: The rst U-Net learns to segment livers from a CT slice.

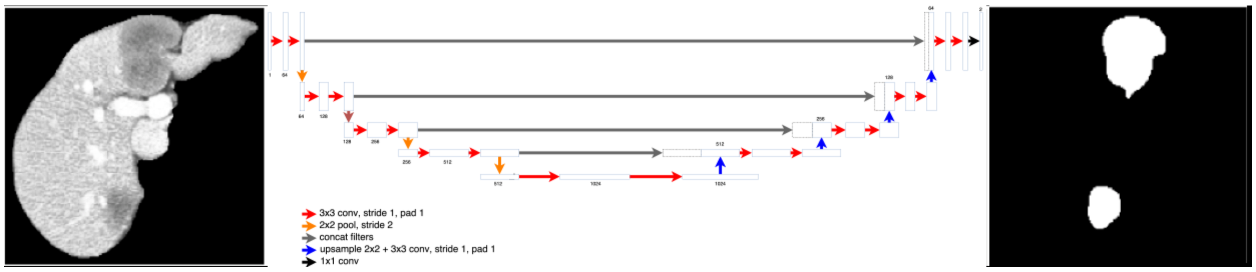

3x3 conv, stride 1, pad 1
2x2 pool, stride 2
concat filters
upsample 2x2 + 3x3 conv, stride 1, pad 1
1x1 conv

Figure 4.11. Step 2 of Cascaded FCN: The second U-Net learns to segment lesions from a liver segmentation mask segmented in step 1 of the cascade

[Adapted from (P. Kr¨ahenb¨uhl, V. Koltun, Efficient inference in fully connected crfs with gaussian edge potentials, in: NIPS, 2011, pp. 109–117)]

4.3.3 Transfer Learning and Pretraining

Transfer learning using pre-trained neural network models is a common concept in deep learning. Neural networks pre-trained on another task, such as a collection of data classification of natural objects, can be used as an initialization of network weight when training on a new task, such as segmentation of medical volume images. The theory behind this idea is that the first layers of neural networks often learn similar concepts to identify basic structures like blobs and edges for other tasks or datasets. When using pre-trained models, these principles have not been trained from scratch again. For our experiments, we have used pre-trained U-Net models given by Ranneberger et al. (2015), which were trained in cell image data segmentation[17].The trained models of liver and lesion segmentation have been released to allow other researchers to begin their training with learned concepts of liver and lesion.

4.4 Quality Measure

Our proposed system performance is evaluated by using consistency indicators in the major challenges for the segmentation of liver and lesion.

Our main metric is the Dice coefficient. We report Volume Overlap Error (VOE), Relative Volume Difference (RVD). Metrics are applied to binary valued volumes, so a metric computed on the lesions for example considers only lesion objects as foreground and everything else as background. We refer to the foreground object in the ground truth as object A, and object B for the predicted object.

4.4.1 Dice Coefficient

The Dice coefficient (DC) is a simple and useful summary measure of spatial overlap that can be used for the productiveness and accuracy studies in the segmentation of the picture. Two medical tests generally found acceptable but complex validation outcomes. This test can be modified for similar testing purposes [25].

The Dice score or F1 measure is evaluates as:

Dice (A, B) =$\frac{2\,|A \cap B|}{|A|+|B|}$-----------------(2)

where the Dice score is in the interval [0,1]. A perfect segmentation yields a Dice score of 1.

4.4.2 Volume Overlap Error (VOE)

VOE is just the complement of the Jaccard coefficient:

$VOE(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$---(3)

4.4.3 Relative Volume Difference (RVD)

RVD is an asymmetric metric. It is defined as follows:

RVD (A, B) =$\frac{|B|-|A|}{|A|}$---(4)

Chapter 5

Results & Discussion

5.1 UNET Architecture

U-net architecture which is a fully convolutional network, we have train that architecture. The principle to train this architecture is to a usual contracting network, layers with upsampling operators instead of pooling. This provides access to the network learn context (contracting path), then localization (expansive path). Due to skip-connexions context information is propagated to higher resolution layers. So, we have input images in the data.py of same size we have & perform axial cuts of our 3D images. So, the image size is 256x256 which we use as input to the network.

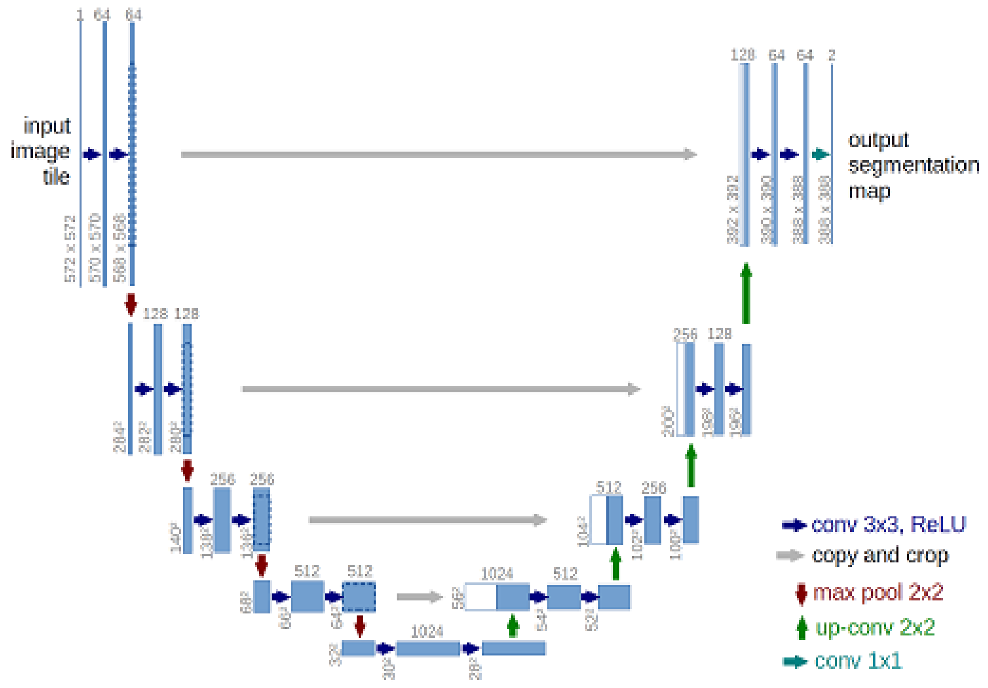


Figure 5.1 Implemented U-net architecture [5]

5.2 Training Setup

We train 3DICARD data set. For training we have got 1087 image slices. Every image go through 2-fold cross validation and take 2sec per step. The train test split ratio is 0.2. Due to the problem of GPU we had to train the data on Google Collaboratory, device was Nvidia Tesla k80. The learning rate is 0.001 and a momentum of 0.8. Metrics that we have used is dice coefficient. Total number of epochs is 20. And finally, it took about 10 hours to train the model.

5.3 Training Curves

We have taken a function to determine whether the model is working swiftly or not. Dice coefficient indicates the similarity between train and test data. Result of training dice co efficient is 87.53. We got the best result for epochs 19.

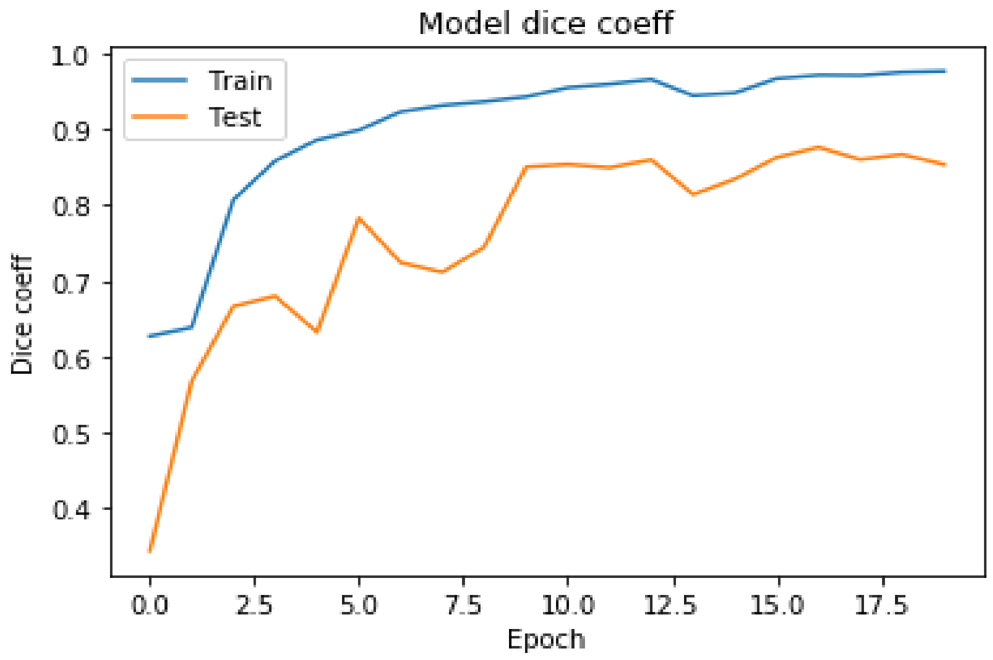


Figure 5.2 Training Curve of Dice Coefficient

From the graph of dice coefficient, we can see that the dice coefficient gets higher by time.

5.4 Results

After training the model we got the results that shows that the predictions of our implemented model give pretty much accurate result of hepatic tumors in the liver.

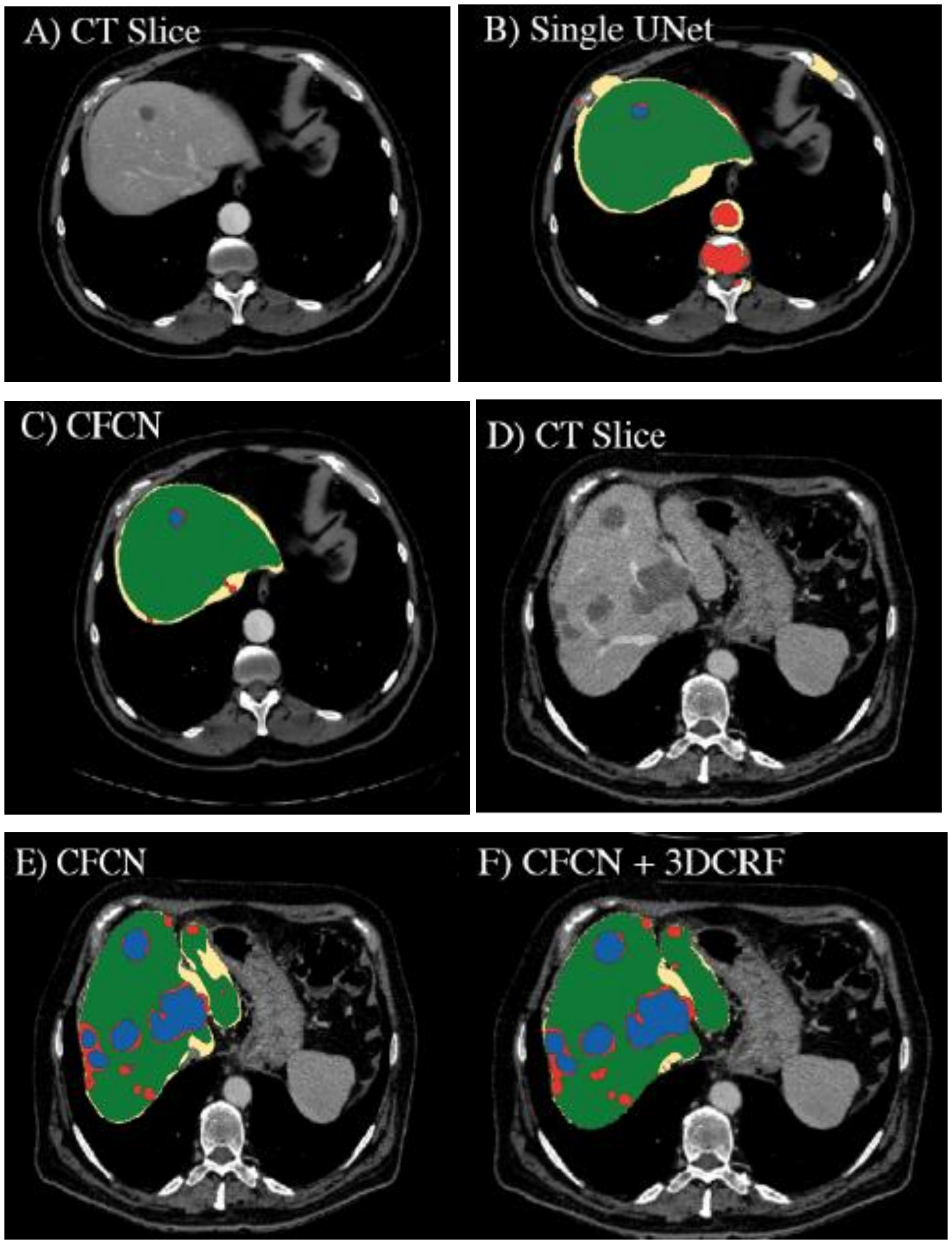


Figure 5.3 Some Example of results Using CFCN and CRF

We can see the green depicts which correctly predicts liver segmentation, yellow mark is for false negative and positive pixels (all wrong predictions), blue mark which shows correctly predicted lesion segmentation and for false negative and positive pixels in lesion is red mark (all wrong prediction). We can see in the first row, the false positive lesion prediction in picture B of a single U-Net, which is eliminated by using CFCN in picture C, is a result of restricting lesion segmentation to the liver ROI region. In the second row, in F increases both liver and lesion segmentation accuracy by applying the 3D CRF to CFCN.

Chapter 6

Conclusion and Future Scope

Conclusion

We studied several image processing techniques for liver segmentation data and implemented a segmentation model. This model can give a very good result even for small dataset and low-resolution images. The more we train the model the more efficient result we can get. Since the dataset consisted of a limited number of images, we can improve it by using more images. The model is fast and lightweight compared to others. So, it is easy to implement the model. The model is also scalable, so it can be implemented for bigger resolution images.

Future Scope

In our project, we work with a given model. We only use the existing data. But in future we want to modify our model by increasing the number of datasets. Now we only find out the dice score. In future we want to implement other metrics in our project. We have another plan with our project. As we implement U-net architecture here, in future we want to implement V-net architecture in our project. With V-net architecture we won’t need to convert the image from 3D to 2D. As V-net is quite expensive it is not used universally.

Bibliography

[1] Walker HK, Hall WD, Hurst JW,"Clinical Methods: The History, Physical, and Laboratory Examinations. 3rd edition,Chapter 94 Evaluation of the Size, Shape, and Consistency of the Liver".editors.Boston: Butterworths; 1990.

[2] Mohammad Hesam Hesamian,Wenjing Jia,Xiangjian He,Paul Kennedy,"Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges"August 2019, Volume 32,Issue 4, pp 582–596.

[3] Gotra A, Sivakumaran L, Chartrand G, Vu KN, Vandenbroucke-Menu F, Kauffmann C, Kadoury S, Gallix B, de Guise JA, Tang A.Liver segmentation: indications, techniques and future directions.Insights Imaging. 2017 Aug;8(4):377-392. doi: 10.1007/s13244-017-0558-1. Epub 2017 Jun 14. Review.PMID: 28616760.

[4] Patrick Christ and Mohamed Ezzeldin A. ElShaer are qualified to endorse. [Automatic Liver and Tumor Segmentation of CT and MRI Volumes using Cascaded Fully Convolutional Neural Networks], Submitted on 20 Feb 2017 (v1), last revised 23 Feb 2017 (this version, v2).

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997

[6] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," IEEE transactions on medical imaging, vol. 35, no. 5, pp. 1285–1298, 2016.

[7] G. A. Baxes, Digital image processing: principles and applications. Wiley New York, 1994.

[8] S. Hijazi, R. Kumar, and C. Rowen, "Using convolutional neural networks for image recognition," Cadence Design Systems Inc.: San Jose, CA, USA, 2015.

[9] Stéfan van der Walt, Johannes L. Schönberger,"scikit-image: image processing in Python",PeerJ Sections BIOINFORMATICS AND GENOMICS

[10] Linda Shapiro,and George C. Stockman (2001): "Computer Vision", pp 279-325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3

[11] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, “scikit-image: image processing in python,” PeerJ, vol. 2, p. e453, 2014.

[12] Zhang, Y. (2011). "Optimal multi-level Thresholding based on Maximum Tsallis Entropy via an Artificial Bee Colony Approach". Entropy. 13 (4): 841–859. Bibcode:2011Entrp..13..841Z. doi:10.3390/e13040841

[13] B. Demirta¸s et al., “Atatürk döneminde egitim alanında ya¸sanan geli¸smeler,” ˘ Gazi Akademik Bakı¸s, no. 02, pp. 155–176, 2008.

[14] C. K. Ingold et al., Structure and mechanism in organic chemistry. Cornell University Press Ithaca, NY, 1969, vol. 1.

[15] Y. Ma, D. Xiang, S. Zheng, D. Tian, and X. Liu, “Moving deep learning into web browser: How far can we go?” in The World Wide Web Conference. ACM, 2019, pp. 1234–1244.

[16] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in International Conference on Medical image computing
and computer-assisted intervention. Springer, 2015, pp. 234–241.

[17] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation,in: MICCAI, Vol. 9351, 2015, pp. 234{241.

[18] N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, J. C. Gee, N4ITK: Improved N3 bias correction, IEEE Transactions on Medical Imaging 29 (6) (2010) 1310{1320. doi:10.1109/TMI.2010.2046908

[19] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, B. Glocker, Ecient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation, Medical Image Analysis 36 (2017) 61{78.

[20] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, CVPR.

[21] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012, pp. 1097{1105.

[22] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, M. Nielsen, Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network, in: MICCAI, Vol. 16, 2013, pp. 246{253.

[23] F. Milletari, N. Navab, S.-A. Ahmadi, V-net: Fully convolutional neural networks for volumetric medical image segmentation, in: 3D Vision (3DV), 2016 Fourth International Conference on, IEEE, 2016, pp.565{571.

[24] O. C icek, A. Abdulkadir, S. S. Lienkamp, T. Brox, O. Ronneberger, 3d u-net: learning dense volumetric segmentation from sparse annotation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2016, pp. 424{432.

[25] Kelly H. Zou, Simon K. Warfield, Aditya Bharatha, Clare M.C. Tempany, Michael R. Kaus, Steven J. Haker, William M. Wells, III, Ferenc A. Jolesz, Ron KikinisKelly H. Zou, Simon K. Warfield, Aditya Bharatha, Clare M.C. Tempany, Michael R. Kaus, Steven J. Haker, William M. Wells, III, Ferenc A. Jolesz, Ron Kikinis,Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index: Scientific Reports, 2004 Feb; 11(2): 178–189. doi: 10.1016/S1076-6332(03)00671-8.

[26] F. Chollet, “Building powerful image classification models using very little data,” Keras Blog, 2016.