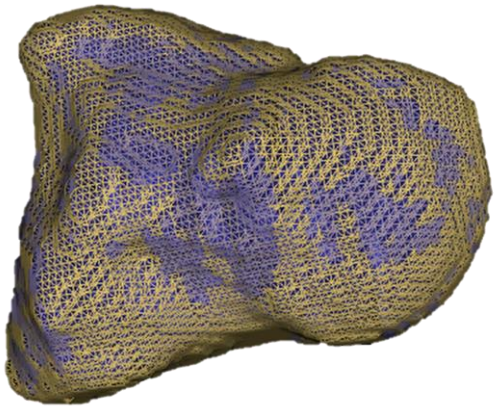
3/31/2021

UNet-based Fully-automatic Segmentation of the Capitate from CT Images



Wenli Xue (Master student BME program, 1284584)
DELFT UNIVERSITY OF TECHNOLOGY

UNet-based Fully-automatic Segmentation of the Capitate from CT Images

Master Thesis (BM51032#32ECTs)

Wenli Xue

(Master student BME program, 1284584)

DATE: 11-02-2021

Period: Jan 2020 – March 2021

Supervisors: Dr. ir. Geert. J. Streekstra (Department of Biomedical Engineering and Physics Academic Medical Center)

Dr. Iwan.Dobbe (Department of Biomedical Engineering and Physics Academic Medical Center)

Ir. Jorg. Sander (Department of Biomedical Engineering and Physics Academic Medical Center)

Prof. Dr. ir. Jaap Harlaar (Department of Biomechanical Engineering

Delft University of Technology)

Table of Contents

Abstract

Osteoarthritis (OA) is a degenerative joint disease and imposes an increasing burden on individuals and public health systems. Most prevalent joints are the knee, hip and hands, including the wrist. In order to enable early treatment of wrist OA, an early-detection method of cartilage loss, a characteristic symptom of OA, is needed. , CT images of the wrist bones. cannot visualize the cartilage itself, but instead use the distance between adjacent bones, to estimate the cartilage thickness. To enable such estimations, bones need to be segmented, which is a laborious task, that would impede any early diagnosis implementation. So automated segmentation of the wrist bones is desired for cost effective and objective assessments. However fully automatic segmentation of CT images is still a technical challenge. Deep learning techniques are considered a potentially successful approach to automate image processing.

The aim of this study is therefore to design and validate an automatic segmentation method of the capitate from CT-images based on a deep learning approach .

For the automated segmentation method of CT images we selected UNet, a type of Convolutional Neural Network. A total of 10 CT images of the capitate, were divided into 3 groups to train (6), validate (2) and test (2) the network, while their corresponding segmented images were used as ground truth. Training and validation set were used during training to build the model, while test set was used after training to evaluate the performance of the model.

Quantitative evaluation of similarity between automatic segmentation by the network and the ground truth was expressed by the Dice coefficient (test data 1: 0.94, test data 2: 0.91) and the Hausdorff distance (test data1: 2.06mm, test data2: 2.55mm). Automatic segmentation took 6.7s for test data 1 and 8.1s for test data 2.

The proposed approach holds promise for applications in fully automatic segmentation of wrist bones, as its performance, characterized by Dice coefficient and Hausdorff distance, is in par with those from other techniques of the same application. The next step in successful clinical implementation of the method is to improve the accuracy, for instance, by using a larger data size, after which the model can be further applied as an automatic quantitative metric in diagnosis of wrist OA.

1. Introduction

Osteoarthritis (OA), a most common degenerative cartilage disease, imposes enormous burden on individuals and healthcare systems[1]. In the year 2020, it was estimated that 25% of the population over the age 18 is affected by OA[2]. Early-detection of cartilage loss could reduce burden of OA as well as enable the development of effective early treatment and prevention. Radiographic joint space width (JSW), an indirect cartilage thickness measurement, is the current gold standard for establishing the diagnosis of OA in clinical practice. Radiographs can clearly depict bony structures; however, it is not capable of direct visualization of cartilage. Therefore, radiographic cartilage thickness is estimated indirectly by the assessment of JSW of adjacent bones. While radiographic JSW is used in clinical routine for OA diagnosis, it is severely limited by its inability of 3D visualization, due to over-projection[3]. With the existence of advanced 3D imaging techniques, over-projection, which is caused by projecting 3D structures onto a 2D plane in regular radiography, can be overcome.

Prior to the present study, a literature review was conducted to investigate state-of-the-art 3D imaging techniques that claim to measure cartilage thickness. The main conclusions are twofold. Firstly, it is becoming increasingly difficult to ignore the significant performance of Convolutional neural network (CNN) on medical image analysis[4]. Secondly, the absence of studies on fully-automatic wrist cartilage from CT images need to be addressed. A brief discussion of the literature review is given below.

Figure 1 is an overview of the degree of automation of the methods found in the literature study. Despite the fact that fully-automatic methods are more favorable, since they minimize the human input and thus reduce the human error, the majority of reported methods require a certain degree of human effort. The degree of automation of the methods is determined by the degree of automation of segmentation, which is a process of extracting objects of interest from an image[5]. Segmentation is a fundamental step for cartilage thickness measurement. CNN is a research hotspot at present for fully automatic medical image segmentation. The reason is that it has achieved astonishing performance on medical image segmentation tasks since 2012[6]. Some studies indicate that CNN-based segmentation methods can achieve better performance on medical images, when compared with human experts[4]. An overview of different segmentation methods can be found in the Appendix.

The number of papers on fully automatic methods are presented in Figure 2, which indicates that most of those methods studied MRI images of the knee and

hip. Although OA most often occurs in knee and hip, the wrist joint is also often affected by OA. While MRI permits direct visualization of cartilage, it has difficulties to depict cartilage layers thinner than 1 mm[7]. Wrist joints contain cartilage layers thinner than 1mm, therefore it would be challenging to measure wrist cartilage the thickness using MRI images. Instead, CT provides clear visualization of wrist bones, which would allow an indirect estimation of cartilage thickness via assessing JWS between adjacent bones.

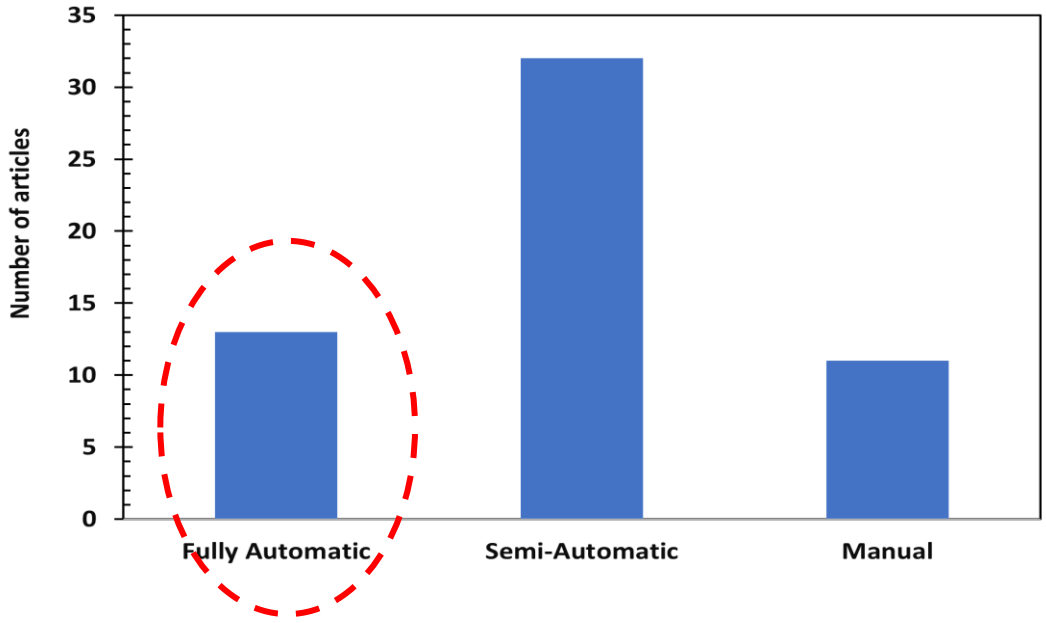


Figure 1. Degree of automation: semi-automatic (31), fully-automatic (13) and manual (11).

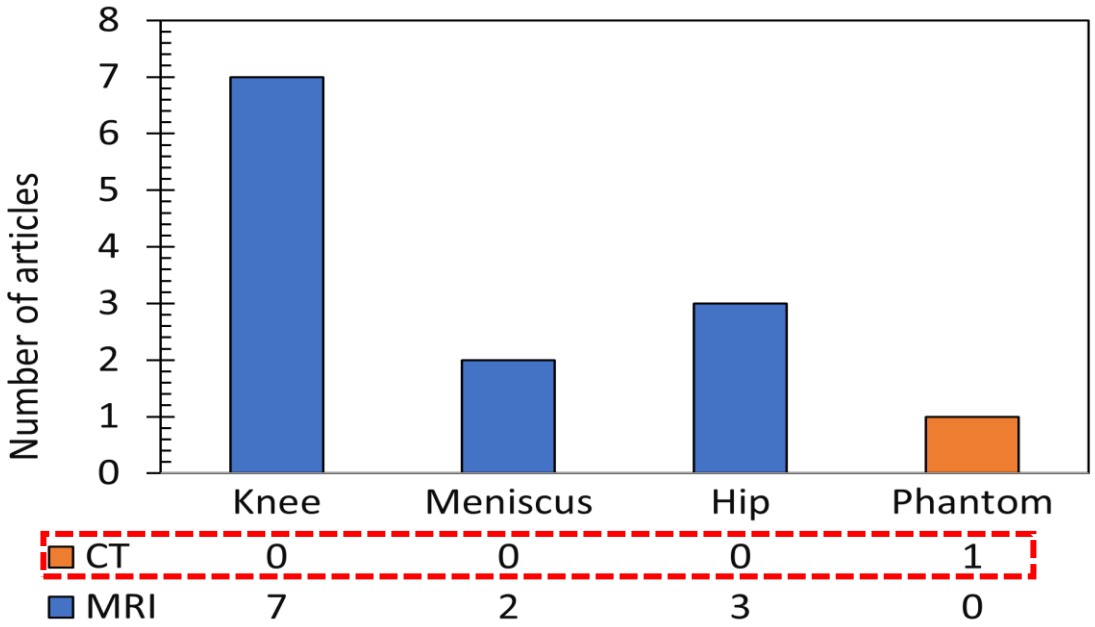


Figure 2. Image sources and subjects in reported studies of fully automatic methods.

In recent years, **CNNs** have been widely used for segmentation of medical images, and excellent results have been reported[4, 6, 8-11]. Therefore, the objective of the present study is to investigate the feasibility of a CNN-based approach to segment capitate bone of the wrist using CT images, in order to narrow the research gap in the fully automatic segmentation of wrist bones. The assessment hypothesis is that CNN-assisted segmentation results have the visual quality, Dice coefficient and Hausdorff distance comparable to those reported in other studies, which are about segmentation of wrist bones with various imaging analysis technologies.

2. Terminology

In this section, the terms that are consistently used throughout this report are explained.

Learnable parameter in the present study refers to the parameter that is learned automatically through the training process.

Weight is used as an alternative for learnable parameter, however, it is specifically employed to describe the learnable parameter outside the convolutional layers, for instance in the fully connected layers.

Hyperparameter is the parameter that is not able to be learned during the training process and need to be pre-set before the training process starts.

Kernel/filter refer to the sets of learnable parameters.

Label stands for the pre-segmented image which is used as the gold standard in this study.

Ground truth is an interchangeable word for label.

Batch is a portion of the complete data. The complete data is divided into smaller groups before processing and each group is called a batch.

Batch size refers to the number of samples in a batch.

Patch is an area or region of an image.

Patch size defines the size of a patch.

Epochs refer to the number of times that the network goes through the complete data. One epoch is completed once the network has processed the entire data.

Iterations is the number of batches that have been processed. During one iteration, one batch is processed.

3. Methods

In this study, a fully-automatic segmentation model which is able to segment CT images of the capitate bone, was generated using a CNN image classifier to assign every voxel of the CT images to a certain class. There are 2 classes: class background and class capitate bone. The data used in this study is from the work of De Roo et al[12]. A specific CNN known as UNet was deployed in this study. Instead of developing a 3D segmentation model, this study focused on the development of a 2D slice-wise segmentation model. 3D CT images were sliced in the Axial plane. The optimizer and loss function used here are Adam and cross entropy loss function as empirically they have achieved better performance in image classification. Detailed descriptions of CNN and CNN-based image segmentation can be found in the Appendix (9.2 Theory).

3.1 Data collection

CT images of right wrist joints and their corresponding label images of the capitate bone of 10 individuals were selected from the data of De Roo et al[25]. The height (x-axis) and width (y-axis) of the images range from 194 to 406 voxels and 384 to 425 voxels respectively. All images share a depth (z-axis) of 363 voxels. The images have a spatial resolution of 0.326mm along the x and y-axes, and 0.330mm along the z-axis. The CT images were segmentated using an in-house software of the Academic Medical Center (AMC), known as Articulus. The segmented images, referred to as label images, served as the ground truth in this study. Articulus is a multi-purpose software that is able to perform image segmentation as well as visualization and evaluation of the segmentation results. The segmentation of Articulus is a semi-automatic algorithm based on level set and region growing.

3.2 Data preparation

The collected raw data was pre-processed. There were three types of data preparation used in the present study: data split, patch selection and data augmentation.

Data split Image data and label images of 10 individuals were randomly split into 3 groups: training (6), validation (2) and test (2). The training group was used for training the model, the validation group was used to optimize the model by adapting hyperparameters and the test group was used to evaluate the performance of the final model (see Appendix 9.2.2.1 and 9.2.2.3).

Patch selection In a wrist joint CT image, the capitate bone occupies less than 1 percent of the voxels. Otherwise speaking, the number of voxels in class

“background” significantly outnumbers the voxels in class “capitate”. Therefore, the data is highly imbalanced. Highly imbalanced data affects the performance of the network adversely. To combat this issue, only cropped regions of the CT images which contain the capitate bone were used for training. Different patch sizes were tested including 32×32, 64×64, 128×128 and 256×256 voxel to find the patch size that optimizes the trade- off between computational time and performance of the network.

Data augmentation is a technique used in deep learning to increase data diversity, by applying transformation to data. The transformations used in this study were random crop, random rotation and random flip.

3.3 UNet

UNet, developed from traditional CNN, has achieved great results for segmenting medical images in recent publications[13-15]. UNet is characterized by its symmetric shape composed of a contracting path and expanding path (see Figure 3)[16]. The blocks of contracting and expanding path follow the architecture of traditional CNN layers: repeated convolutional layers each followed by a Relu and max pooling (see Figure 3—5). In the expanding path, the max pooling is replaced by up sampling. There are no fully connected layers in UNet, which allows the network to accept inputs of any size. In fully connected layers, the size of weight matrix, a hyperparameter, determines both input and output size. This is because in fully connected layers, every input is connected with every output by a weight that needs to be trained. However, in convolutional layers, the number of learnable parameters is independent of input and output size. Therefore a fully connected layer requires a fixed input size, while a convolutional layer accepts inputs of any size. Aforementioned, the pooling operation makes the network more robust to the changes in location of the feature in the input (see Appendix 9.2.1.3 pooling layer). Nevertheless, this could result in coarse segmentation due to the location information loss. To achieve fine segmentation, a connection is added between the corresponding contracting and expanding path. This connection bypasses pooling operations and concatenates the feature map of the contracting path with the feature map of the expanding path at the same level, which allows the localized context to be retained from contracting path of the network[16]. A detailed explanation of the contracting and expanding path is given in Figure 15 and 16, where an exemplary input image is used.

In the present study, the contracting path was composed of 4 times repeated application of two 3×3 convolutional layers, each followed by a Relu and a 2×2 max pooling with stride 2 (see Figure 4). Batch normalization was deployed before each Relu. Batch normalization, a technique used to stabilize the training

process, uses the mean and standard deviation of each batch.[16] At each step of the contracting path, the number of filters was doubled, starting with 32 filters at the first convolution (see Figure 4). In the expanding path, each step consisted of a 2×2 up sampling with stride 2, concatenate connection, and a 3×3 convolutional layer followed by a Relu (see Figure 5). At the last layer, a 1×1 convolution was used to reduce the number of output to the number of classes.

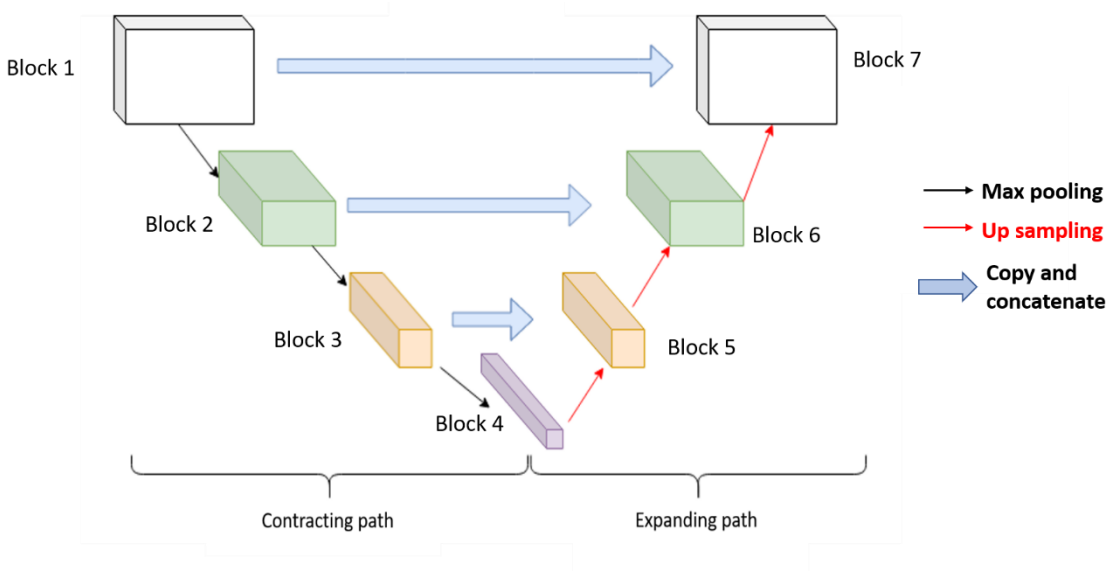


Figure 3. A diagram of UNet[17].

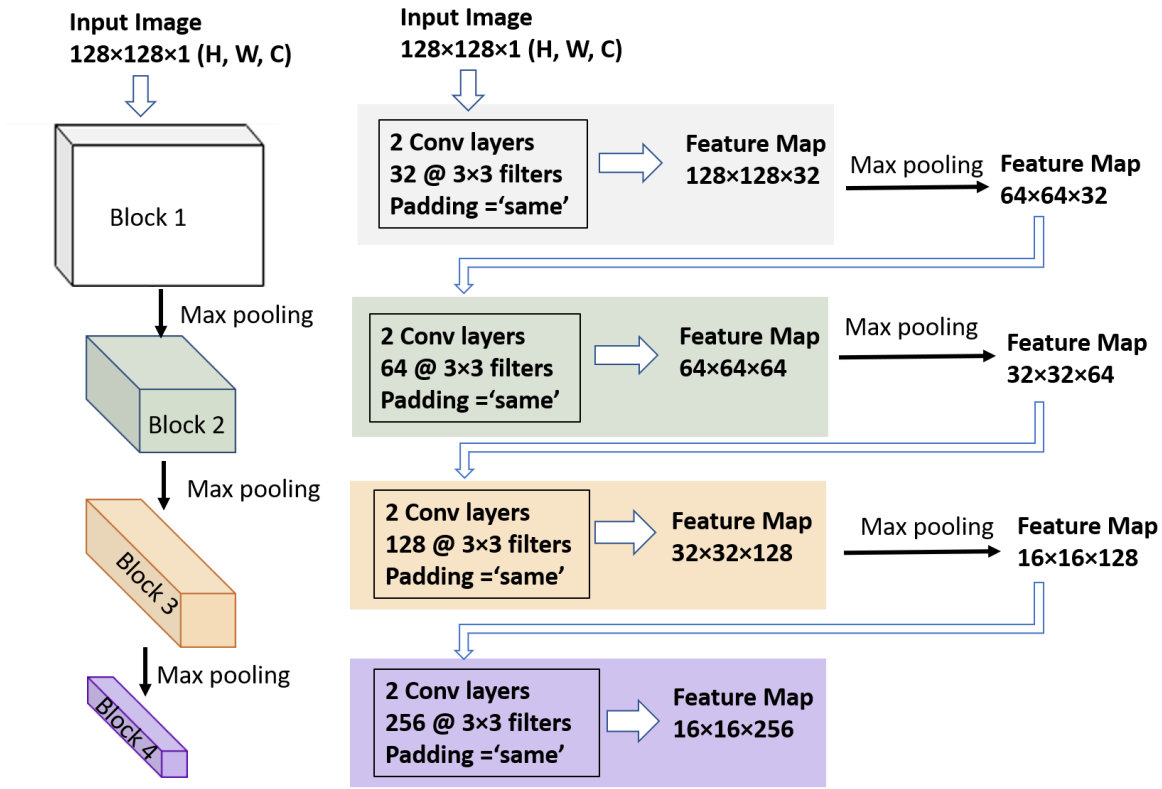


Figure 4. Detailed explanation of the contracting path. The input image has a size of 128,128,1 in height (H), width (W) and color channel (C) respectively. There are 4 blocks in the contracting path. Each block has 2 convolutional layers with each convolutional layer followed by a Relu. @ means times and thus 32 @ 3×3 filters means 32 filters and each filter has a size of 3×3. The number of filter doubles at each step, starting with 32 filters. Padding = 'same' is used to preserve the height and width of the input image during convolution. Max pooling has the size of 2×2 with strides 2. On the contracting path, the heigh and width of image gradually decreases, while the depth gradually increases. Starting from 128×128×1 to 16×16×256.

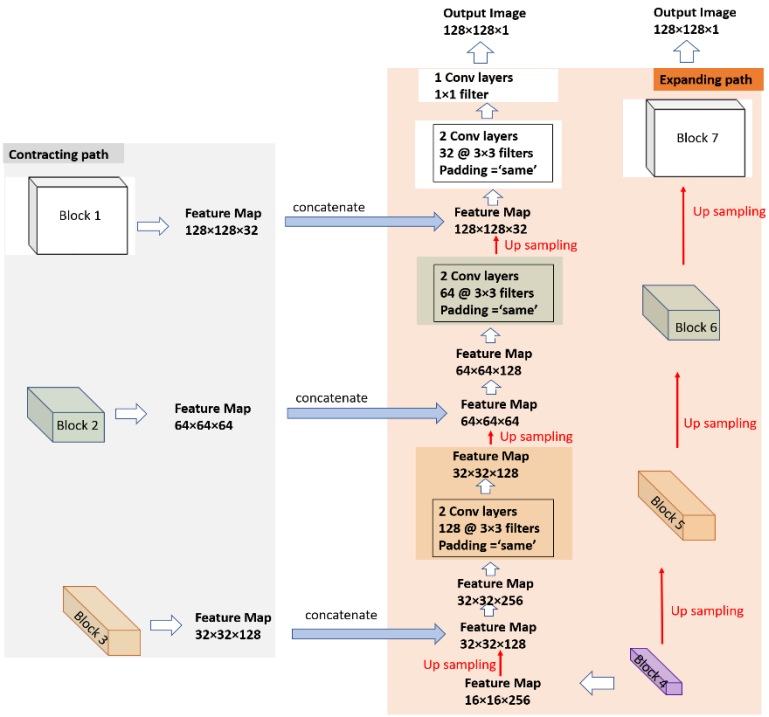


Figure 5. Detailed explanation of the expanding path. In the expanding path, each block consisted of a 2×2 up sampling with stride 2, concatenate connection, and a 3×3 convolutional layer followed by a Relu. In the expanding path, the height and width of the image gradually increases, while the depth gradually decreases. Starting from 16×16×256 to 128×128×1.

3.4 Loss function and optimizer

In the present study, cross entropy loss function combined with Adam optimizer was used, as the combination is typically used in medical image segmentation tasks. Learning rate is an important hyperparameter as it determines how much the learnable parameters are adjusted at each update. A higher learning rate allows a faster converge, at the cost of missing the global minima. Therefore various combinations of learning rate and weight decay were examined, in order to find the suitable pair. The values of learning rate used were 0.1, 0.01, 0.001 and 0.0001. Weight decay of value 0, 0.1 and 0.001 were tested.

3.5 Training and Validation

During the training process, both training set and validation set were used. The training set was used to train the network by adjusting the learnable parameters to minimize the loss. The validation set was used for frequent evaluation of the trained network to help fine tune the hyperparameters and select the best model. The best model is a combination of a set of learnable parameters and hyperparameters that has the best performance on the validation set.

The code of the present study was implemented in Python 3.7.6 and pytorch was used as seep learning framework. The training was run on a NVIDIA GPU, namely NVIDIA GeForce RTX 2080 SUPER, which has a total memory of 8192MB. CUDA version 11.0 was used in this study. The training process was monitored via Tensorboard, a visualization toolkit that is able to visualize metrics as well as images during training and therefore gives an insight into the training process. By doing so, the inputs and outputs of the network were constantly monitored, which helped to adjust the hyperparameters of the network. The network was training with 1000 iterations per epoch.

3.6 Evaluation

After training, the performance of the selected best model was evaluated using the test set. The evaluation in the present study was divided into two categories: visualization and metrics evaluation. The visualization evaluation was carried out by comparing the overlay images of the ground truth and segmentation produced by the model visually, using Articulus. Furthermore, evaluation metrics were introduced to quantitatively assess the difference between the segmentation generated by the model and its corresponding ground truth. The metrics included in this study were Dice coefficient, Hausdorff distance as well as a distance map produced by Articulus, and the computational time.

4. Results

The initial values of the hyperparameters are an educated guess, for instance the initial patch size is set to 32×32 as it provides a balance between sufficient context and reasonable computational time. The segmentation outputs generated by the initial values of hyperparameters are send to Tensorboard to be visually evaluated and then optimized until the optimal set of hyperparameters are found, which is able to balance the performance of the network and the computational time.

4.1 Preliminary results

The network had difficulties detecting the capitate when the patch size was set to 32×32 or 64×64, or the learning rate was set to 0.1, 0.01 or 0.001 with a weight decay of 0 or 0.1, or batch size with a value of 16 or 32 (see Figure 6). The training process is slow with 256 as patch size or 0.0001 as learning rate or 128 as batch

size. In comparison with patch size 256×256, patch size 128×128 allows a faster converge and similar segmentation output.

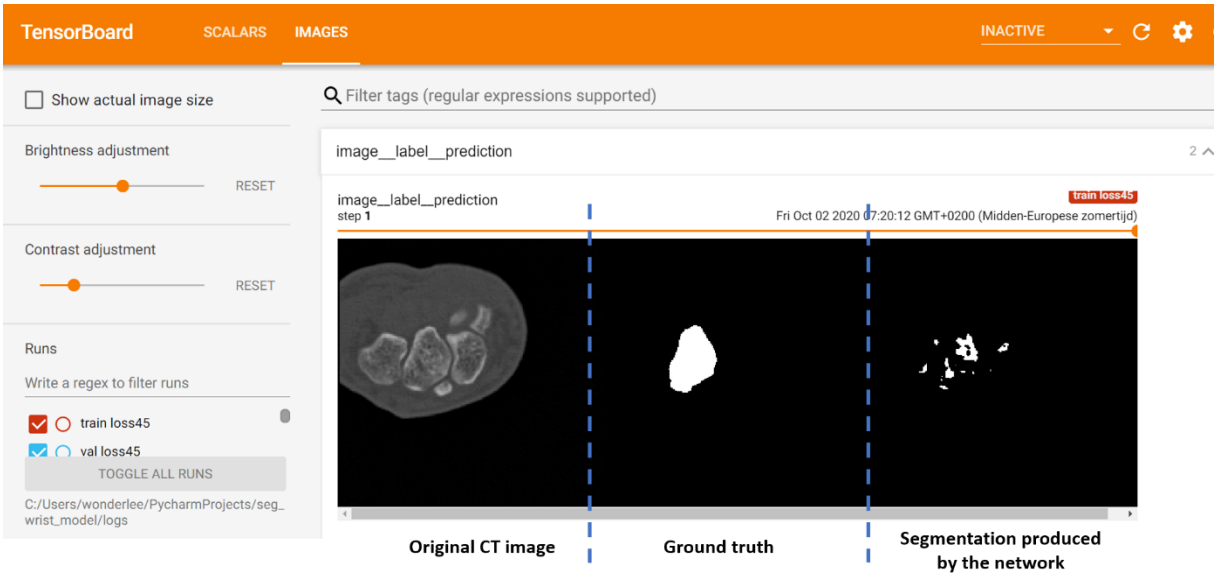


Figure 6. An example of unsuccessful segmentation produced by the model. The desired result would be the segmentation produced by the model (right) is similar to the ground truth (middle).

4.2 Optimal result

Through the experiments, the optimal set of hyperparameters was determined as: the value of patch size is 128×128, learning rate is 0.001 with weight decay of 0.001 and batch size was 64. This optimal set of hyperparameters is used to train the network in the present study. Figure 7 shows the loss evolution in both training and validation processes, as functions of iterations. Overfitting starts at epoch 7 (around 7000th iteration), as there was 1000 iterations per epoch. and therefore the model saved at this epoch is the selected model, which will be evaluated using the test data.

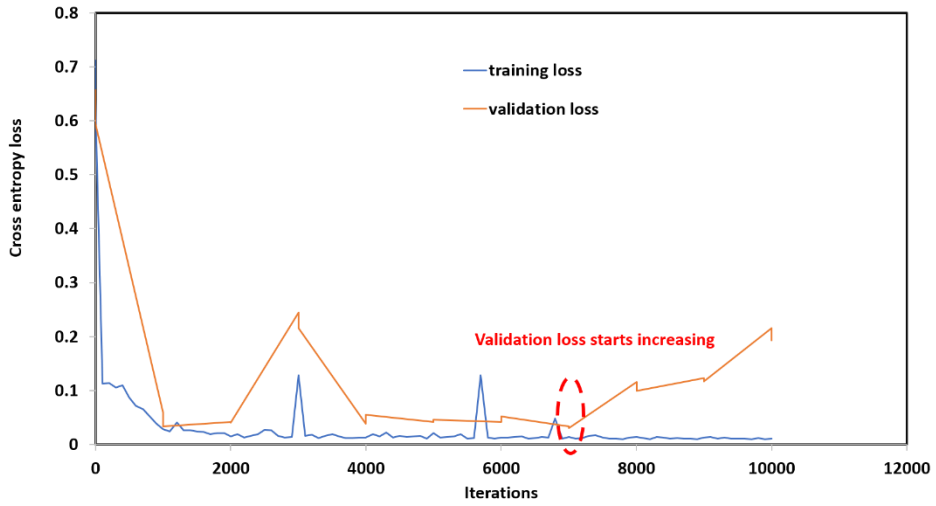


Figure 7. Training loss and validation loss versus iterations. This loss plot was obtained with patch size is 128×128, learning rate is 0.001 with weight decay of 0.001 and batch size was 64. The training was completed with 10 epochs and each epoch has 1000 iterations.

4.3 Evaluation

The evaluation is carried out by using the selected model to segment the test data. The segmentation produced by the model is compared with the ground truth visually and quantitatively. The 3D overlay image of segmentation produced by the model and its corresponding ground truth is generated by Articulus (see Figure 8). The overlay image is visually assessed slice by slice through 3 orthogonal planes.

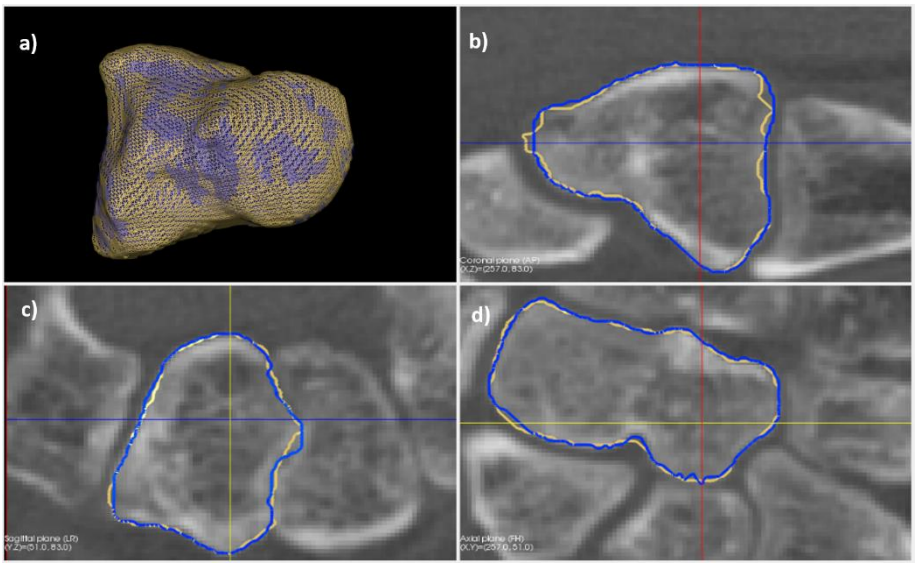


Figure 8. Overlay image of the segmentation produced by the model (blue) and its corresponding ground truth (gold). Image a is the 3D overlay image. Image

b, c and d shows a slice of the 3D overlay image viewed from coronal, sagittal and axial plane respectively.

The 3D distance map, which visualizes the distance between corresponding points of segmentation produced by the network and the ground truth, is created by Articulus to help evaluating the result. Figure 9 is a screenshot of the 3D distance map of test dataset number 1 and 2, and the scalars including the maximum, minimal and mean distance as well as the standard deviation are given together with the Dice coefficient, the Hausdorff distance and the computational time, in table 1. Figure 10 shows boxplots representing point-to-point distances between segmentation performed by the network (datasets 1 and 2) and the ground truth.

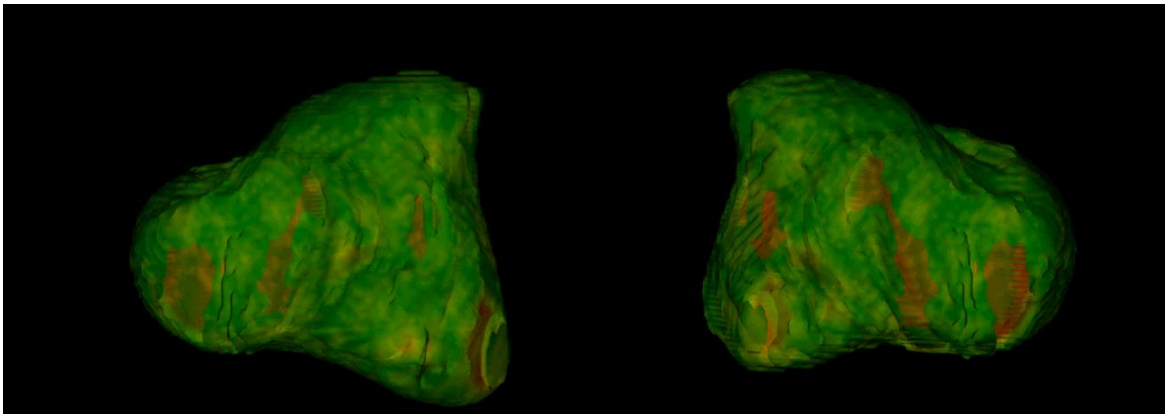


Figure 9. 3D distance map of test data 1(left) and 2 (right). Color map legend is defined as: green (0 mm), yellow (0.5mm), red (> 1mm).

Table 1. Quantitative evaluation of the segmentation results of test data 1 and 2.

	Dice coefficient	Hausdorff distance (mm)	Distance map (mm)				Computational time (s)
			Max	Min	Mean	Standard deviation	
test data 1	0.9388	2.0616	4.2939	0	0.4221	0.7017	6.7151
test data 2	0.9089	2.5475	5.0279	0	0.3539	0.7297	8.0743

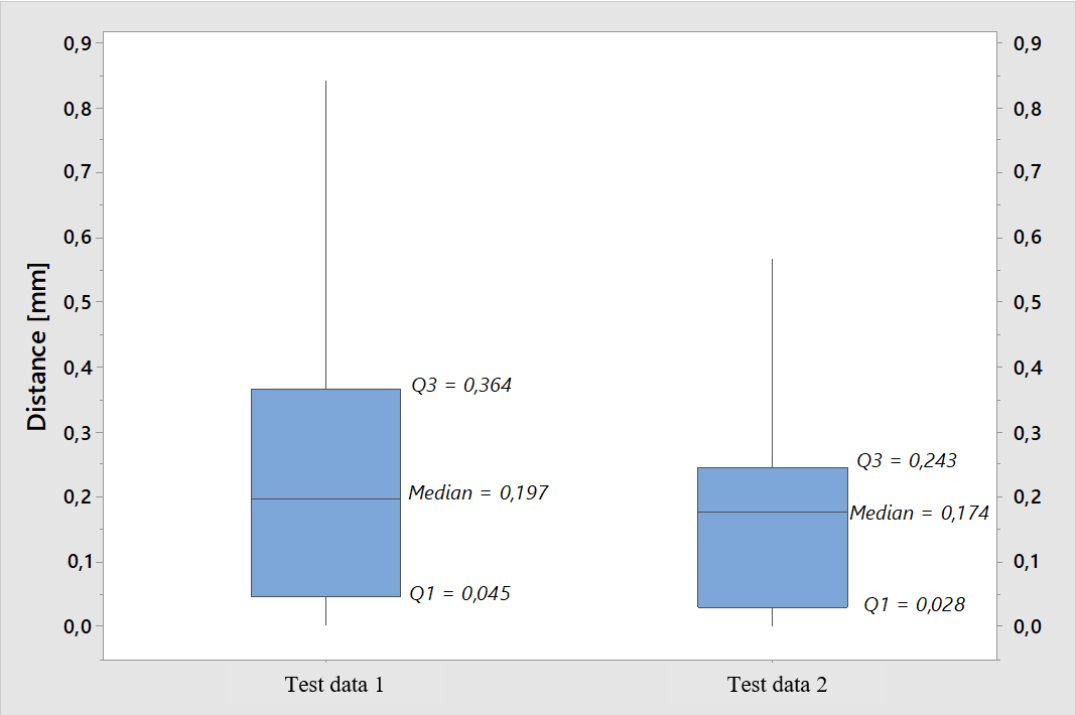


Figure 10. Boxplots of the distance distribution for distance maps from test data 1 and 2. The summary statistics used to develop the boxplot are the median of the data (line), the lower (25%) and the upper quartiles (75%) (box limits).

5. Discussion

The aim of the present study was to develop a UNet implementation for segmentation of the capitate bone and to evaluate its performance. The performance of the UNet is compared visually and quantitatively to the semi-automatic segmentation results obtained using Articulus, which served as the ground truth.

Visual evaluation was carried out by going through the 3D overlay image of UNet-assisted segmentation and its corresponding ground truth of test data 1 and 2 slice by slice respectively. For both test data, a substantial agreement, between the UNet-assisted segmentation and its corresponding ground truth, has been observed visually. It was noticed that most regions of UNet-assisted segmentation overlap its corresponding ground truth (e.g., figure 8). This is supported by the 3D distance maps from the test data. The 3D distance map computes and colors the distance between the point of segmentation produced by the model and its corresponding point from the ground truth. Figure 9 shows most regions of the distance map are green, which means that the distance between the corresponding points of the model segmentation and ground truth is close to 0. Therefore, the model segmentation highly agrees with the ground truth visually.

Quantitative evaluation was conducted by quantifying how much the UNet-assisted segmentation differs from its corresponding ground truth. Instead of manual segmentation, segmentation produced by Articulus, a semi-automatic in-house segmentation software, was used as ground truth in the present study. Although its accuracy level still has room for improvement, Articulus has been reported as a reliable segmentation tool in several studies[18-20]. There are three metrics used for quantitative evaluation: distance distribution, Dice coefficient and Hausdorff distance. The distribution of distances between UNet-assisted segmentation and its corresponding ground truth is visualized by a boxplot. Dice Coefficient and Hausdorff distance are commonly used in evaluating the performance of medical image segmentation methods[21-22]. Dice coefficient evaluates the performance of the segmentation over the entire image by measuring the similarity of UNet-assisted segmentation and its ground truth. Hausdorff distance indicates the largest segmentation error. Additionally, the computational time needed for segmentation is used as a secondary metric to evaluate the efficiency of the model. Detailed discussion of these metrics are presented as below:

1) Quantitative evaluation using boxplot: The distance maps of both test data have been further analyzed quantitatively by creating boxplots. The boxplot shows the distance distribution. For test data 1, the boxplot indicates that 75% of the differences between UNet-assist segmentation and the ground truth is smaller than 0.364mm (see figure 21 (left)). For test data 2, 75% of the distance lies below 0.242mm (see Figure 21 (right)). It would be desirable to keep the distance error below 0.1mm, since the maximum thickness of capitate cartilage is approximately 1mm[23]. This could be achieved by using a bigger data set and/or adding more data augmentation techniques. The model could also be improved by combining it with other segmentation techniques, for instance a UNet initialized level-set segmentation.

2) Quantitative evaluation using Dice Coefficient: The relatively low Dice coefficient that we found in this study was to be expected, since both the training and generalization of the model was based on a limited number of datasets: 6 and 2 individuals respectively. In contrast, other studies, in which larger dataset were included, reported much lower Dice coefficients.[24-25]. The obtained Dice coefficient value of test data 1 is 0.9388, while it is 0.9089 for test data 2 (see Table 1). The highest Dice coefficient that was reported in a study by Brui. et.al. was 0.81 for their CNN-assist wrist cartilage segmentation[24]. Meng. et.al., have developed a CNN-assist segmentation model for wrist joints and the obtained Dice coefficient value of this model is 0.78 ±0.06[25]. The low Dice coefficient values of those 2 studies might be caused by the type of images used for training.

One of those studies used MR images for CNN-assist wrist cartilage segmentation. MR has difficulties capturing cartilage thinner than 1mm, while wrist cartilage can be thinner[7]. Therefore, it could be challenging for the network to segment the cartilage layers from MR images, let alone measuring its thickness. In the study of Meng. Et.al., radiographic images of wrist bones have been used for training. Radiography is not sufficient to capture accurate 3D structures, which would lead to feeding the network with inaccurate information and thus have an adverse effect on the trained model. Overall, CT images might be considered as an optimal choice for CNN-assisted wrist bone segmentation.

3) Quantitative evaluation using Hausdorff Distance: The value of the Hausdorff distance for test data 1 and 2 is 2.0616mm and 2.5475mm accordingly (see Table 1). Those values are higher than the reported value of CNN-assist segmentation for wrist joints (1.56±0.30mm) by Meng et al.[25] and lower than the values (2.8±2.7mm) provided by Forster et al., using a level-set based semi-automatic carpal bone segmentation toolkit[26]. Hausdorff distance is determined exclusively by the largest segmentation error. A segmentation method can achieve accurate segmentation over most of the image, while having large error at one or a few locations. Therefore, lower Hausdorff distance values do not necessarily correlate with higher segmentation quality. However, a lower value of Hausdorff distance is preferred in this study, as it would lead to a more accurate estimation of the JSW at every location and thus more accurate diagnosis of OA. In clinical practice, the diagnosis of wrist OA is based more on the symptoms and physical examination than JSW due to the its complex nature. There is no exact threshold for JSW to diagnose wrist OA, since it may differ from individual to individual. For the same individual, it may change with age and JSW is location dependent, which varies over the entire contact region of a joint. Therefore, the change of JSW over time, instead of its absolute value, is of clinical interest in OA diagnosis. In order to provide a JSW-guided OA diagnosis, more researches are required to investigate the difference between OA and age related JSW change, as well as location precise JSW distribution over the entire contact area of a joint, for instance measuring the JSW during motion. An accurate segmentation technique with lower Hausdorff distance would be a first step towards those researches.

The low Hausdorff distance value of the study of Meng et al., could be explained by the remarkable size of training data[24], data of 290 individuals, whereas in the present study data of 6 individuals was used. Comparing to the present study, the study of Forster et al., has a relatively large set of training data of 80 individuals. However, the reported Hausdorff distance of their study is higher than the present study. The could occur because of weak or missing edges since MRI images were used in their study, which may not able to capture thin cartilage layers as wrist

cartilage. The Hausdorff distance of the present study could be improved by using a larger dataset. Moreover, a study of Karimi et al., has proposed a novel loss function for CNN-based segmentation that is able to reduce the Hausdorff distance by approximately 18-45%, comparing with commonly used loss functions e.g. cross entropy loss function, without comprising the Dice coefficient[22].

4) Quantitative evaluation using computational time: The computational time required for segmentation assisted by UNet has been improved by an order of magnitude, as comparing to manual segmentation, which typically requires minutes[24]. Semi-automatic segmentation methods can complete the task within several seconds, for instance Articulus is able to segment a Capitate around 10 seconds, however human input is needed. UNet-assisted segmentation completes segmentation in a few seconds without the need of human input. In the present study, the computational time for the test data 1 and 2 is 6.7151s and 8.0743s. Owing to the requirement of human input, semi-automatic methods are less efficient than fully automatic methods such as UNet-assisted segmentation.

It is noticeable that the loss curves have a few spikes (see Figure 7). The reason could be that the data used for training is unrepresentative and/or the learning rate is too high, which makes the network unstable[27]. Therefore, the solution would be adding more data and/or fine tune the learning rate.

The present study has several limitations. A small size training data with a single label of capitate bone has been used to develop the model. The performance of the model can be further improved by using more training data and/or more data augmentation techniques. However, acquiring medical images can be difficult, due to privacy and cost concerns. Therefore, the critical question is how much data is sufficient for achieving a specific target performance e.g. segmentation error is within 0.1 mm. It is difficult to give a particular number to this question, as the size of the data required is dependent on the nature of the problem such as the number of classes need to be classified, how different are the classes etc. It would be interesting for future study to investigate the impact of the size of data on the classification accuracy. Hypothetically, the classification accuracy rises with the increasing data size and eventually reaches a stable phase, where more data will not significantly improve the performance of the model.

Another limitation is that the model basically segments 2D images, which together makes up a 3D image. A 3D approach or an approach that segments in more planes and combines these results, may give a better 3D results.

Additionally, more labels can be added to the training data to further develop the model to be able to segment not only the capitate bone but other carpel bones as

well. Theoretically, adding more labels could improve the accuracy of segmentation, since there is more context for the model to learn[16].

6. Conclusions

In the present study, a fully automatic segmentation method for the capitate bone from CT images based on UNet has been developed. The proposed method has achieved a high degree of agreement with the ground truth both visually and quantitatively. Moreover, it significantly lessens the computational time, comparing to manual and semi-automatic methods. This study demonstrates the feasibility of a UNet-based fully automatic segmentation method for carpel bones from CT images and it is the first step towards a UNet-based fully automatic method that is able to measure the thickness of wrist cartilage.

7. Acknowledgement

Uiteindelijk, is the first word that came into my mind when I typed the last sentence of my thesis. Indeed, it is finally finished, after 1 year of hard-work, sweat and occasional tears, and the dissertation in front of me is the certificate of my effort and a record of a long journey.

A long journey that, luckily, I did not walk alone.

During my thesis project, I was blessed with the support and encouragement from many brilliant scientists. This chapter is devoted to you all.

I want to thank my supervisors from AMC: Dr. ir. Geert. J. Streekstra, Dr. Iwan Dobbe and ir. Jorg Sander. Geert and Iwan, I still remember the beginning of my project, when I asked you if I can work in your group for my thesis, after a memorable 4 months of internship and you said yes. Moreover, you both welcomed me back with open arms and provided all the support and supervision a student can ever ask for. I specifically enjoyed our interactions in the weekly meeting every Friday: you constantly challenged me in my topic and forced me to read into every single aspect of the technology, just so that I could provide the answers you were asking for, which in turn brought so much insight and stimulated my progress; when you were taking a break from challenging me, you could always noticed when I was not very sure about my work or my ability and provided me with your own life experiences, which every time cheered me up and motivated me to try harder. I also enjoyed our casual chats, after the project-related topics were done. I will always remember Geert's enthusiasm about his

boat trip and Iwan's frustration when he was forced to work at home when his kids were at home.

Jorg, you were my supervisor and guide in deep-learning and Python programming, a topic I have barely touched before starting this project. You did not just teach me the specific method in dealing with the topic of my project, more importantly, you showed me the correct way-of-working in deep-learning research, about how to critically evaluate my results and conclusions. The contents as well as the way-of-working will be the most valuable assets in my future career and I owe them to you.

I also want to show my gratitude to my supervisor from TU Delft, Prof. Dr. ir. Jaap Harlaar. Jaap, thank you so much for accepting me as your student, even when my topic is not directly linked to your research. When we had our meetings, I was always impressed by your sharp questions and comments, which always triggered me to view my study from a different angle. I really appreciated your support on both contents and scientific thinking.

Now that my project is done and I will move on to my next journey, I want to thank you all again for all the support and supervision in the past year, for without them, my thesis project would have been much harder. I sincerely hope our paths will cross again.

8. References

1. Kloppenburg, M.; Berenbaum, F., Osteoarthritis year in review 2019: epidemiology and therapy. *Osteoarthritis and Cartilage* **2020,** *28* (3), 242-248.
2. Chen, D.; Shen, J.; Zhao, W.; Wang, T.; Han, L.; Hamilton, J. L.; Im, H.-J., Osteoarthritis: toward a comprehensive understanding of pathological mechanism. *Bone Res* **2017,** *5*, 16044-16044.
3. Segal, N. A.; Frick, E.; Duryea, J.; Nevitt, M. C.; Niu, J. B.; Torner, J. C.; Felson, D. T.; Anderson, D. D., Comparison of tibiofemoral joint space width measurements from standing CT and fixed flexion radiography. *J. Orthop. Res.* **2017,** *35* (7), 1388-1395.
4. Yadav, S. S.; Jadhav, S. M., Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data* **2019,** *6* (1), 113.
5. Nadipally, M., Chapter 2 - Optimization of Methods for Image-Texture Segmentation Using Ant Colony Optimization. In *Intelligent Data Analysis for Biomedical Applications*, Hemanth, D. J.; Gupta, D.; Emilia Balas, V., Eds. Academic Press: 2019; pp 21-47.
6. Rawat, W.; Wang, Z., Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation* **2017,** *29*, 1-98.
7. Streekstra, G. J.; Strackee, S. D.; Maas, M.; ter Wee, R.; Venema, H. W., Model-based cartilage thickness measurement in the submillimeter range. *Medical Physics* **2007,** *34* (9), 3562-3570.
8. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; Fei-Fei, L., ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* **2015,** *115* (3), 211-252.

9. Ma, J.; Song, Y.; Tian, X.; Hua, Y.; Zhang, R.; Wu, J., Survey on deep learning for pulmonary medical imaging. *Frontiers of Medicine* **2020,** *14* (4), 450-469.
10. Kermany, D. S.; Goldbaum, M.; Cai, W.; Valentim, C. C. S.; Liang, H.; Baxter, S. L.; McKeown, A.; Yang, G.; Wu, X.; Yan, F.; Dong, J.; Prasadha, M. K.; Pei, J.; Ting, M. Y. L.; Zhu, J.; Li, C.; Hewett, S.; Dong, J.; Ziyar, I.; Shi, A.; Zhang, R.; Zheng, L.; Hou, R.; Shi, W.; Fu, X.; Duan, Y.; Huu, V. A. N.; Wen, C.; Zhang, E. D.; Zhang, C. L.; Li, O.; Wang, X.; Singer, M. A.; Sun, X.; Xu, J.; Tafreshi, A.; Lewis, M. A.; Xia, H.; Zhang, K., Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell* **2018,** *172* (5), 1122-1131.e9.
11. Hesamian, M. H.; Jia, W.; He, X.; Kennedy, P., Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges. *Journal of Digital Imaging* **2019,** *32* (4), 582-596.
12. Dobbe, J. G. G.; Roo, M. G. A. d.; Visschers, J. C.; Strackee, S. D.; Streekstra, G. J., Evaluation of a Quantitative Method for Carpal Motion Analysis Using Clinical 3-D and 4-D CT Protocols. *IEEE Transactions on Medical Imaging* **2019,** *38* (4), 1048-1057.
13. Gadosey, P. K.; Li, Y.; Adjei Agyekum, E.; Zhang, T.; Liu, Z.; Yamak, P. T.; Essaf, F., SD-UNet: Stripping Down U-Net for Segmentation of Biomedical Images on Platforms with Low Computational Budgets. *Diagnostics (Basel)* **2020,** *10* (2), 110.
14. Song, W.; Zheng, N.; Liu, X.; Qiu, L.; Zheng, R., An Improved U-Net Convolutional Networks for Seabed Mineral Image Segmentation. *IEEE Access* **2019,** *7*, 82744-82752.
15. Liu, L.; Cheng, J.; Quan, Q.; Wu, F.-X.; Wang, Y.-P.; Wang, J., A survey on U-shaped networks in medical image segmentations. *Neurocomputing* **2020,** *409*, 244-258.
16. Goodfellow, I.; Bengio, Y.; Courville, A., *Deep Learning*. The MIT Press: 2016.
17. Adaloglou, N., Intuitive Explanation of Skip Connections in Deep Learning **Mar 23, 2020**.
18. Bor, P.; Colman, K.; Dobbe, J.; Stull, K.; Streekstra, G.; Oostra, R.-J.; Rijn, R.; De Boer, H.; van der Merwe, A., *Towards "virtual" forensic anthropology: the accuracy of 3D skeletal reconstructions from full body CTscans*. 2016.
19. Colman, K.; Dobbe, J.; Stull, K.; Ruijter, J.; Oostra, R.-J.; Rijn, R.; van der Merwe, A.; De Boer, H.; Streekstra, G., The geometrical precision of virtual bone models derived from clinical computed tomography data for forensic anthropology. *International Journal of Legal Medicine* **2017,** *131*.
20. de Roo, M. G. A.; Dobbe, J. G. G.; Ridderikhof, M. L.; Goslings, J. C.; van der Horst, C. M. A. M.; Beenen, L. F. M.; Streekstra, G. J.; Strackee, S. D., Analysis of instability patterns in acute scaphoid fractures by 4-dimensional computed tomographic imaging – A prospective cohort pilot study protocol. *International Journal of Surgery Protocols* **2018,** *9*, 1-5.
21. Taha, A. A.; Hanbury, A., Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Med Imaging* **2015,** *15*, 29-29.
22. Karimi, D.; Salcudean, S. E., Reducing the Hausdorff Distance in Medical Image Segmentation With Convolutional Neural Networks. *IEEE Transactions on Medical Imaging* **2020,** *39* (2), 499-513.
23. Gilbert, S. M., DC; Casey, JA; Crisco, JJ, Quantification of Carpal Cartilage Facet Morphology using Micro-CT. (55th Annual Meeting of the Orthopaedic Research Society).
24. Brui, E.; Efimtsev, A.; Fokin, V.; Fernandez, R.; Levchuk, A. G.; Ogier, A. C.; Melchakova, I.; Bendahan, D.; Andreychenko, A., *Deep learning-based fully automatic segmentation of wrist cartilage in MR images*. 2018.
25. Liang Kim, M.; Azira, K.; Muhamad Hanif Ahmad, N.; Maryam Kamarun, N.; Belinda, P.-M.; Yan Chai, H.; Maheza Irna Mohamad, S.; Khin Wee, L., Carpal Bone Segmentation Using Fully Convolutional Neural Network. *Current Medical Imaging* **2019,** *15* (10), 983-989.
26. Foster, B.; Joshi, A. A.; Borgese, M.; Abdelhafez, Y.; Boutin, R. D.; Chaudhari, A. J., WRIST: A WRist Image Segmentation Toolkit for carpal bone delineation from MRI. *Comput Med Imaging Graph* **2018,** *63*, 31-40.
27. Smith, L. N. In *Cyclical Learning Rates for Training Neural Networks*, 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), 24-31 March 2017; 2017; pp 464-472.
28. Starmans, M. P. A.; van der Voort, S. R.; Castillo Tovar, J. M.; Veenland, J. F.; Klein, S.; Niessen, W. J., Chapter 18 - Radiomics: Data mining using quantitative medical image features. In *Handbook of*

Medical Image Computing and Computer Assisted Intervention, Zhou, S. K.; Rueckert, D.; Fichtinger, G., Eds. Academic Press: 2020; pp 429-456.
29. Meyer-Baese, A.; Schmid, V., Chapter 13 - Computer-Aided Diagnosis for Diagnostically Challenging Breast Lesions in DCE-MRI. In *Pattern Recognition and Signal Analysis in Medical Imaging (Second Edition)*, Meyer-Baese, A.; Schmid, V., Eds. Academic Press: Oxford, 2014; pp 391-420.
30. Sharma, N.; Aggarwal, L. M., Automated medical image segmentation techniques. *J Med Phys* **2010,** *35* (1), 3-14.
31. Minnema, J.; van Eijnatten, M.; Kouw, W.; Diblen, F.; Mendrik, A.; Wolff, J., CT image segmentation of bone for medical additive manufacturing using a convolutional neural network. *Computers in Biology and Medicine* **2018,** *103*, 130-139.
32. Jain, V., Everything you need to know about "Activation Functions" in Deep learning models. **2019**.
33. Yamashita, R.; Nishio, M.; Do, R. K. G.; Togashi, K., Convolutional neural networks: an overview and application in radiology. *Insights into Imaging* **2018,** *9* (4), 611-629.
34. LeCun, Y.; Bengio, Y.; Hinton, G., Deep learning. (1476-4687 (Electronic)).
35. .
36. Ramachandran, P.; Zoph, B.; Le, Q. V. Searching for Activation Functions 2017, p. arXiv:1710.05941. https://ui.adsabs.harvard.edu/abs/2017arXiv171005941R (accessed October 01, 2017).
37. Nair, V.; Hinton, G. E., Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Omnipress: Haifa, Israel, 2010; pp 807–814.
38. Glorot, X.; Bordes, A.; Bengio, Y., Deep Sparse Rectifier Neural Networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Geoffrey, G.; David, D.; Miroslav, D., Eds. PMLR: Proceedings of Machine Learning Research, 2011; Vol. 15, pp 315--323.
39. Ke, Q.; Liu, J.; Bennamoun, M.; An, S.; Sohel, F.; Boussaid, F., Chapter 5 - Computer Vision for Human–Machine Interaction. In *Computer Vision for Assistive Healthcare*, Leo, M.; Farinella, G. M., Eds. Academic Press: 2018; pp 127-145.
40. Pandya, M. D.; Shah, P. D.; Jardosh, S., Chapter 3 - Medical image diagnosis for disease detection: A deep learning approach. In *U-Healthcare Monitoring Systems*, Dey, N.; Ashour, A. S.; Fong, S. J.; Borra, S., Eds. Academic Press: 2019; pp 37-60.
41. Sharma, A., Understanding Activation Functions in Neural Networks. **30 Mar, 2017**.
42. Das, R.; Chaudhuri, S., *On the Separability of Classes with the Cross-Entropy Loss Function*. 2019.
43. Cao, J.; Su, Z.; Yu, L.; Chang, D.; Li, X.; Ma, Z. In *Softmax Cross Entropy Loss with Unbiased Decision Boundary for Image Classification*, 2018 Chinese Automation Congress (CAC), 30 Nov.-2 Dec. 2018; 2018; pp 2028-2032.
44. Martinek, V., Cross-entropy for classification. **May 22, 2020**.
45. Kingma, D.; Ba, J., Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* **2014**.
46. Ruder, S., An overview of gradient descent optimization algorithms. **2016**.
47. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y., Random Erasing Data Augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence* **2020,** *34* (07), 13001-13008.

9. Appendix

9.1Segmentation

Segmentation is a process of extracting objects of interest from an image[5]. It is a fundamental step for cartilage thickness measurement. In the literature review,

most commonly used segmentation methods were divided into three groups: manual, semi-automatic and fully-automatic and evaluated. Manual segmentation by an expert is commonly accepted as the ground truth. While this segmentation technique utilizes expert knowledge, it is very time consuming and it is not highly precise due to inter-observer and intra-observer varibility[28-29].

There is a variety of approaches available for semi-automatic segmentation. In the next paragraphs, a brief discussion of the most used semi-automatic segmentation techniques including: thresholding, region growing and edge are presented. More semi-automatic segmentation techniques and detailed description of those techniques can be found in the study of Neeraj Sharma[30].

Thresholding, the most popular segmentation technique for image segmentation, uses an optimal threshold to separate the pixels in foreground and background. An advantage of this segmentation method is that it is simple and easy to implement, however its main drawback is that the performance of this method is influenced by the presence of artifacts.

Region growing is widely used in medical image segmentation and it is based on the assumption that pixels within a region have similar properties. The process is initialized with a seed point, which is selected according to the criteria for homogeneity (mostly its intensity similarity). The neighboring pixels with similar properties are clustering together to form a region. The region is growing continuously until all pixels with similar properties are assigned to this region. The main limitation of this method is that the segmentation result is largely dependent on the choice of seed point.

Edge detection method relies on the detection of edges, the boundaries that separate distinct segments. Image gradient is on the basis of edge detection methods e.g. Hessian matrix, Canny and Laplacian. The boundaries are generated by combining the detected edges and then different regions are separated. While this technique has been frequently deployed, it is highly noise sensitive which means image noise can be misclassified as edge pixels.

Fully-automatic segmentation is another approach for image segmentation. There are mainly three types of techniques used in this approach: statistical model-based segmentation, Atlas-based segmentation and CNN based segmentation.

In case of **statistical model-based segmentation**, a statistical feature representative model of the population is used. The presence of noise and artifacts cannot affect the performance of this method and thus it is better suited for medical image segmentation, while comparing with the segmentation techniques mentioned above[30-31]. However, this method may fail to provide accurate

segmentation result when an individual is not represented by the statistical model[31].

Atlas refers to pre-labeled reference images, which can be chosen randomly from the population. Although there is evidence that Atlas guided segmentation is able to compete with manual segmentation[30], its performance significantly depends on the selection of Atlas. The idea Atlas should be able to reflect the feature varieties of the entire population.

In recent years, **CNN** has been rapidly recognized as a segmentation technique in medical image process, since excellent segmentation results from medical images produced by CNN have been reported[4, 6, 8-11]. The main advantages of CNN-based segmentation are: 1). It is capable of automatically and adaptively learning the features solely from data by itself, instead of requiring hand-craft features provided by experts. 2). It is computationally efficient once the model has been trained; 3). It is possible to work with a small dataset. 4). It is possible to provide radiologist level results. The main drawback of CNN-based segmentation is that the process is considered as a black box, as the decision making is not visualized. This makes the implementation of CNN in clinical practice challenging.

9.2Theory

In this section, the background knowledge and techniques required for better understanding of this study are discussed including the basics of a CNN and its architecture, image segmentation techniques and semantic segmentation.

9.2.1 What is CNN?

CNN is a special type of deep learning model that is inspired by the human visual cortex and it is designed to automatically and adaptively learn patterns such as lines and curves from low to high level. There are 3 types of layers/building blocks mostly used to build a CNN including convolutional layer, a pooling layer and a fully connected layer. Stacking a repetitive sequence of the former two layers i.e., several convolutional layers and a pooling layer, followed by one or more fully connected layers yields a CNN. The repetitive sequence of convolutional layers and a pooling layer perform feature extraction, while the fully connected layers transfer the extracted features into output such as classification.

9.2.1.1 Convolutional layer

A convolution layer, a core building block of the CNN, is a combination of a set of learnable parameters called kernels or filters and an activation function. Kernels/filters perform a linear operation i.e., convolution on the input, whereas

the activation function adds nonlinearity that represents the biological neuron behavior to a CNN[32].

Convolution

Convolution is a mathematical operation that convolves the input with kernels/filters. In a CNN, the input is an array of numbers and therefore, a CNN takes an input image as an array of pixel values. A kernel is defined by two hyperparameters. Hyperparameters are parameters that can't be learned through training and need to be pre-set by the operator including kernel size and number of kernels. Kernel size should be smaller than the size of its input. Increasing the number of kernels could increase the capacity of a CNN to learn more complex features.

During the convolution, each kernel moves along the height and width of the input image, as it multiplies the values in the kernel with the pixel values of the input array at every position, till all positions are crossed. The multiplications of every position are summed up to produce a single output value associated with the position (see Figure 1). The matrix that contains the output values of all positions is called a feature map. This procedure is repeated for all kernels to create feature maps that represent different features of the input.

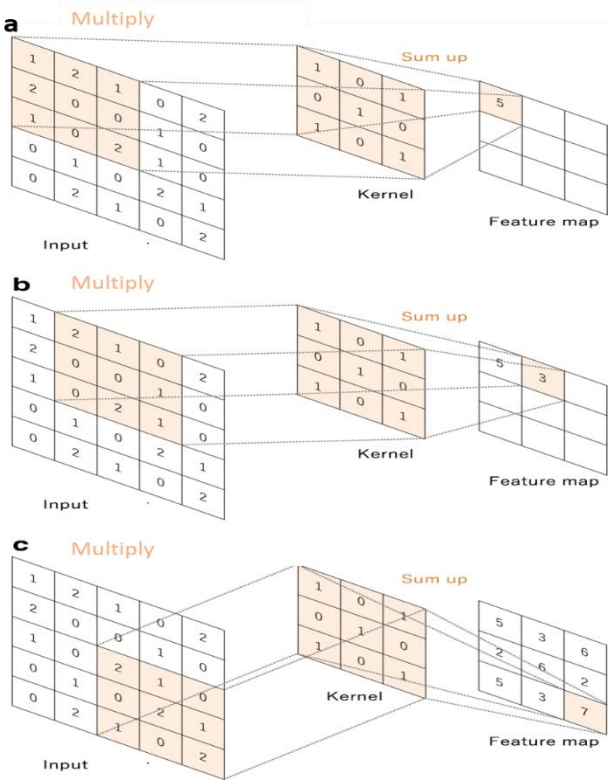


Figure 1. An illustration of the convolution operation (modified from Fig.3 from Yamashita et al.(2018)[33]). A kernel with size 3×3, stride 1and no padding slides across the input till all positions are covered. The values in the kernel are

multiplied by the values in the input array at every position and the multiplications of a position are summed up to obtain a single output value. The matrix that contains the output values associated with all positions of the input is called a feature map.

The size of a feature map is controlled by 3 hyperparameters: stride, padding, filter size and depth (see Equation 1). Stride is the distance between two consecutive kernel positions. Padding the image boarder with zeros would allow the boarder pixels to play a bigger role in determining the output, since the convolution operation is not able to place the center of a kernel on the outmost components of the input. Depth is the number of kernels used.

$$W = \frac{w-F+2P}{S} + 1, \quad H = \frac{h-F+2P}{S} + 1 \qquad \text{Equation 1}$$

Where *W*, *H* are the width and height of feature map, *w*, *h* are width and height of the input image, *F* is the size of the filter, *P* is amount of zero padding and *S* is the stride. Therefore, the size of feature map is H ×W × Depth.

9.2.1.2 Nonlinear activation layer

The outputs of each convolution layer are then passed through a nonlinear activation function such as Relu, sigmoid or hyperbolic tangent (tanh) function (see Figure 2). The main objectives of nonlinear activations are to add nonlinearity and differentiability to the network. A convolution is a linear operation which is inadequate to be used to model complex datasets such as images and videos, since those datasets contain multiple dimensions and can't be simply represented by linear transformations. Differentiability is important because it is mandatory for performing backpropagation. Backpropagation is an algorithm used to optimize learnable weights during training (See section 9.2.2.2.3). The mostly used nonlinear activation function is Relu[33-38], which gives an output x if x is positive, otherwise it gives 0 ($f(x) = \max(0, x)$). The reason that Relu is the most commonly used activation function in CNN is that the model uses Relu requires less training time and achieves better performance[16]. A detailed discussion about different activation functions can be discovered in the book Deep learning by Goodfellow. There is no learnable parameters or weights in any activation layer.

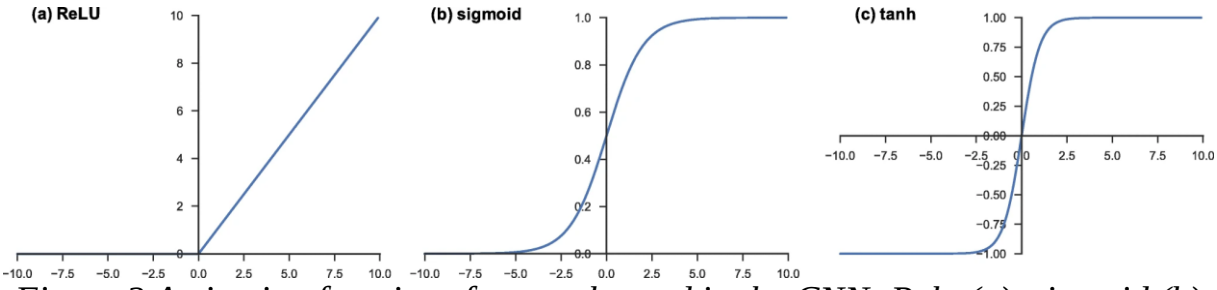


Figure 2.Activation functions frequently used in the CNN: Relu (a), sigmoid (b) and tanh (c)[33].

9.2.1.3 Pooling layer

A pooling layer, placed in-between consecutive convolutional layers in CNN, is a form of down-sampling, which adds location translation invariance to the network as well as reduces the computational time by reducing the learnable parameters[16]. The output feature map of the convolutional layer is sensitive to the location of the features in the input, which means small changes in the location of a feature in the input e.g. cropping and rotation of the input can result in different feature maps. A pooling layer down-samples the output feature map of the convolutional layer and creates an approximate version of the feature map that is more robust to the changes in the position of the feature in the input[16]. There is no learnable parameter in any pooling layer, whereas there are hyperparameters i.e., kernel size, stride and padding in a pooling layer which are similar to convolutional layers. There are two types of pooling: max pooling and average pooling. Max pooling extracts salient features such as edges, whereas average pooling picks smooth features. Max pooling has become the default pooling operation because empirically it achieves better performance[33, 39-40].

Max pooling

Max pooling returns the maximum value of the area of the input covered by the kernel and dismisses all the other values (see Figure 3).

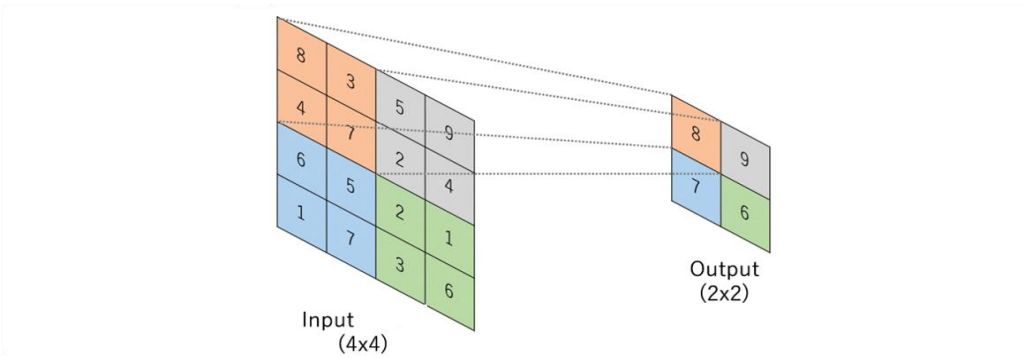


Figure 3.An example of max pooling (modified from Fig.6 from Yamashita et al.(2018)[33])). A kernel with size 2×2, strid 2 and no padding slides cross the input and outputs the maximum value in each position it covers, while dismisses all the other values. The size of output is smaller than the initial input after max pooling.

Average pooling

Another pooling operation is average pooling. Average pooling returns the average value, rather than the maximum value, of all the values from the area covered by the kernel (see Figure 4).

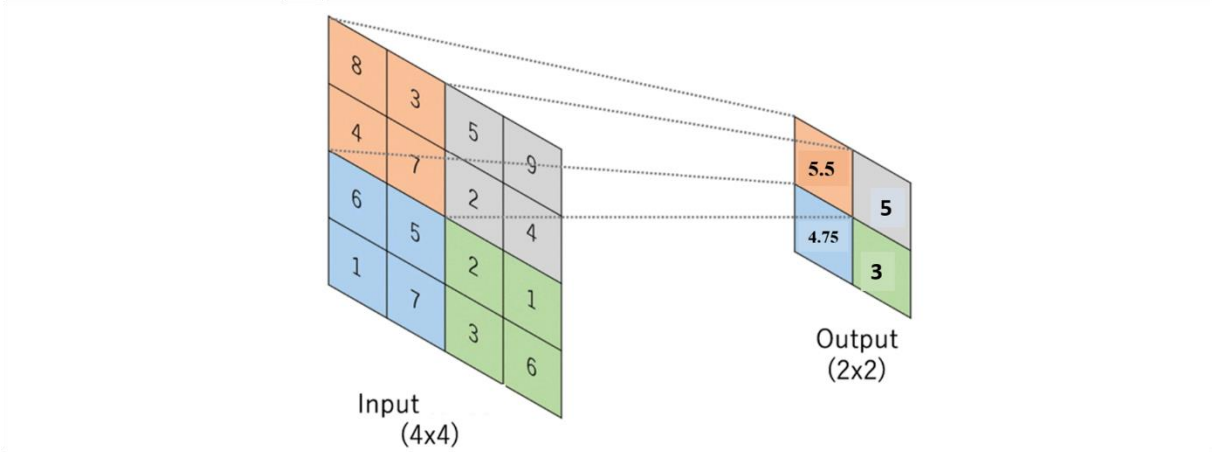


Figure 4.An example of average pooling (modified from Fig.6 from Yamashita et al.(2018)[33])). A kernel with size 2×2, strid 2 and no padding slides cross the input and outputs the average value in each position it covers, resulting in down-sampling of the input.

9.2.1.4 Fully connected layer

The output of the final pooling layer is flattened into a 1D vector and passes through one or more fully connected layers, otherwise known as dense layers. In fully connected layers, every output of the previous layer is connected to every input of the successive layer by a weight, which needs to be learned through the training process. The number of outputs of the final fully connected layer is the same as the number of classes. Each fully connected layer is followed by a nonlinear layer.

9.2.1.5 Last activation layer

The last fully connected layer is usually connected to an activation function, which converts the output values of the last fully connected layer to probabilities of all classes. For a multi-class classification task, softmax is the common choice[33]. Softmax function is a generalized sigmoid function (see Equation 2-3). It often follows the last fully connected layer to yield probabilities of an input belonging to different classes. The sum of softmax probabilities always is 1. A higher value of softmax suggest a higher probability. Softmax has a number of outputs, representing the predicted probability of different classes. The output with the highest value represents the predict class (see Figure 5).

$$f(x)_{sigmoid} = \frac{\exp(x)}{\exp(x)+1} \qquad \text{Equation 2}$$

$$f(x_i)_{softmax} = \frac{\exp(x_i)}{\sum_{k=0}^{k-1} \exp(x_k)} \; for\; i = 1, 2 \ldots, k \qquad \text{Equation 3}$$

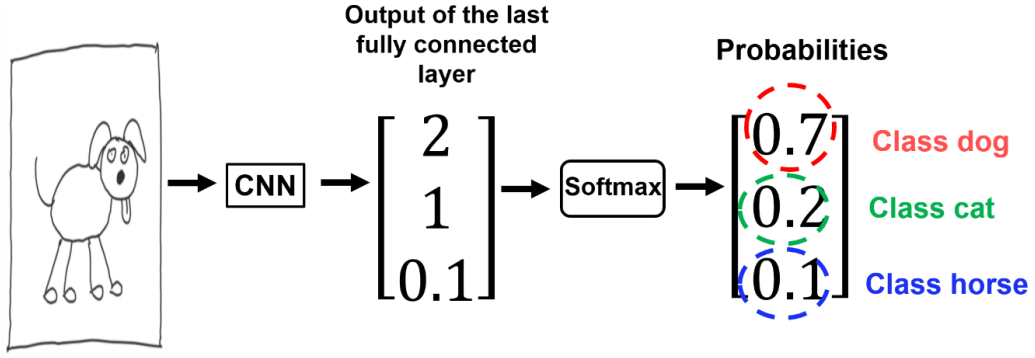


Figure 5. An example of softmax operation. There are three classes here: class dog, class cat and class horse. For the input image of a dog, , the probability of it belongs to class cat is 0.7, dog class is 0.2 and horse class is 0.1 The predicted class is class dog[41].

9.2.2 CNN-based image segmentation

Image segmentation is a process of partitioning an image into multiple segments of pixels[5]. Image segmentation can be otherwise considered as a process of classification of every pixel in an image, which can be achieved by using CNN.

9.2.2.1 Input data and labels

Input data and labels are the most critical elements in deep learning projects that are tailored to image segmentation. As a well-known proverb originating from computer science states: “Garbage in, garbage out.” The quality of the input data and labels affect the chance of success of a deep learning project. Therefore, The quality control of the input data and labels is mandatory. The collected input data and labels are normally split into 3 sets: a training set, a validation set and a test set (see Figure 6). A training set is used to train a model via forward propagation and back propagation, as described in the overfitting section. A validation set is used to monitor the training process, adjust the hyperparameters and make model selection decision (See section 9.2.2.3). A test set is used to evaluate the performance of the selected model at the very end of the project. Each set has its own function in the training process, which is explained in the overfitting section.

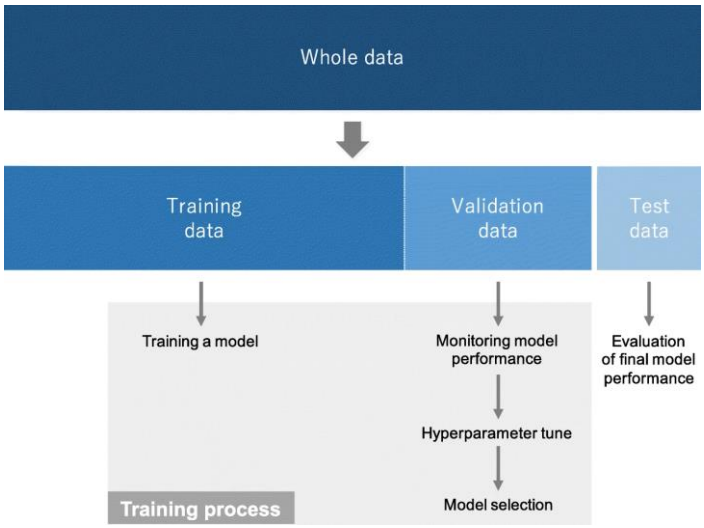


Figure 6. Collected data and labels are split into three sets: a : a training set, a validation set and a test set[33]. *Each set has its own purpose in the training process, which is discussed in the folowing sections.*

9.2.2.2 Training a CNN

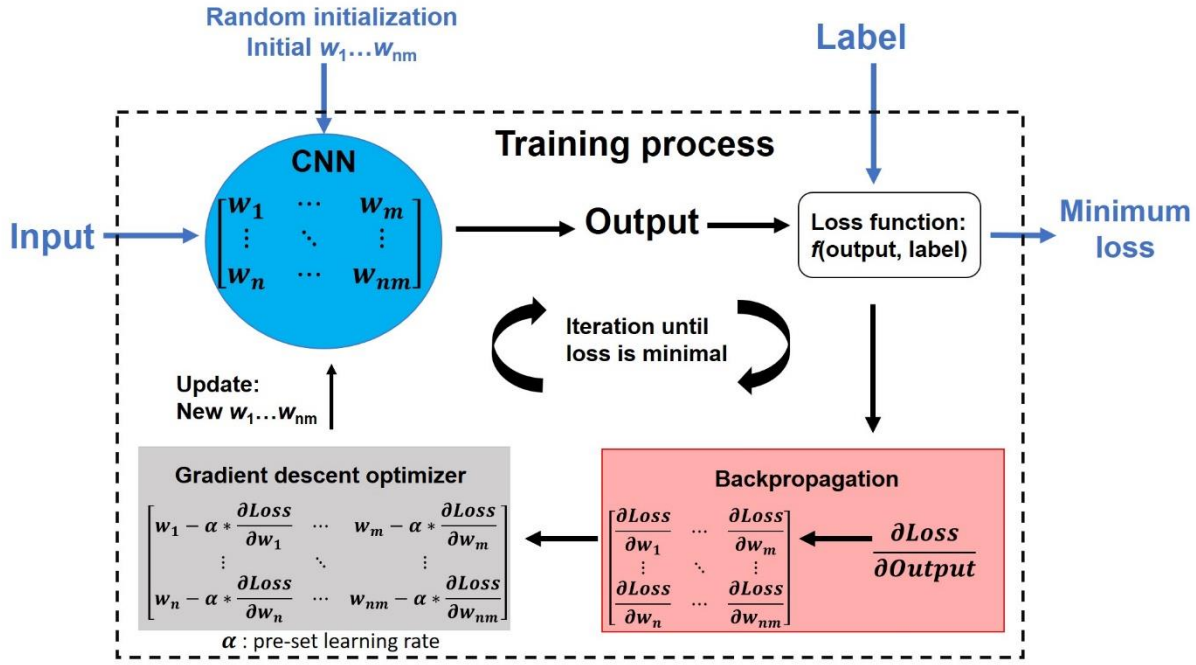
Training a CNN is a process of finding specific sets of learnable parameters for kernels in the convolutional layers and weights in the fully connected layers that minimize the differences between the ground truth and the output prediction. The training process is a repetitive process of forward propagation and backpropagation (see Figure 7). Forward propagation is an algorithm of calculating output from input. Backpropagation is an algorithm that computes the gradient of the loss in the reverse order: from output to input. Forward propagation

and backpropagation are executed with the assistance of the loss function and an optimizer.

Figure 7. Scheme of the training process.

9.2.2.2.1 Loss function

The loss function, also referred to as cost function, is used to calculate the



difference between output predictions of the model and the given ground truth labels. Cross entropy is a commonly used loss function for multiclass image segmentation[33, 42-43]. Cross entropy measures the difference between the predicted and true probability distributions for all classes. The predicted probability distribution, a vector, represents the predicted probabilities of all classes, summing up to 1. It is the output of the softmax layer. The true probability distribution is a vector with 1 for the actual class and 0 for all other classes (see Figure 8—9). Cross entropy decreases as the predicted probability converges to the ground truth and the ideal cross entropy value is 0 (see Equation 4).

$$Loss = -\frac{1}{N}\left(\sum_{i=1}^{N} y_i * \log\left(\hat{y}_i\right)\right) \qquad \text{Equation 4}$$

Where N is number of classes, y_i is the true probability of class i (0 or 1) and $\hat{y}_i$ is the predicted probability of class i (a value between 0 and 1).

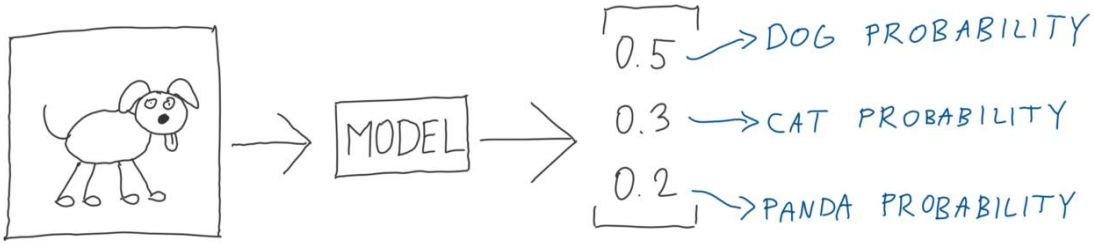


Figure 8.Image classification of 3 classes: dog, cat and panda[44]. The model gives predicted probability for each class. The probability of the image belongs to class dog, class cat and class pands is 0.5, 0.3 and 0.2. The sum of the probabilities of all classes is 0.5 + 0.3 + 0.2 =1.

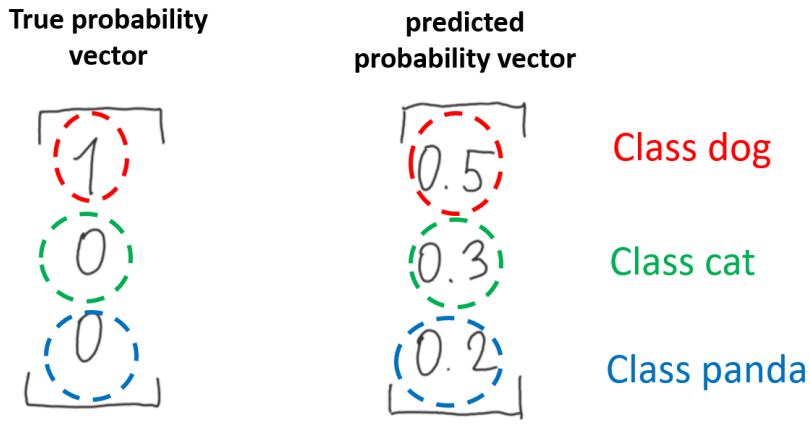


Figure 9.True probability vector (left) and predicted probability vector (right)of a dog image. True probability vector has value 1 for the class dog and 0 for other 2 classes. The prediction, produced by the model, gives class dog a probability of 0.5 and the other 2 classes a probability of 0.3 and 0.2 respectively.

9.2.2.2.2 Optimizer

An optimizer is an algorithm or method that iteratively updates the learnable parameters of the network in order to minimize the loss function (i.e., learnable parameters of kernels and weights between each connection of fully connected layers,).

Gradient descent, the commonly used optimization algorithm, calculates the gradient of loss function and iteratively moves in the direction of the steepest descent to minimize the loss function. Figure 10 illustrates this concept with a simplified example where a loss function has only one learnable parameter w. With multiple learnable parameters, the gradient is a vector of partial derivatives of the loss function with respect to all learnable parameters. The gradient descent algorithm has been studied extensively, resulting in many modified versions of the algorithm, which improve its performance. Among those modified gradient

descent algorithms, Adam is the most recommended one in medical image segmentation[45-46], which gives different learning rate for different learnable parameters.

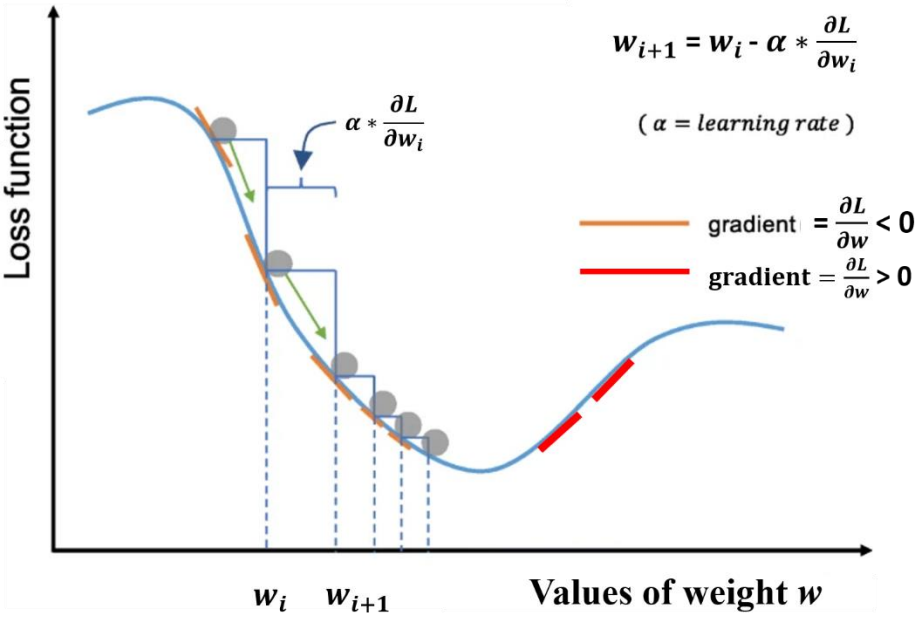


Figure 10. An illustration of how gradient descent minimizes loss function(modified from Fig.7 from Yamashita et al.(2018))[33]. Considering a simplified example as showing in the polt: a loss function with a single learnable parameter w In order to find the minimal loss, gradient descent starts with choosing a starting point w_i and then calculates the gradient of the loss function at this point. The gradient is a partial derivative of the loss with respect to w, which gives the direction of steepest increase of the loss function. The gradient descent algorithm takes a step in the negative direction of the gradient ($\frac{\partial loss}{\partial w} < 0$). The step size is determined by a hyperparameter known as learning rate (α).

9.2.2.2.3 Backpropagation

The backpropagation algorithm, which propagates the error from the loss function backwards through the network, calculates the partial derivatives of the loss function w.r.t each learnable parameter. The derivative associated with each learnable parameter determines how to change each learnable parameter in order to minimize the loss function (see Section 9.2.2.2.1). Figure 7 explains the backpropagation process training. The training starts with initialization of weights and propagates forward through the network. The outputs of the network are used to calculate the loss / error for the given set of weights. If the error is minimum, the model is ready to be tested on new input data. Otherwise, the weights will be updated using error backpropagation to minimize the loss function.

9.2.2.3 Overfitting

Overfitting is a condition when the model fits too well to the training data to the extent that it is difficult for the model to generalize well from the training data to a new set of unseen data, therefore, the performance of the model on a new data is poor. For instance, if the model memorizes irrelevant information or noise, instead of learning the general underlying pattern of the training data, it will make predictions based on noise rather than underlying patterns of the training data. In this case, the performance of the model on the training data can be very well, while very poor performance on an unseen, new data. A routine check for detecting overfitting to the training data is to monitor the training and validation loss[33]. If the model performs much better on the training data than on the validation data, then the model is likely overfitting to the training data (see Figure 11). There are several solutions to avoid overfitting including: training with more data, data argumentation, weight decay and early stop. Among those solutions, training with more data is the best solution[33], as a larger dataset provides more diversities, however, it is often hard to create a large dataset with medical images due to data privacy concerns[4, 33]. Data augmentation, modifies data via random transformation such as cropping, rotating and flipping to provide more information for the network to learn, is an effective method to limit overfitting[47]. Weight decay can limit overfitting by adding a regulation term to the loss function (see Equation 5). The regulation term encourages small weights and keeps the model simpler to avoid overfitting (see Equation 6)[16].

$$Loss_{new} = Loss_{original} + \frac{1}{2}\lambda\|w\|^2 \qquad \text{Equation 5}$$

$$w_{i+1} = w_i - \alpha\frac{\partial Loss_{new}}{\partial w_i} - \alpha\lambda w_i \qquad \text{Equation 6}$$

Equation 5 and 6 demonstrate how the regulation term helps to avoid overfitting. The regulation term works based on the assumption that less learnable parameters generate a simpler model, which is less prone to overfitting. Equation 5 is the new loss function with a L2 regulation, where λ is a coefficient and w is a set of learnable parameters. Equation 6 is the updated learnable parameters using the new loss function. Where w_{i+1} is the updated learnable parameter, w_i is the original learnable parameter and α is learning rate. The term $-\alpha\lambda w_i$ introduced by the regulation term leads decays all learnable parameters proportionally and thus keep them small.

Early stop refers to stopping the training at the point where overfitting occurs and the model saved at this point is the model selected for further evaluation. The performance of the selected model is evaluated by testing it on new unseen data,

known as the test set. The results from the test set confirm how well the model performs on new (unseen) data.

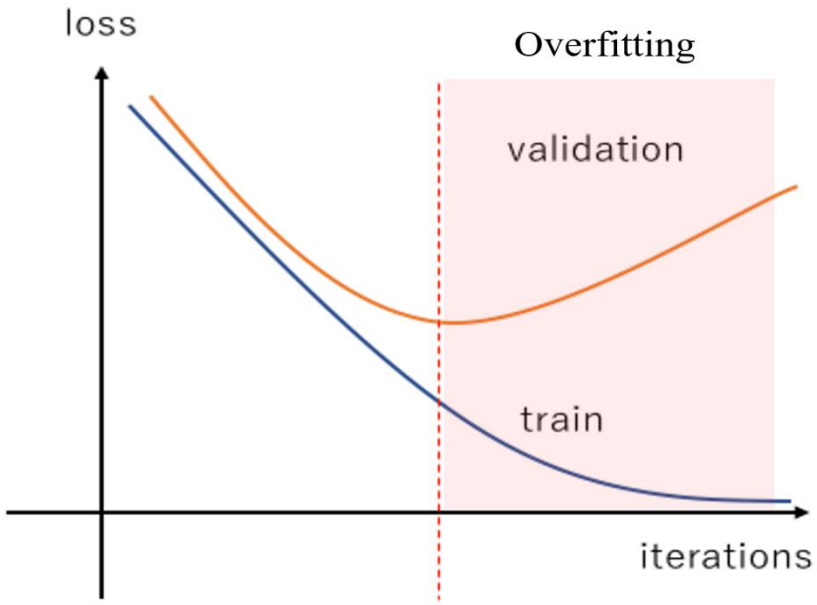


Figure 11. An illustration of monitoring training and validation loss to detect overfitting. Overfitting occurs when the validation loss starts to increase, whereas the training loss keeps decresing.