



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INTELLIGENT SYSTEMS

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

**DETECTION, EXTRACTION AND MEASUREMENT
OF THE CONTOUR AND CIRCUMFERENCE OF
THE METACARPAL BONES IN X-RAYS OF THE HU-
MAN HAND**

DETEKCIA, EXTRAKCIA A MERANIE KONTÚRY A ŠIRKY METAKARPÁLNYCH KOSTÍ ĽUDSKEJ
RUKY Z RÖNTGENOVÝCH SNÍMKOV

BACHELOR'S THESIS

BAKALÁŘSKÁ PRÁCE

AUTHOR

AUTOR PRÁCE

MATEJ OTČENÁŠ

SUPERVISOR

VEDOUCÍ PRÁCE

prof. Ing., Dipl.-Ing. MARTIN DRAHANSKÝ, Ph.D.

BRNO 2021

Bachelor's Thesis Specification



Student: **Otčenáš Matej**
Programme: Information Technology
Title: **Detection, Extraction and Measurement of the Contour and Circumference of the Metacarpal Bones in X-Rays of the Human Hand**
Category: Image Processing

Assignment:

1. Get familiar with image processing and segmentation of an X-ray of the hand.
2. Design algorithm for detection and extraction of contour and measurement of the circumference of the third metacarpal bone.
3. Implement the designed algorithm and test it on the provided database by the Department of Anthropology, Faculty of Science, Masaryk University.
4. Review the achieved results and discuss the possible extensions.

Recommended literature:

- STOLOJESCU-CRISAN, Cristina; HOLBAN, Stefan. An Interactive X-Ray Image Segmentation Technique for Bone Extraction. In: *IWBIO*. 2014. p. 1164-1171.
- JENA, Shweta; SAHU, Barnali; JAGADEV, Alok Kumar. Analysis of medical X-ray bone images using image segmentation. In: *Intelligent Computing, Communication and Devices*. Springer, New Delhi, 2015. p. 787-794.
- AREECKAL, Anu Shaju; SAM, Mathew; DAVID, S. Sumam. Computerized radiogrammetry of third metacarpal using watershed and active appearance model. In: *2018 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2018. p. 1490-1495. HUE, Tran Thi My; KIM, Jin Young; FAHRIDDIN, Mamatov. Hand bone radiograph image segmentation with roi merging. *Recent Researches in Mathematical Methods in Electrical Engineering and Computer Science*, 2011, 147-154.
- DAVIS, Luke M., et al. On the segmentation and classification of hand radiographs. *International journal of neural systems*, 2012, 22.05: 1250020.

Requirements for the first semester:

- Items 1 and 2.

Detailed formal requirements can be found at <https://www.fit.vut.cz/study/theses/>

Supervisor: **Drahanský Martin, prof. Ing., Dipl.-Ing., Ph.D.**
Head of Department: Hanáček Petr, doc. Dr. Ing.
Beginning of work: November 1, 2020
Submission deadline: May 12, 2021
Approval date: November 11, 2020

Abstract

This thesis aims to detect and subsequently extract the contour of the third metacarpal bone of the human hand from X-ray images and measure the circumference. The thesis describes segmentation of image using various methods for object detection which will be used for eventual measurements.

Abstrakt

Cieľom tejto práce je detekovať a následne extrahovať kontúru tretej metakarpálnej kosti ľudskej ruky z röntgenových snímok a zmerať jej šírku. Práca popisuje segmentáciu obrazu pomocou metód na detekciu objektov, ktoré sa následne využijú za účelom konečných meraní šírky kosti.

Keywords

Mask R-CNN, Detectron2, image preprocessing, image segmentation, pattern recognition, human hand detection, X-ray image, machine learning, watershed segmentation, Python, metacarpal bone, edge detection, noise reduction, computer vision, OpenCV

Klíčové slová

Mask R-CNN, Detectron2, predspracovanie obrazu, segmentácia obrazu, rozpoznávanie vzorov, detekcia ľudskej ruky, röntgenový snímok, strojové učenie, watershed segmentácia, Python, metakarpálna kosť, detekcia hran, odstránenie šumu, počítačové videnie, OpenCV

Reference

OTČENÁŠ, Matej. *Detection, Extraction and Measurement of the Contour and Circumference of the Metacarpal Bones in X-rays of the Human Hand*. Brno, 2021. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor prof. Ing., Dipl.-Ing. Martin Dražanský, Ph.D.

Rozšírený abstrakt

Úlohou tejto práce je vytvoriť algoritmus, ktorý bude schopný detekovať a extrahovať kontúru tretej metakarpálnej kosti, ktorá je umiestnená v strede ľudskej ruky a zmerať jej šírku v najjužšej časti tela kosti. Na úvod práca rozoberá súčasný stav v oblasti spracovania obrazu, segmentácie a detekcii útvarov v obraze. Rovnako je predstavený zjednodušený pohľad na ľudskú ruku z pohľadu morfológie, ktorý je nevyhnutný pre pochopenie celkovej problematiky, a súčasne pre stanovenie validnej metodiky, ktorá bude schopná sa vysporiadať s detekciou objektu tretieho metakarpu. V tejto časti sú taktiež predstavené tradičné metódy na predspracovanie obrazu a jeho segmentáciu, ale aj pokročilejšia metodika na detekciu a extrakciu útvarov, akou je napríklad konvolučná neurónová sieť.

V práci je následne vytvorená kapitola k využitiu tradičných metód na segmentáciu obrazu, ktorá popisuje neúspech pri použití na rôznych typoch neuniformných digitálnych röntgenových snímkoch. Táto časť môže slúžiť pre prípadné zlepšovanie segmentácie medicínskych snímkov pre obsiahlejšiu databázu ako záchytný bod. Ak by niekto pracoval s obdobnou problematikou, môže sa vďaka tomu vyhnúť neaplikovateľnosti metód, a tak zefektívniť svoj postup použitím vhodnejších metód.

Táto kapitola následne uvádza funkčný návrh a implementáciu algoritmu. Ten sa efektívne za použitia špeciálne dizajnovanej konvolučnej siete *Mask R-CNN*, pomocou knižnice *Detectron2*, vysporiadať s detekciou masky tretej metakarpálnej kosti, čo je pre celý algoritmus dôležitý krok. Táto kapitola popisuje postupný vývoj algoritmu od vytvorenia modelu na detekciu masky až po analytický návrh na zmeranie šírky kosti.

Práca sa následne venuje rozšíreniu algoritmu. Spôsob rozšírenia bol komunikovaný s ústavom antropológie Masarykovej univerzity, kde bolo navrhnuté rozšírenie, ktorého úlohou je zlepšiť analytické meranie kosti. Kosť je rozdelená na tri časti, a to ľavá a pravá kompakta a vnútorná dutina kosti. Úlohou rozšírenia je zmerať najjužšiu časť šírky kosti pre každú z týchto častí zvlášť, čo prinesie benefity pre antropologický ústav v prípade, že chcú preskúmať detailnejšie samotný vývoj kosti jedinca.

V krátkosti je v tejto kapitole popísaná aj adresárová štruktúra programu a spustenie programu ako terminálovú aplikáciu.

V tejto práci sa v ďalšej kapitole predstavujú štatistické výsledky, ktoré analyzujú konečnú presnosť algoritmu. Tieto výsledky sa venujú správne a nesprávne zmeraným vzdialenostiam či už pre celú šírku kosti alebo pre každú z troch individuálnych častí kosti zvlášť. Výsledky sú súčasne odrazom presnosti detekcie masky kosti, kde správne zmeranie šírky kosti je plne závislé od správne detekovanej masky. Kapitola súčasne popisuje štruktúru a pôvod databázy röntgenových snímkov. Sú tu taktiež podrobne analyzované jednotlivé problémy a nedostatky vytvoreného algoritmu a ich teoretické odstránenie či aspoň zmierenie. V krátkosti je popísaná optimalizácia algoritmu na rýchlosť a porovnanie časov pri spracovaní snímkov pred optimalizáciou a po optimalizácií.

Na záver je celkové zhodnotenie výslednej práce, kde je súčasne venovaná časť možným rozšíreniam či zdokonaleniam algoritmu.

Detection, Extraction and Measurement of the Contour and Circumference of the Metacarpal Bones in X-rays of the Human Hand

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of prof. Ing., Dipl.-Ing. Martin Drahanský Ph.D. The supplementary information was provided by doc. Miroslav Králik. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Matej Otčenáš
May 10, 2021

Acknowledgements

I would like to sincerely thank to prof. Ing., Dipl.-Ing. Martin Drahanský Ph.D, who gave me this opportunity to work on this thesis, for his ideas and advices which lead to completion of this work. I would also like to thank to doc. Miroslav Králik and Mgr. Anna Škultétyová from the Department of Anthropology of Masaryk University for their help with creation of algorithm and advices about biological structure of the human hands.

Contents

1	Introduction	3
1.1	Aim of the Thesis	4
1.2	Contents	4
2	State of the Art	5
2.1	Human Hand Anatomy	5
2.2	Third Metacarpal Bone Structure	6
2.3	Radiography	6
2.3.1	X-ray Image	7
2.4	Digital Image	8
2.5	Human Hand Digital Image Processing	9
2.6	Traditional Image Processing Methods	9
2.6.1	Grayscaleing	9
2.6.2	Noise Reduction	9
2.6.3	Thresholding	10
2.6.4	Adaptive Thresholding	11
2.6.5	Otsu's Thresholding	11
2.6.6	Canny Edge Detection	12
2.7	Medical Image Segmentation	14
2.7.1	Semantic Segmentation	16
2.7.2	Instance Segmentation	16
2.7.3	Watershed Segmentation	17
2.8	Supervised Learning	18
2.8.1	Mask Region-based Convolution Neural Network (Mask R-CNN)	18
2.9	Image Processing Tools	19
2.9.1	OpenCV	19
2.9.2	Detectron2	19
3	Designed Solution and Implementation	21
3.1	Analysis of Medical Images Issues	22
3.2	Median Noise Reduction	22
3.2.1	Proposed Otsu's Threshold	22
3.3	Application of Watershed Segmentation	22
3.4	Canny Edge Detector Applied for Watershed	23
3.5	Evaluation of Tested Methods Issues	23
3.6	Design for Pattern Recognition with CNN	24
3.6.1	Using Mask R-CNN by Detectron2	24
3.6.2	Custom Model Creation	24

3.6.3	Image Conversion to Compatible Format	24
3.6.4	LabelMe Annotation	25
3.6.5	Training Model	27
3.6.6	Loading Image Dataset	27
3.6.7	Creating Inference on Image Dataset	28
3.6.8	Image Binary Mask Creation	28
3.6.9	Contour Creation	29
3.6.10	Contour Coordinates Export	29
3.6.11	Analytical Bone Width Measurement	29
3.6.12	Bone Circumference Export	34
3.6.13	Collection of Resulting Data	35
3.7	Advanced Algorithm Extension	36
3.7.1	Model Creation for ROI	36
3.7.2	Bone Measurement in Three Separate Regions	37
3.8	Usage of the Algorithm	38
3.8.1	Directory Structure	39
3.8.2	Program Running Options	40
4	Testing on Given Database	41
4.1	Used Database	41
4.2	Measurement of the Bone Full Width	41
4.3	Measurement of the Three Separate Bone Regions	42
4.4	Issues of the Created Algorithm	42
4.4.1	C Language Optimization	43
4.5	Computation Time per Image	43
4.6	Hardware Resources of Testing Device	44
4.7	Comparison with Manual Measurements	44
5	Conclusion	46
	Bibliography	48
A	Comparison of Automated and Manual Measurements	53
B	Attached DVD Content	56

Chapter 1

Introduction

Nowadays, besides studying medical literature or performing long X-ray observations, there are other possible ways to carry out a detailed analysis of the human body to identify diseases. Thanks to technical progress it is possible to apply a wide range of algorithms, which can detect a variety of arising diseases not even a human eye can recognize.

Various organ disorders can be detected through application of these technical methods, such as examining lung X-ray to diagnose pneumonia, brain screens to brain tumour detection or metacarpal bones investigation to detect early stage of rheumatic disease. If we could prevent earlier such diseases we would be able to avoid more fatalities every day. One of the major parts in this segment is digital image processing.

This thesis describes the digital image processing to detect the contours of the metacarpal bones in the human hand which circumference is then measured. Such an approach aims to create a final program with an output database of measurement values which will serve the Faculty of Science of Masaryk University(MUNI) for detailed analysis thanks to which they avoid manual measurements, which can lead to inaccuracies caused by human error. This approach will ensure more accurate and effective evaluations of outcomes that monitor various adverse growth factors or detect genetic disorders.

To obtain such measurements, it is necessary to detect the individual edges of the bones as accurately as possible. However, this approach has to deal with several problems, such as the overlapping of individual metacarpal bones or the nonuniformity of the X-ray images.

Traditional image processing itself, consists of several steps. First, the image is loaded and the possible noise reduction is performed, which can be done using various filters, which will blur the image as a result. This is followed by image contrast enhancement and image segmentation, which separates the foreground from the background using the right algorithm technique. At last the edges of desired object can be detected with the following contour detection if needed.

However, this approach can often be insufficient and inaccurate when it comes to more complex extraction of objects in the image. The most serious problem can arise in image segmentation, where we try to extract a relatively unusual specific shape. This shape can be, for example, a kind of irregular polygon, which can be covered in the picture by many different but also similar shapes, which, however, are not the subject of research.

Thus, if one of the options presented, traditional methods for a larger amount of tested data begin to fail.

In such case, it can be better achieved by relatively more complicated, but more accurate methods for processing and segmenting the medical image, such as neural networks, more specifically convolutional neural networks(CNN).

1.1 Aim of the Thesis

This work aims to design and implement an algorithm that detects the contour of third metacarpal bone in the human hand from X-ray images and measure its circumference at the narrowest part. The thesis focuses on creating an algorithm to automate measurements on X-rays, which are provided by the Faculty of Science MUNI so that there is no need to measure and click points manually and to reduce the deviations that can arise as a human factor failure.

1.2 Contents

The chapter 2 focuses on the anatomy of the human hand itself, which is essential for a closer understanding of the problem of metacarpal bones detection. The chapter also describes current traditional image processing methods for pattern recognition using image segmentation and edge detection that can be used for X-ray images. Despite the usability of such methods, the chapter describes the possibilities of using deep learning with CNN, which are now very popular in segmenting the image and pulling a specific mask from the image. Chapter 3 describes a specific design of an algorithm for image segmentation of the bone using basic image preprocessing methods and their effects on the images. The section 3.6 in this chapter focuses on the use of convolutional neural networks, which appear to be more reliable in obtaining a mask for the third metacarpal bone, which does not require image preprocessing and also describes a proposal for analytical measurement of bone in its narrowest part. Furthermore, chapter explains the use of traditional methods in image processing in combination with CNN, such as Canny edge detector. On the basis of continuous communication with the anthropological institute, the section 3.7 describes a specific extension of algorithm, which is able to measure the individual narrowest parts of the bone in three separate regions. The bone structure itself is introduced in the section 2.2, which describes in detail what parts of the bone is talked about. The section 3.8 is focused on the proposed design, implementation and usage of an algorithm. The chapter 4 focuses on the testing of the program that was created on the given available database of manual measurements provided by the Faculty of Science MUNI. The chapter also focuses on hardware resources and possible speed optimization because of the complexity and computational difficulty of the algorithm. At last, the chapter 5 summarizes this thesis and discusses the possible extensions and improvements for the created algorithm.

Chapter 2

State of the Art

2.1 Human Hand Anatomy

Human hand is composed of number of bones, that are shown on image 2.1 of which every single one has its own purpose. Very important part of bone examination is reveal of genetic and growth disorders, which can be detected directly by observation of inner hand bones, for example metacarpal bones. This approach enables revelation of Turner's syndrome, dyschondrosteosis or hypochondroplasia [22]. X-ray images show the natural deformation of the bone itself, which occurs more frequently with the aging of the organism. The picture 2.2a shows frequent overlapping of the bones [37]. On the contrary, children's bones as seen on the picture 2.2b are predominantly separated from each other and do not overlap, which offers an advantage and simplification for segmentation of the image and detection of the edges. This phenomenon is caused by the fact that the skeleton in children is mostly composed of cartilaginous bones, which contain only small amount of inorganic substances and minerals such as calcium and this results in lower density of the bones. Growing older, these bones start to change and soft cartilage ossifies, and therefore the bone density grows bigger [36].

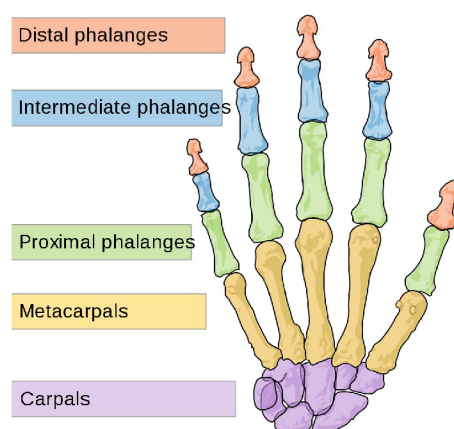


Figure 2.1: **Human hand bones.** Main interest of the human hand bones in this work are the metacarpals shown as yellow bones in the image [6].

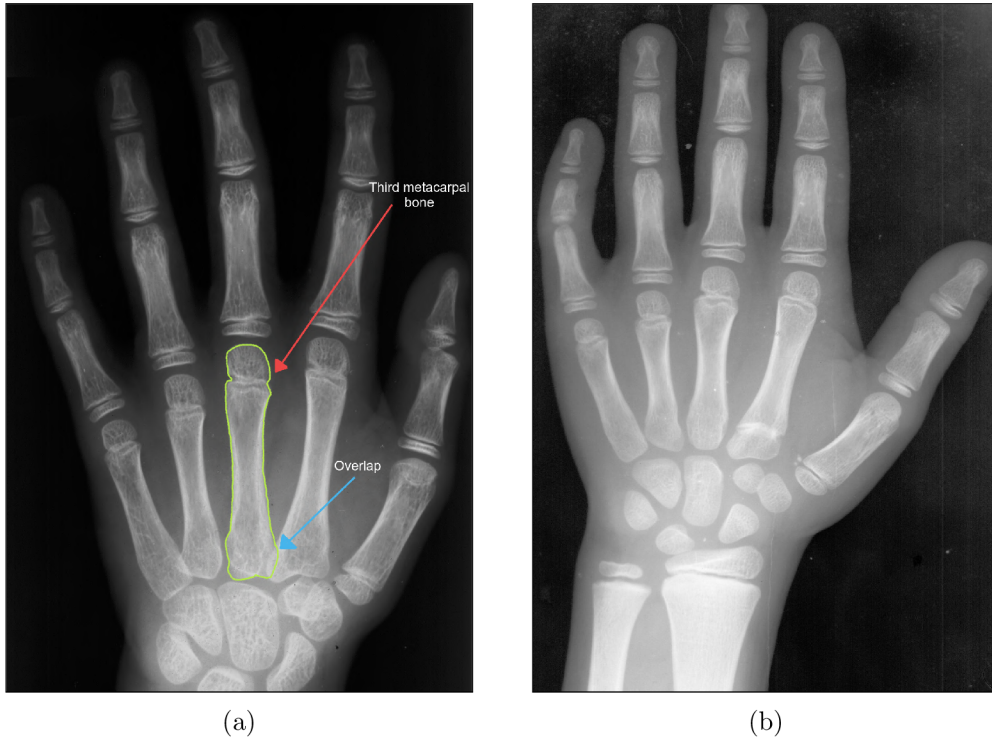


Figure 2.2: **X-ray images comparison.** (a) Hand of the grown up human showing the overlap of third metacarpal bone. (b) X-ray image of child human hand with no overlap.

2.2 Third Metacarpal Bone Structure

Each of the metacarpal bones has basic structure. In general, the bones of metacarpals are made up of three parts. The first is the *base*, which forms the foundation of the bone, then the largest part is the *body* or *shaft* of the bone. Finally, the bone is finished with the so-called *head* of the bone as shown in the picture 2.3b [32]. The head and base of the bone are also medically called *epiphyses*. The body of the bone is professionally called *diaphysis*, which is the most important part of measuring bone width. From the radiological point of view, it is very important to describe the physical structure of the bone. It consists primarily of two specific parts, which are color-coded on the X-ray image. These parts are *compact bones* and *medullary cavity*. The image 2.3a shows in more detail what part of the bone it is. If we work with the X-ray image shown on picture 2.3d, the compact bone has a light color and is located on the edges of the bone. Contrariwise, the medullary cavity is significantly darker. In regard to image processing, the bone can be divided into three separate regions shown on image 2.3c.

2.3 Radiography

Radiography is a method for creating image throughout X-ray pictures and gamma emission to visualize inner body parts. This procedure is mainly used in medicine or anthropology and is usually connected with bone diagnostics [9].

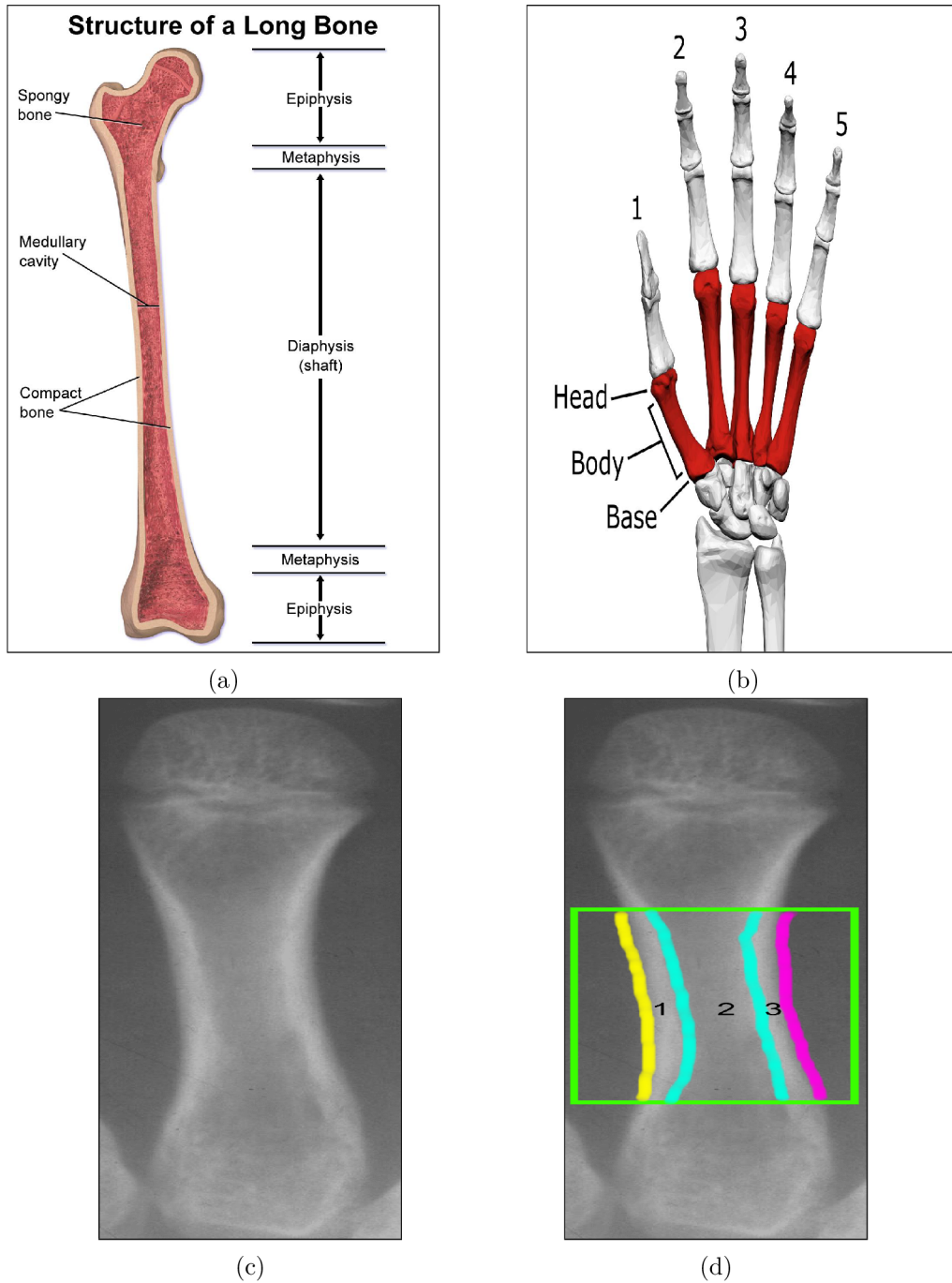


Figure 2.3: **Third metacarpal bone structures.** (a) Medical view of long bone structure [7]. (b) Basic bone structure [5]. (c) X-ray image of third metacarpal bone. (d) X-ray image showing three individual segments.

2.3.1 X-ray Image

X-ray images are produced due to electromagnetic radiation, which is absorbed in the individual parts of the body. X-rays travel through the body and are absorbed in different amounts by tissues possessing different radiological density, having strong impact on the fi-

nal formation of the X-ray image. If the density of the displayed organ is low, its brightness is significantly lower and the object appears darker in the final image. On the other hand, if the displayed tissue's density is bigger, the final image seems brighter and its color resembles white. Roentgen images of the bones and their eventual brightness is influenced for example by the amount of calcium in the bones – the bigger the amount of calcium, the lighter the shade of the displayed bone [39, 28].

2.4 Digital Image

It is possible to imagine a digital image as a “function $f(x,y)$ of two continuous variables x and y ” [8]. In order to be able to process this image digitally, it must be sampled and transformed into a numerical matrix, which in the end represents a three-dimensional image where the third dimension stands for proportion of colors in colored image or the amount of light in grayscale image. This image is then composed of items called pixels. In the most common way the pixels are sorted into the rectangular shape using the pixel coordinate system where the y -axis is vertically flipped. Image 2.4 shows the layout of digital image with better understanding of the coordinate system. The image width represents the actual number of columns, and at the same time the image height is the number of rows. Number of pixels can be defined as $Pixels = M \times N$, where M is the number of rows and N is the number of columns [41].

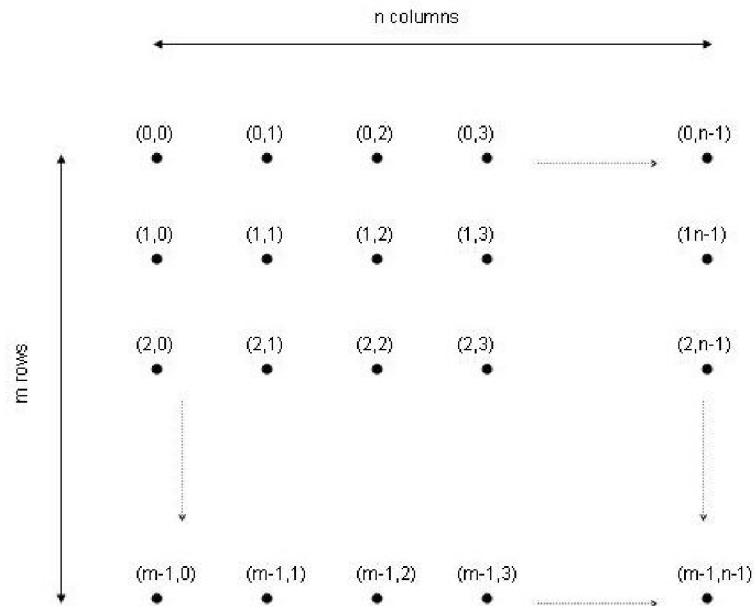


Figure 2.4: **Digital image layout.** The image depicts how the individual pixel coordinates are perceived in the digital images coordinate system [31].

2.5 Human Hand Digital Image Processing

Image processing is very important and rather difficult task always including a number of essential steps. Application and use of this discipline is broad and is encountered on daily basis for instance in modern systems for object recognition in car ride, facial recognition to unlock the phone, in photo and picture editors to improve their quality and last but not least, in analysis of medical images, such as X-ray images to reveal diverse anomalies and disorders [24, 44]. Digital image of medical pictures may originate from various sources, of which every one uses different technique for its performance, namely magnetic resonance imaging or computer tomography. Their area of interest are various body parts and body structures such as soft tissues or bones, amongst which especially the bones are most frequently examined using X-radiation and are also the major subject of this thesis, specifically the processing of digital X-ray images.

2.6 Traditional Image Processing Methods

2.6.1 Grayscale

One of the introductory and fundamental steps of preprocessing of the image is the conversion of the picture from the color channel to typically 256 values of gray where the lowest value is 0(black color) and the highest is 255(white color). Digital image is usually composed of three color channels of which every channel presents one individual color. They mostly represent red, green and blue color(RGB) and every one of these channels contains luminance value to define the brightness of the color. To acquire grayscale image which will only consist of the levels of gray, it is necessary to remove all three of the color channels leaving only the value of the luminance according to formula [47, 1]:

$$Luminance = 0.299 * Red + 0.587 * Green + 0.114 * Blue \quad (2.1)$$

This process is of great importance for improvement of the image segmentation and edge detection, which would not be possible to reveal in colored pictures. It is a preferred approach in image analysis as it enables to omit the excess amount of information in form of color value and by doing so, simplifies computational process. In above mentioned methods, these information do not feature as valuable.

In case of X-ray images, this step is not to be performed in any way due to the fact, that the final X-ray images in digital form are comprised solely of shades of grey color and ignore the color channels from its very essence. This is because the color in X-ray images is non-essential and in the same way it is not useful in the exposure of the disorders and anomalies we use roentgen imagining for.

2.6.2 Noise Reduction

Very important part of preprocessing of the image is smoothing, which is manifested by removal of the noise from the digital image. The problem of a great noise is, that it can cause reduction of visibility of certain inspected parameters in the image, such as the edges of the bones in an X-ray image [43]. Due to this feature, it is necessary to use particular

methods to remove the pre-formed noise, but yet be careful of undersegmentation caused by excessive smoothing. Contrariwise, in case of unrestricted noise it will lead to reverse effect which is oversegmentation.

The removal of the noise may be performed using so called noise reduction filters, whose principle is based mainly in averaging ambient pixel values into one ultimate value. This filter can be pictured as two dimensional matrix with appropriately chosen size, which is usually known under the name *kernel* or *mask* and contains values depending on the type of selected filter, which is then enclosed to the large two dimensional matrix of the image from the left to the right and from the top to the bottom direction. The removal of the noise may be performed using so called noise reduction filters, whose principle is based mainly in averaging ambient pixel values into one ultimate value [45].

There are several types of filters and they all differ in their properties regarding filtration of the image. One of the most popular filters is the so-called Gaussian filter, which is also used in the Canny edge detector method. Application of the correct filter depends on the type of the noise or on individual latter focus of image processing. Testing individual filters and comparing the quality of their results with regard to required properties of subsequent analysis is extremely important. The effect of noise reduction can be seen on images 2.5.

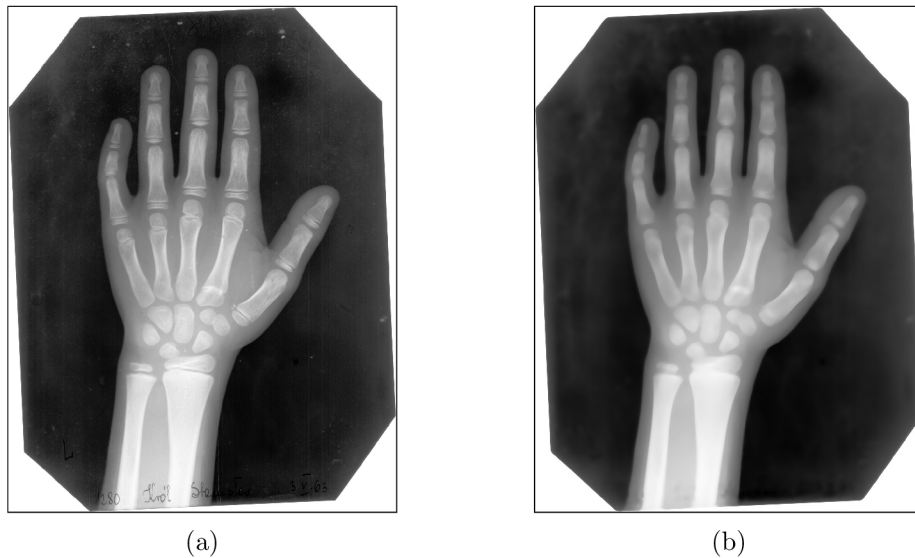


Figure 2.5: **Denoising effect.** (a) Normal X-ray image. (b) Blurred X-ray image.

2.6.3 Thresholding

One of the basic algorithms used for image segmentation is thresholding. The output of this algorithm is an image in binary form, which means it is containing only two colors and that is white with pixel value of 255 and black with value of 0 which can be seen in the pictures 2.7. The basic thresholding function is defined by formula 2.2.

$$g(x, y) = \begin{cases} 255 & \text{if } (x,y) > \mathbf{T} \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

Variable \mathbf{T} is a suitable threshold.

If this classic threshold approach would be used on graphically enhanced image from the X-ray dataset, freed of noise, output shown on image 2.7a is quite unpleasant.

However, the thresholding method may have several other approaches for image segmentation, which are a bit more complicated but are able to segment the image with better accuracy. It can be segmented using the so-called Otsu's thresholding method described in the section 2.6.5 or using the adaptive thresholding described in the section 2.6.4.

2.6.4 Adaptive Thresholding

This method of thresholding is slightly more complicated than conventional thresholding, but it is more applicable. Unlike basic thresholding, which has one threshold value set for the entire image, adaptive thresholding changes the individual thresholds for certain local pixel ranges in the processed image. The result of using the algorithm can be seen in the image 2.7c, which achieves much better results than basic thresholding [33].

2.6.5 Otsu's Thresholding

The most complicated of the mentioned methods is Otsu's thresholding. This algorithm works best with so-called bimodal images, which contain two peaks in their histogram in the grayscale version, such as in the image 2.6a. The algorithm assumes that the image contains two peaks, which represent two classes. These two classes portrait the object and the background. Using these two peaks, it calculates the optimal threshold, which is approximately the mean value between the two peaks according to which the image is further segmented. The mean value should occur between the two peaks in the so-called valley. In the case of an image from the X-ray database, it can be seen that the histogram 2.6b is relatively usable, but not ideal enough to perform good Otsu's thresholding. Nevertheless, the results of segmentation using this algorithm are much better than basic thresholding from section 2.6.3 [29, 33].

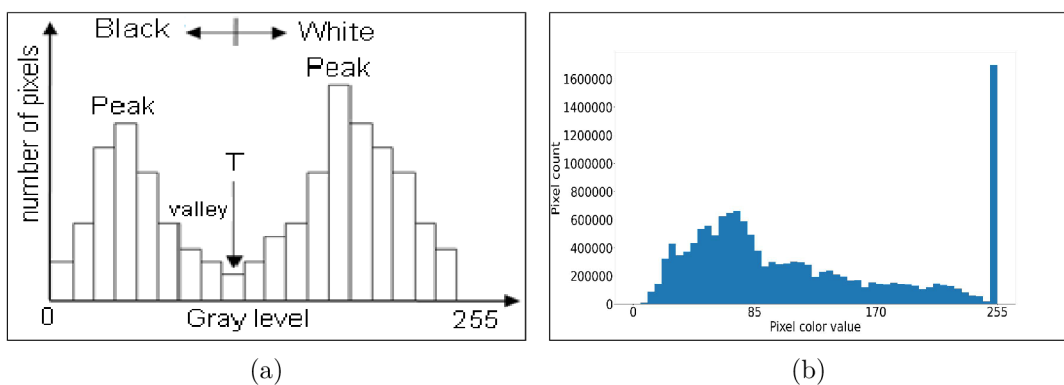


Figure 2.6: **Otsu's images of histogram.** (a) Example of histogram with two peaks for bimodal image [29]. (b) Histogram of image from X-ray database.

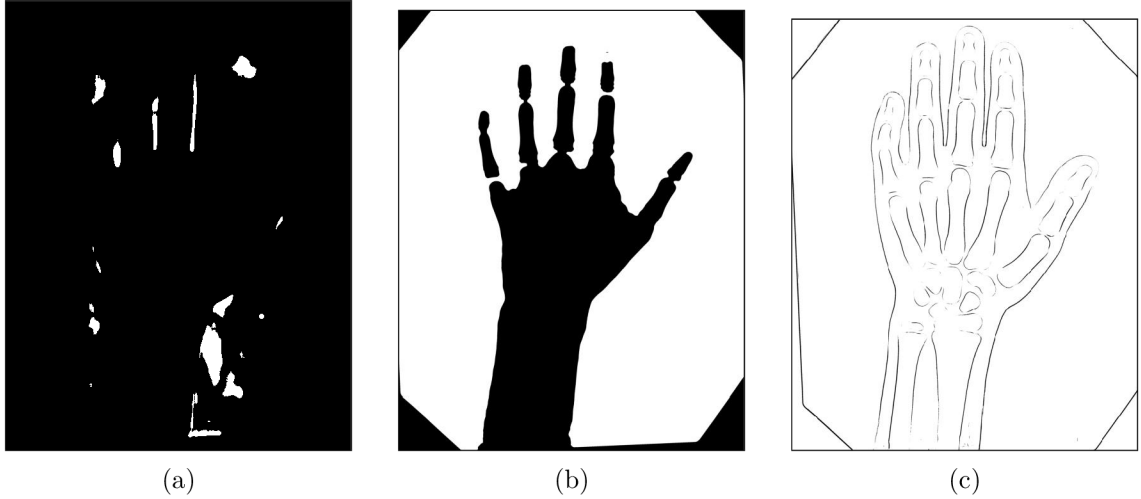


Figure 2.7: **Thresholded pictures.** (a) Classic threshold. (b) Otsu's threshold. (c) Adaptive threshold.

2.6.6 Canny Edge Detection

It is a method that is one of the most popular for edge detection, but it is certainly not suitable for every application. One of the advantages of this detector is that it is able to create edges with a width of one pixel. This edge detector is very sensitive to noise, therefore the image preprocessing is very important. The resulting output of the Canny edge detector of graphically enhanced image shows the figure 2.11 [25]. However, one of the best approaches for application of Canny edge detector is on binary image mask. This mask consists only of two pixel values which are 0 and 255. If we apply this edge detector to the image for binary mask of third metacarpal bone depicted in the image 2.10a, then we can safely consider the output of the detector to be practically error-free, as can be seen in the image 2.10b.

Noise Reduction in Canny

The first step of the Canny edge algorithm is a noise reduction, where a Gaussian filter is used. This filter performs convolution on the processed image to smooth the image and remove unwanted noise from the image. An example of a 5x5 convolution matrix is expressed by the equation 2.3 [42, 35]:

$$M = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \quad (2.3)$$

Finding Intensity Gradient of the Image

After the smoothing effect, the image is then filtered using *Sobel filter* in vertical and horizontal direction of the image matrix. Convolution matrices for obtaining a gradient by the Sobel method can be seen in the equations 2.4 and 2.5 [34]:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.5)$$

This filtration is used to obtain the first derivative in horizontal direction G_x and vertical direction G_y . These derivatives can be represented as two images, from which is possible to obtain an *edge gradient* using the equation 2.6 and its direction for each pixel using the equation 2.7 [35].

$$Edge_Gradient(G) = \sqrt{G_x^2 + G_y^2} \quad (2.6)$$

$$Angle(\theta) = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (2.7)$$

Non-maximum Suppression

Main idea behind the Non-maximum Suppression is to create thinner edges of the detected object. This part of the algorithm works by finding the pixel with the maximum value in an edge.

Pixel A in the image 2.8 is on the edge in vertical direction and the gradient direction is perpendicular to the edge. Pixel B and C are in gradient directions which means that pixel A is checked towards pixel B and C to check if it creates a local maximum. In case that there is created a local maximum the point is then taken to the next part of the algorithm, otherwise, it is suppressed to value of zero [35].

Hysteresis Thresholding

Last important part of this algorithm decides if the detected edge is the real edge or not. For this part, two threshold values are established. First threshold value is the minimal value and the second is the maximal value. If there is any edge, which has the intensity gradient more than maximal threshold value then these potential edges are considered to be sure edges. On the other hand, edges which has the intensity gradient value below the minimal threshold value are considered as sure non-edges and are then discarded from the detected edges. Moreover, if there are edges which has the intensity gradient between

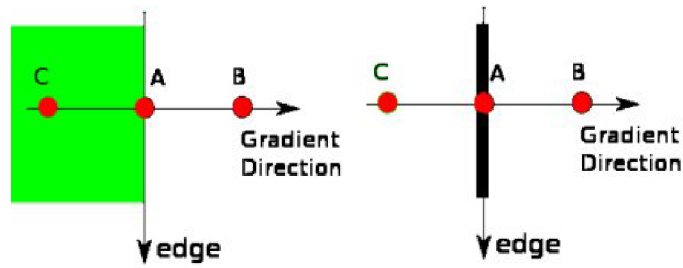


Figure 2.8: **Non-maximum suppression.** Image shows algorithm detection if the pixel *lies* on the edge or not [35].

these two threshold values but at the same time are connected with the sure edges, then these edges are also considered to be the sure edges, otherwise they are considered as sure non-edges.

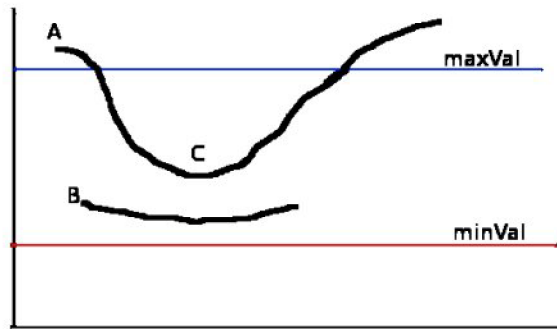


Figure 2.9: **Hysteresis thresholding.** Edge detection example for three different edges [35].

Image 2.9 describes the decision making algorithm for edge detection. If there would be three different edges detected, called edge A, B and C then the thresholding is the key for making decision. As can be seen in the image, the intensity gradient is greater than the maximal threshold value and is then considered as the sure edge. Edge B lies above the minimal threshold value but at the same time is below the maximal threshold value. In that case, if the edge is not connected with any different edge, is then considered as sure non-edge. At last, edge C is the same case as the edge B, but it is connected with the edge A which is the sure edge and thus it is also considered to be the sure edge [35].

2.7 Medical Image Segmentation

Image segmentation in medicine has a wide range of application and is used large-scale for detailed diagnosis to uncover health problems which may not be visible to the naked eye of a layman. Although, at the same time, it is a demanding process, which requires choosing the most suitable method depending on the type of image, its quality and focus on the specific problem, demanding a much detailed analysis.

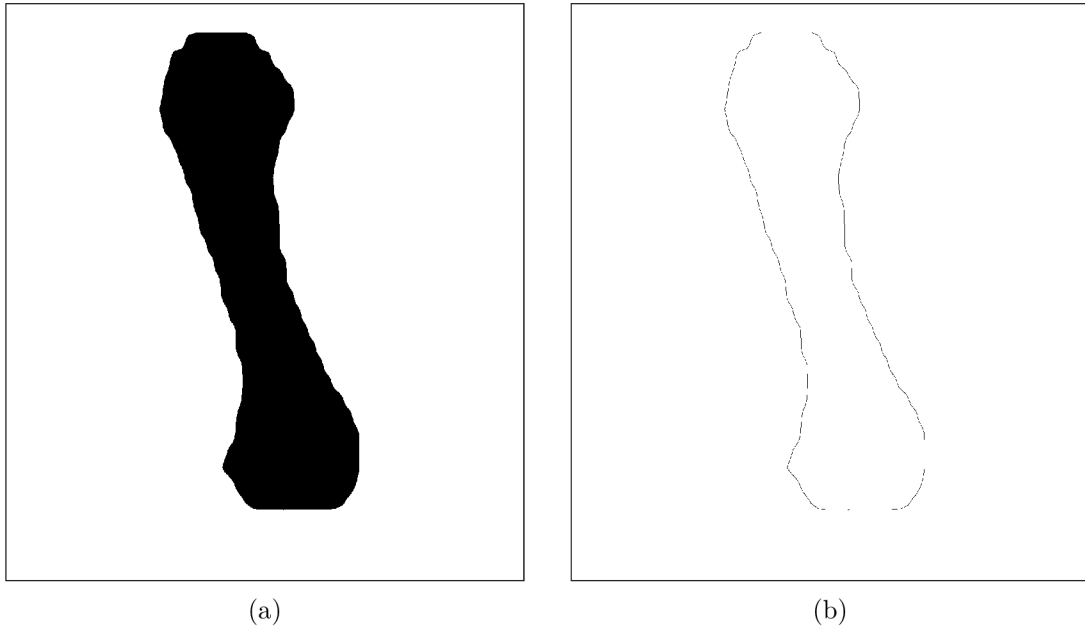


Figure 2.10: **Canny edge algorithm effect for binary image.** (a) Binary image mask. (b) One pixel wide contour obtained using Canny edge detection.

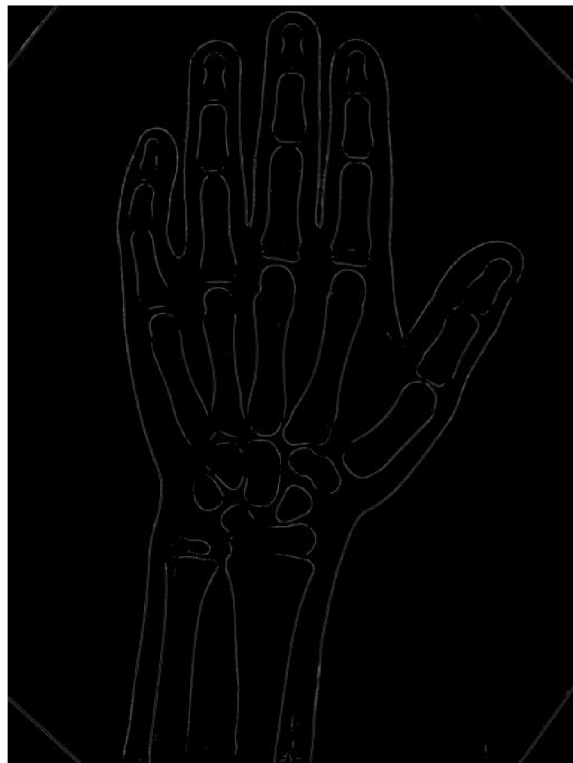


Figure 2.11: **Canny edge detector output.** This image shows canny edge detector output used on noise reduced and adaptive thresholded image.

The very segmentation of the image can be described as a process in which the image is being subdivided into so-called segments or subsections, all of regular or irregular shapes. This process is mostly used to find dividing lines among the objects in the image [46].

2.7.1 Semantic Segmentation

The main goal of semantic segmentation is to classify the same or similar objects in an image under the common class [27, 15]. Such classes can be anything, such as a person, a car, a dog, a cat and so on. However, this object can also be something unusual, such as the metacarpal bones of the human hand. The image 2.12 shows a custom mask represented by matrix value of one for each classified object. One of the biggest shortcomings of semantic segmentation is that it is not able to individually label each detected object separately. In cases like this, it would not be possible to distinguish between different dog breeds or different types of cars. This problem also occurs in the classification of the third metacarpal bone, since semantic segmentation is not able to differentiate between different types of bones. Nevertheless, this problem can be solved using another type of segmentation and that is the instance segmentation. The difference between these types of segmentation is shown in the images 2.13a and 2.13b. The instance segmentation is described more in detail in the section 2.7.2.

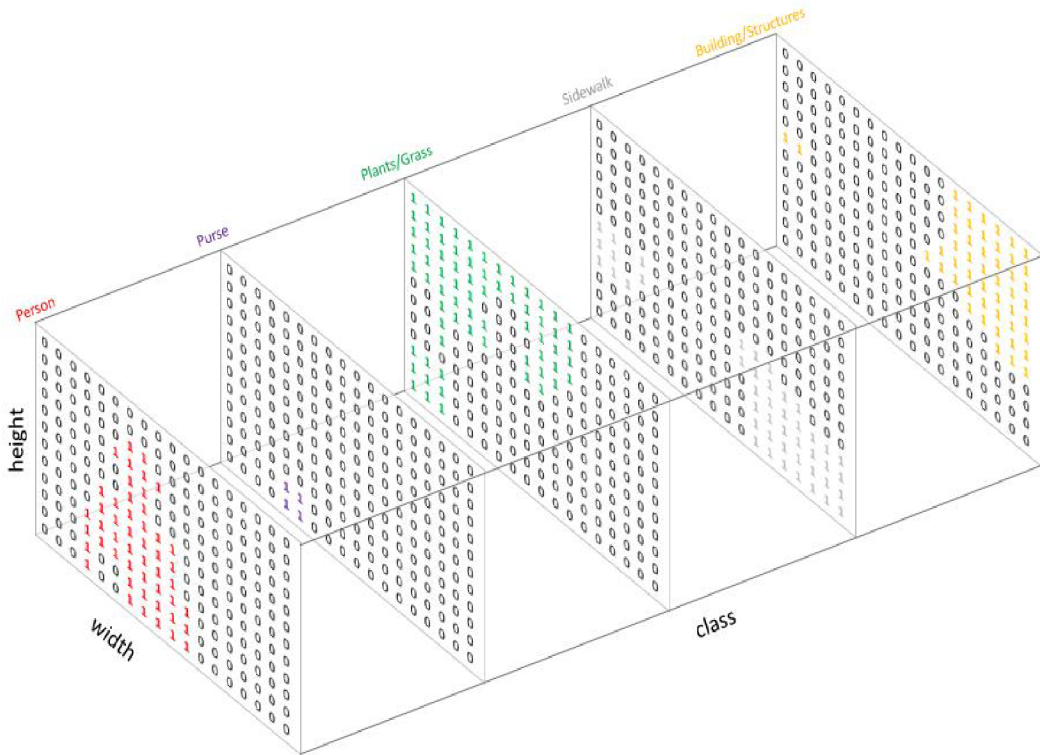


Figure 2.12: **Semantic segmentation classification mask.** The image shows classification of individual objects in image matrix [19].

2.7.2 Instance Segmentation

The instance segmentation has one big advantage over semantic segmentation in case it is necessary to determine more detailed information of detected objects. This segmentation method gives different labels to detected objects in the image of the same class [15]. This method, as well as semantic segmentation, is often used by CNN. In order to be able to use one of the mentioned segmentation methods, it is first necessary to train individual

images using CNN and create a model from them. The training of such a model is more time consuming in the instance segmentation than, for example, in semantic segmentation. If the instance segmentation is compared to the semantic segmentation from the perspective of the detection of the third metacarpal bone, it is possible to assume that the instance segmentation will be more suitable. If it is necessary to distinguish the third metacarpus from the rest of similar-looking bone shapes, it is certainly better if this bone is detected using the method that shows the image 2.13b. Semantic segmentation could be a more suitable method in case where it is necessary to identify whether it is, for example, bone in the hand or some type of soft tissue. As can be seen in the picture 2.13c, the result of one of the algorithms, which will also be described in the section 3.6, is relatively precisely drawn bone mask and therefore it appears as one of the appropriate approaches to segmentation of the third metacarpus.

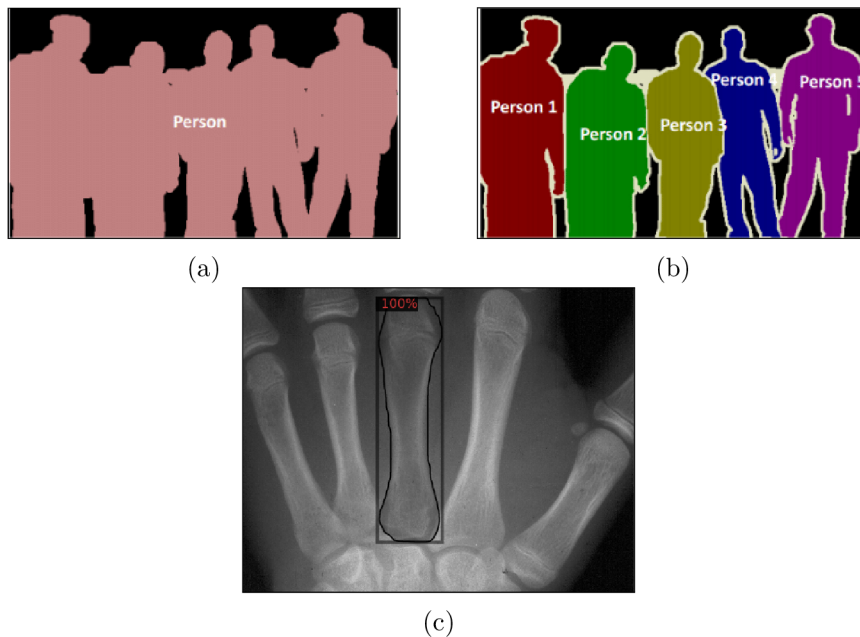


Figure 2.13: **Semantic and instance segmentation comparison.** (a) Semantic segmentation example [49]. (b) Instance segmentation example [49]. (c) Result of instance segmentation on X-ray database image.

2.7.3 Watershed Segmentation

Watershed algorithm, whose essence is derived from natural and basic behavior of natural elements is very important and frequently used in medical image segmentation [18]. The image can be portrayed as topographic surface, which is made of so-called ridges and valleys. If we consider an image in 3D-form, in which the light areas are ridges and dark areas are valleys, we can define a concept of catchment basins. Then, the algorithm segments thus created catchment basins. The problem occurs if the image is too noisy and as a result, oversegmentation appears. This can cause a formation of large amount of regions or so-called segmented areas [38, 11]. Therefore, it is very important to create adequate image enhancement. Image 2.14 shows watershed segmented image with proper image enhancement.



Figure 2.14: **Watershed image segmentation output.** Image shows the result of watershed segmentation on one of the X-ray database image. Many inaccuracies can be observed in this method output.

2.8 Supervised Learning

Quality methods for pattern recognition often utilize training dataset that contain many well annotated images. Training dataset has inputs and correct outputs. These two states allow the model to learn over time. During the learning process the algorithm measures its accuracy through the so-called loss function¹, adjusting until the output of the loss function is minimal number. In this thesis, there is used supervised learning based on manual image annotations of the third metacarpal bone using neural networks algorithm [10, 12]. Supervised learning uses training data on the basis of which a final test model for future inference is then created. In case of X-ray images, inference will be performed on the database of images from the anthropological institute for segmentation and contour detection of the third metacarpus.

2.8.1 Mask Region-based Convolution Neural Network (Mask R-CNN)

According to [16], Mask R-CNN is an extension of Fast R-CNN, which consists of two parts. The first part is the region proposal network (RPN), which features bounding rectangles for mandatory objects. Next part extracts features using RoIPool² from each candidate box and performs classification and bounding-box regression³.

¹Loss function evaluates how good is the model created on training dataset. In case the predictions are not well, the loss function will output the higher number, otherwise the output of loss function will be lower number [2].

²According to [20] Region of Interest Pooling creates the fixed-size feature maps from non-uniform inputs using the operation of max-pooling on the inputs.

³Bounding-box regression is technique to refine or predict bounding boxes in object detection [23].

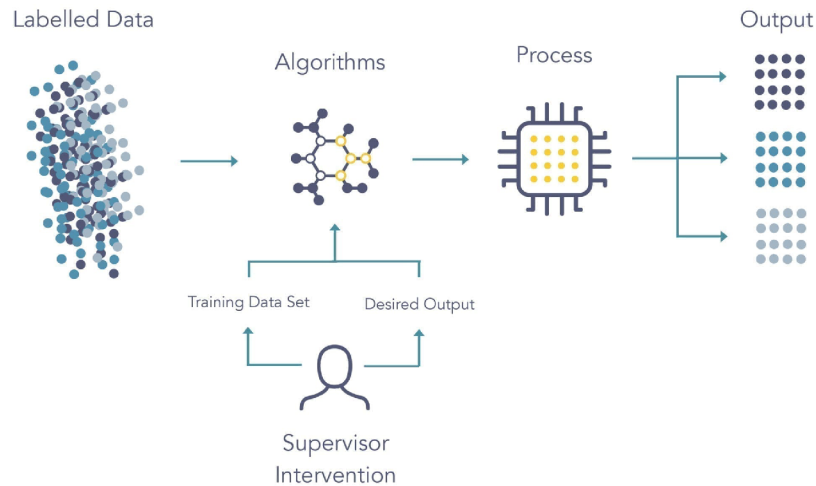


Figure 2.15: **Supervised learning.** Image shows basic idea of supervised learning [14].

2.9 Image Processing Tools

Nowadays, to achieve algorithmic results in image processing, it is possible to use one of many available tools or libraries. They can be applied in gradual decomposition of problems in the image pre-processing and its analysis. Working with the digital image, of course, is not easy and requires many mathematical operations, which are often computationally demanding and algorithmically complicated. In such case, it is convenient to use one of already available tools to work with computer vision.

2.9.1 OpenCV

One of the most famous libraries available to public is OpenCV. The library has a wide range of algorithms, which are optimized at speed and provide high rate of abstraction of the algorithm functionality, and therefore bring simplicity and effectivity together into work itself. This thesis uses algorithms for image processing, segmentation and edge detection.

2.9.2 Detectron2

In addition to libraries and tools that use traditional methods for working with images, such as OpenCV, there are also libraries that can be used, for example, for image processing using neural networks. Detectron2 is a library that offers several ways to perform pattern recognition, such as semantic or instance segmentation. The library contains several ImageNet⁴ pre-trained models that can be used to create a custom model to solve a specific problem. If the framework from the section 2.8.1 is used, it is possible to use baselines based on FPN⁵ (Feature Pyramid Network) in combination with ResNet⁶ (Residual Neu-

⁴<https://imagenet.stanford.edu/>

⁵<https://arxiv.org/abs/1612.03144>

⁶<https://arxiv.org/abs/1512.03385>

ral Network) backbone for mask and box prediction. These models provide the created architecture for CNN, thanks to which it is not necessary to create your own architecture for neural networks, which greatly facilitates the work itself and the creation of a reliable model. When working with the library itself, the use of GPU acceleration is very effective, however, if the computing device does not have a quality graphics card, it is possible to use only the basic computing unit (CPU) for calculations. Detectron2 uses Python syntax and contains a custom abstraction with a number of functions that can be used to create a custom image segmentation model, for example using the instance segmentation method, and perform inferences on test data at the same time. The output after inference is an input image with a predicted mask, such as in the image [2.13c](#) [51].

Chapter 3

Designed Solution and Implementation

At first, this chapter describes an algorithms based on empirical discoveries of the tested methods for image noise reduction, edge detection and usage of traditional segmentation methods for pattern recognition. Created flowchart for this solution can be seen in image 3.1. This chapter also describes designed solution for desired algorithm using CNN which seems to be more usable than the classic segmentation methods. Furthermore, this chapter describes the advanced algorithm extension which is mainly focused on measurement of the bone in more detail than just the basic algorithm. At last, this chapter explains the basic usage of the algorithm including tools which were used for development and the simple directory tree structure of the program.

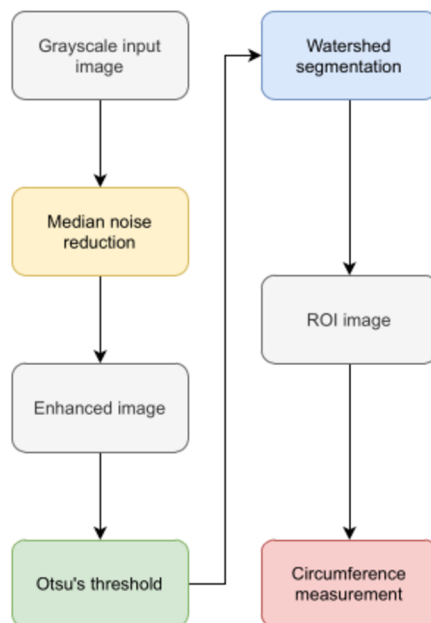


Figure 3.1: **Flowchart.** Image depicts flowchart for pattern recognition using traditional methods.

3.1 Analysis of Medical Images Issues

For a correct design, it is essential to determine and analyze any possible issues that might appear in X-ray image processing beforehand. It is important to realize that despite the resemblance of the images, every individual image is different and displays bones of the hand of people of different age and sex. One of the primary problems which requires attention is so called overlap of the bones, which can be caused for instance by the wrong position of the hand on the scanning device as well as by the age of the concerned patient.

Another issue might be a defective result of the illumination of the bone itself. Any errors may be caused by the position of the hand towards the device responsible for sending out X-rays or by the contents of inorganic substances in the bones, as they result in higher density. Third metacarpal bones are mostly light in their middle part which is the body of the bone. Contrariwise, in the area of the head and the base of the bones they are darker, which shows a negative effect on the edge detection, because it is dependent on clear transition between light and dark shade of grey [48].

Last of the analyzed problems is the very imperfection of the created 2D X-ray images, which causes complications for the measurement phase of the width of the third metacarpal bone. To securely measure the width of the bone in its narrowest area, there has been a mathematical model designed using analytical approach.

3.2 Median Noise Reduction

This smoothing filter was used based on testing of multiple smoothing filters, among which the median filter achieved the best results for X-ray images. It is an effective way which is not computationally difficult and at the same time the edges are preserved for further use.

3.2.1 Proposed Otsu's Threshold

Among all the threshold methods that were presented in the section 2.6.3, the most suitable way was to use Otsu's threshold, thanks to which it was possible to follow well on the following methods in image processing, which very well separates the foreground from the background of the image. By thresholding, an approximated area is obtained where the hand is on the X-ray image, which is sufficient information for watershed segmentation.

3.3 Application of Watershed Segmentation

The principle of algorithm watershed was described in part 2.7.3. This way of segmentation is often used while segmenting medical image and it is very sensitive to noise which can lead to oversegmentation [11, 3]. However, if the image is pre-processed well, the algorithm is able to deal with the problem of overlapping of the bones quite easily and effectively. Overlapping occurs in most of the cases of the images in dataset. The algorithm was relatively effective in separating the individual regions that are in the image, but the results of segmentation are very inaccurate and practically unusable for further use. As can be seen in the picture 2.14 the third metacarpus was separated from the remaining bones, which was

proved mainly due to the good position of the bone, which did not interfere too much with the secondary bones and could also contribute to appropriate illumination of the picture. However, this approach worked only for a few images from the entire database, which causes problems for the robustness of the algorithm, which must work for the statistical majority of images, so it is possible to assess that segmentation using traditional methods is unusable to solve a problem that would be sufficient for anthropological department.

3.4 Canny Edge Detector Applied for Watershed

After using the watershed segmentation method, the edges were detected using Canny edge detector. This method is one of the essential steps after segmentation of the image. However, in this case, the results of watershed segmentation are not as good as necessary for a high quality edge detection. The results of application for Canny edge method can be seen in image 3.2. This edge detection was applied for the segmented hand from the picture 2.14.

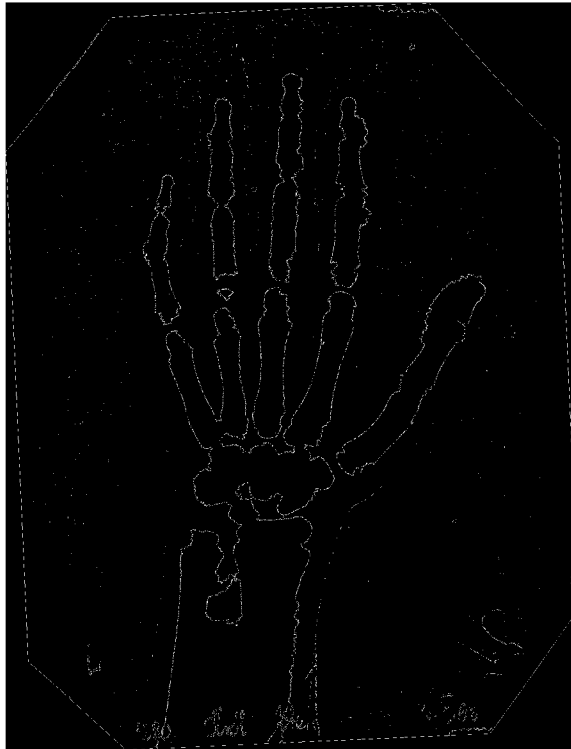


Figure 3.2: **Canny edge application.** Image shows the application of the edge detector after watershed segmentation.

3.5 Evaluation of Tested Methods Issues

To create a correct algorithm design, it was necessary to test algorithms for preprocessing and segmentation of the image and analyze the achieved results. Afterwards, it was possible to draw a conclusion regarding which of the algorithms was the most suitable one. This particular situation leads us to the conclusion that it is not possible to use only

a simple watershed segmentation, as it is not competent to sufficiently deal with the overlap of the bones. It is necessary to use a more complex segmentation algorithm to detect the edges more clearly.

3.6 Design for Pattern Recognition with CNN

Since the use of traditional methods for pattern recognition has not proved to be a reliable method, it is essential to choose a method using convolutional neural networks. By creating a model using the Detectron2 library, the algorithm will predict the mask of the third metacarpal bone. This chapter also interprets actual process of contour detection and calculation of the width of the third metacarpal bone in the area where the bone reaches its most narrow diameter. The algorithm is designed to function properly in an available dataset of X-ray images and to provide uniform results for the statistical majority of the processed images. One of CNN's strong points is no need to use pre-processing methods, such as noise reduction or normalization. Despite the fact that the segmentation of traditional methods failed in the approach from the beginning of the chapter 3, even these methods won't be avoided completely. After obtaining the mask, which is the most important task in CNN usage, it is necessary to display the bone as a binary image, which is very effectively aided by the classic separation of foreground from background, which in this case will not be performed by any of the OpenCV library function but a specially created function. Extraction of a bone contour in the width of one pixel can be performed using the Canny edge detection method, which works error-free for a binary image with removed artifacts. Individual steps of the designed algorithms are displayed in the proposed flowchart 3.3.

3.6.1 Using Mask R-CNN by Detectron2

As mentioned in previous sections, the proposed algorithm for bone mask segmentation uses a specially created convolutional neural network Mask R-CNN, which is built into the library Detectron2, and therefore will be the main focus and starting point in this chapter.

3.6.2 Custom Model Creation

To predict the mask, it is necessary to train the model, which will detect the third metacarpal bone from X-rays. The Detectron2 library is used to train the model, which enables the creation of a custom model. In order to create that, it is necessary to annotate a certain part of X-ray images, specifically the third metacarpus, thanks to which it is possible to determine the so-called ground truth image¹.

3.6.3 Image Conversion to Compatible Format

In order to annotate the database, it is required to perform an image format conversion. As the result of the X-ray image database being in TIF format, and the selected annotation

¹Ground truth image in conjunction with machine learning prediction is an ideal expected image mask. Ideally, the output masks from the predictions should achieve the similar results as ground truth images.

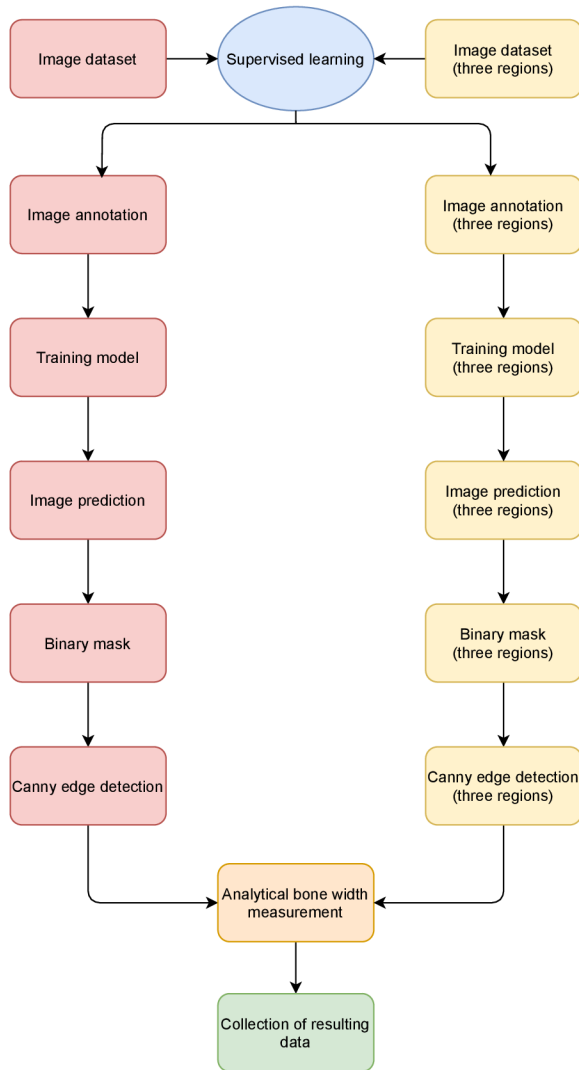


Figure 3.3: **Flowchart.** Image shows flowchart for pattern recognition using CNN.

tool being unable to process images in this format, the images have been converted to JPEG format.

3.6.4 LabelMe Annotation

There are a number of tools for image annotation, and at this point it depends mostly on the tools and libraries that are used in the algorithm. From the point of view of the Detectron2 library, which uses the COCO dataset for object detection introduced in paper [26], to create an object detection model, it is vital to work with files in a special JSON format called JSON COCO. This is a special format that describes how labels and metadata are stored in the image dataset [17]. Images of count 180 from the X-ray images of human hand database were used for the image annotation. After thorough communication with the staff from the anthropological institute, the final annotation was performed by the institute itself due to expertise and accuracy of labeling. X-rays are not always easily recognizable

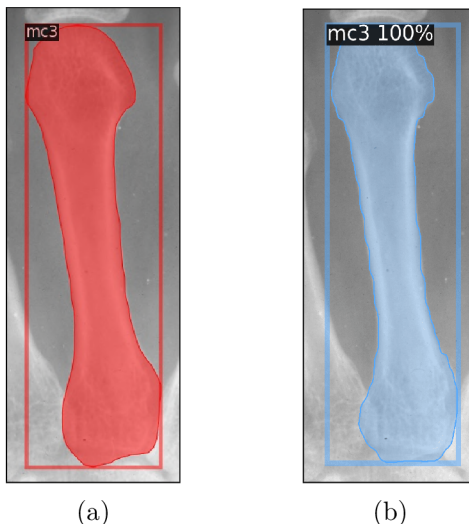


Figure 3.4: **Ground truth comparison.** (a) Ground truth image. (b) Predicted image mask.

and contain many artifacts and bone overlaps, which often only experienced radiologist are familiar with. Situation is no different with metacarpal bones, and an inexperienced person can often annotate incorrect regions or parts of the bone which it actually does not contain [50]. The image 3.5 shows the way of how to annotate images.



Figure 3.5: **Annotation in LabelMe.** Very precise annotation made by MUNI anthropological institute.

Conversion to COCO JSON Format

Annotation tool LabelMe, which creates its own JSON file for each annotated image. After that, the available script called *labelme2coco.py* is used, which facilitates the rest of the annotation work. In order to be able to create a model for object detection, it is necessary to convert all the labeled JSON files into a single COCO dataset annotation JSON file which currently provides the mentioned script taken from the work [52].

3.6.5 Training Model

After a successful conversion, the training process can be started. Training on annotated images from the X-ray image dataset is performed using the Detectron2 library. The training process itself is not a simple task. It requires a lot of testing and experimentation. In order to be able to create a model as accurately as possible, it is necessary to work with several parameters that are required when working with neural networks. In addition, the training process is often very time consuming and depends a lot on the computer resources on which the training is performed. For this reason, the training of the model was performed using Google Colaboratory, which provides quality GPU acceleration [13]. Metrics for model training:

- Configuration file: **mask_rcnn_R_101_FPN_3x.yaml** loading model zoo configuration, using ResNet and FPN (Feature Pyramid Networks) backbone
- Model weights: **mask_rcnn_R_101_FPN_3x.yaml** loading pre-trained weights based on model zoo from Detectron2 library
- Number of iterations: **10000**
- Number of parallel data loading workers: **2**
- Number of images per batch across all machines (depends on number of GPU's): **2** meaning that each GPU will see two images per batch
- Learning rate: **0.00025** is the hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated, it has big impact for resulting model
- Batch size per image: **128** number of samples or images, that will be propagated through the network
- Number of classes: **1** is the number of detected thing classes in one image

During the training and search phase of the most suitable model, many combinations were created to find the most relevant one. The results were empirically researched and manually compared. The output of the training process is a model in PKL format with the name *model_final_10k.pth*.

3.6.6 Loading Image Dataset

The algorithm is adapted to be able to load a database of images from a specific directory, including recursively all subdirectories in the image database. This method of reading images is thus designed because the provided database is divided into two subdirectories. These directories are MALES and FEMALES. X-rays were taken from many people over a period of several years and were also differentiated by gender. After loading the image from the dataset, inference, contour detection and analytical measurement of bone width are performed on each of these images.

3.6.7 Creating Inference on Image Dataset

Creating predictions on images is relatively easy thanks to Detectron2. The principle is based on the use of function `predict(...)`, which returns several types of predictions such as bounding box coordinates, percentage accuracy of the detected object and mask of the predicted object.

3.6.8 Image Binary Mask Creation

To obtain the contour of the bone, a binary mask of the object must be created. It is one of the most important steps, which greatly facilitates the subsequent steps in the algorithm. Nevertheless, the creation of a binary image of the detected bone does not present a complication from the programming perspective. The code 3.1 converts the mask of the detected object, which is part of the detector from the Detectron2 library and is composed of the values *True* or *False*. From these values it creates a binary mask with a range of 0-255 values, where each value of *True* is assigned the value 255 as white color and vice versa the value of *False* is assigned a black color with the value 0.

```
def binaryMask(mask):
    height = mask.shape[0]
    width = mask.shape[1]

    # black image matrix with zero values
    binary_mask = np.zeros((height, width))
    binary_mask += mask

    # binarization of image to values 255 or 0
    for y in range(height):
        for x in range(width):
            # obtaining value of pixel, which can be 0 or 1 (False/True)
            pixel = binary_mask.item(y, x)
            if pixel:
                binary_mask[y,x] = 255
            else:
                binary_mask[y,x] = 0
    return binary_mask
```

Listing 3.1: **Python code function.** Function for binary mask creation.

The result of the binary mask created in this way can be seen in the image 2.10a. The mentioned function processes each mask of the image separately, which means that in a case that the prediction of object detection has found several bones of the third metacarpus, then each such bone will be processed separately. In most cases, the database contains an X-ray image of the left hand, but there are some cases where both hands were scanned, which can be seen in the image 3.6.

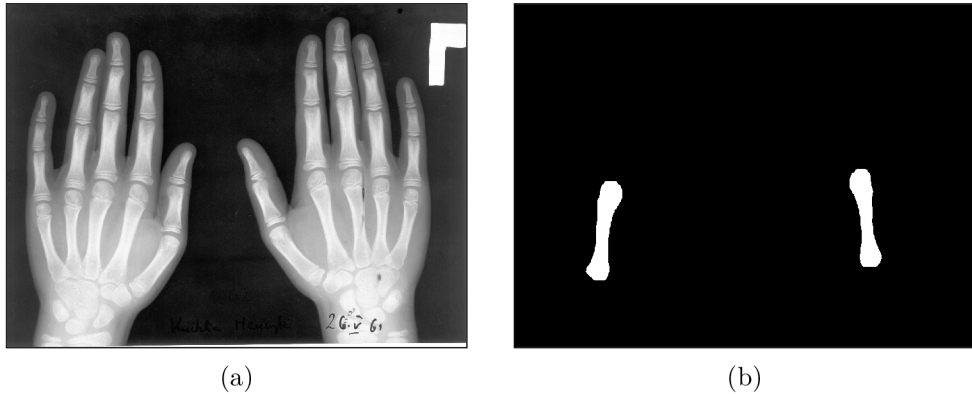


Figure 3.6: **X-ray image with both hands and the output mask for the third metacarpus.** (a) X-ray image. (b) Binary mask.

3.6.9 Contour Creation

The traditional methods which were described in the beginning of the chapter 3 as unsuitable for segmentation and contour detection, are nevertheless used in this algorithm, in which they have proven to be effective and reliable. The following steps would not be possible without obtaining a binary mask of metacarpal bones using instance segmentation and Mask R-CNN, as this is a critical part of the algorithm. The `Canny(...)` method from the OpenCV library is used to extract the contour of a bone whose width is perfectly one pixel and at the same time the bone is fully drawn, which meets all the requirements and works uniform for every image of a binary mask. The output from this part of the algorithm is in the image 2.10b.

3.6.10 Contour Coordinates Export

Obtaining coordinates from the bone contour closes the first part of the algorithm, whose task is to detect the contour. In addition to the outputs of the algorithm in the form of images, which are stored in individual directories, the outputs are also text files. This step was designed following the preferences of people from the anthropological institute, whose recommendations were outputs in TPS format. In TPS format, each image has its own file with the name of the bone, which contains the contour coordinates of each detected bone in the format $[x,y]$.

3.6.11 Analytical Bone Width Measurement

The second part of the algorithm is the analytical measurement of the width of the third metacarpal bone. This part of the program does not use any of the methods for pattern recognition and computer vision. Analytically, the measurement approach works primarily with individual pixel coordinates by comparing their mutual distances. The task is not simple and is relatively computationally demanding, as the images from the database are in high resolution and iterations through the image matrix are quite time-consuming.

Bone Center Line Approximation

Approximation of the center line of the bone is the most important, and also the most difficult task for finding the distance. Since it is possible to compare the distances of each pixel from the left side of the bone with each pixel on the right side, it is essential to divide the bone into two parts. The most relevant way to perform this division is by using the center line approximation. Since the bone is not a symmetrical shape, it is necessary to find a center line in an irregular polygon. This requires the connection of two points. These points are then used to divide bone into two as symmetrical halves as possible.

In the section 3.6.7, it is mentioned that in addition to the bone mask of Detectron2, the coordinates of the predicted bounding rectangle are also obtained. These are two points, the coordinates of which are the point P1[topleft_x, topleft_y] and the point P2[bottomright_x, bottomright_y]. These coordinates are in the algorithm necessarily for the approximation of the center line. The function middleX(...), the code of which can be seen in 3.2, is used for the approximation of the middle X coordinate by arithmetic mean. This X coordinate is calculated for both top and bottom of the bone. It uses the coordinates of points from the bounding rectangle. After calculating the middle X coordinates, the bone can be easily divided into two parts using the center line. The bone image is shown in the 3.7 image.

```
def middleX(top_or_bottom_y, bottomright_x, topleft_x, offset, img):
    height = img.shape[0]
    width = img.shape[1]
    x_coords = list()

    for i in range(offset):
        for y in range(height):
            for x in range(width):
                if x >= topleft_x and x <= bottomright_x:
                    # detecting pixels from the top left or bottom right y
                    # coordinate on the way up
                    if y == top_or_bottom_y - i:
                        if img[y,x] == 255: # edge found
                            x_coords.append(x)
                    # detecting pixels from the top left or bottom right y
                    # coordinate on the way down
                    if y == top_or_bottom_y + i + 1:
                        if img[y,x] == 255: # edge found
                            x_coords.append(x)

    # mean average from all x coordinates found
    avg_x = np.mean(x_coords)

    # middle x coordinate
    return round(avg_x)
```

Listing 3.2: **Python code function.** Function of middle x coordinate approximation for center line.

Since bone structure is often irregular and the prediction of the location of the bone vertices is not possible, it is necessary to collect X coordinates from the top or bottom of the bone. The top or bottom of the bone may contain various local maxima or minima, and if the diameter X of the coordinates were formed only in the range of one pixel of the Y coordinates, this would often lead to false and inaccurate conclusions of the approximation of the middle X coordinate. Therefore, an offset is used in the algorithm, which is set to the empirically tested magic number 40. This number means that all X coordinates from the range of 40 pixels of Y coordinates are taken, and they are used to calculate the mean average. In such laborious and computationally demanding way, which requires a lot of iterations, it is possible to approximate with sufficient precision where the middle X coordinate will be located.

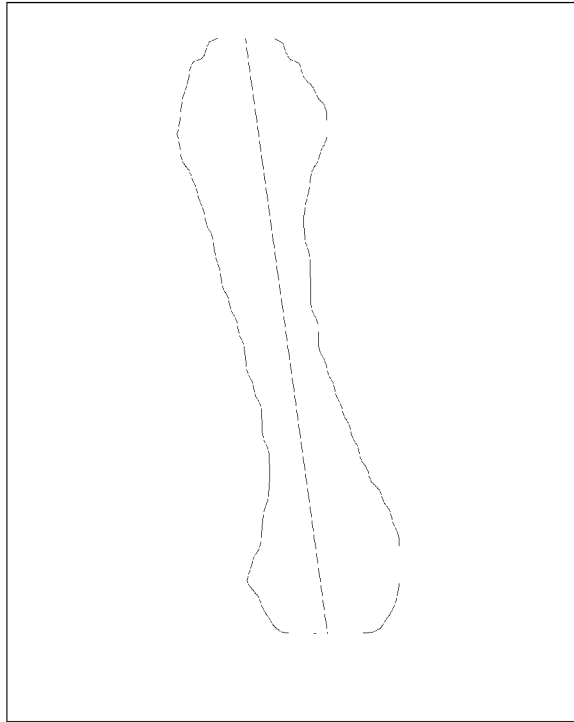


Figure 3.7: **Bone with center line.** Image shows approximated center line using two points with computed middle X coordinate.

Bone Division

To calculate the distance between two points of the left part of the bone and the right part of the bone, it must be determined which of the points will belong to the right and left side. To do this, there is used the obtained center line of the bone. In order to decide whether a given point can be found on the right or left side, the task is to determine whether a given point lies below or above the given abscissa, which forms the center line of the bone. According to [30], it is possible to use a cross product as a decision threshold. Therefore,

in the case of this algorithm, the following must apply [40]:

$$\begin{aligned}
 A &= (x_1, y_1) \\
 B &= (x_2, y_2) \\
 P &= (x, y) \\
 d &= (x - x_1)(y_2 - y_1) - (y - y_1)(x_2 - x_1)
 \end{aligned}
 \tag{3.1}$$

In this equation must hold that if $d < 0$, then the point P lies above the line or on the left side, otherwise the point lies on the opposite side. Points A and B create the center line.

```

def divide(topleft_x,topleft_y,bottomright_x,bottomright_y,offset,img):
    left_side = list()
    right_side = list()

    # lambda expression for point location
    # determination returns True or False
    cross_product_isabove = lambda p3,p1,p2: np.cross(p3-p1, p2-p1) < 0

    # center line points
    p1 = np.array([topleft_x,topleft_y])
    p2 = np.array([bottomright_x,bottomright_y])

    # image shape
    height = img.shape[0]
    width = img.shape[1]

    for y in range(height):
        for x in range(width):
            if y >= topleft_y + offset and y <= bottomright_y - offset:
                if img[y,x] == 255:
                    p3 = np.array([x,y])
                    if cross_product_isabove(p3, p1, p2):
                        left_side.append([y,x])
                    else:
                        right_side.append([y,x])

    return left_side, right_side

```

Listing 3.3: **Python code function.** Function divides bone into two halves using center line.

It is important to realize that in order to obtain relevant points for both left and right side, the distance between two points, whether on the head or the base of the bone, must not be taken into account. Since the goal is to find the distance in the narrowest part of the bone from the anthropological point of view, and thus from the point of view of bone structure, it will certainly not be located on the head or base of the bone but will be located in the body of the bone. However, from the computer algorithm point of view, this fact should be included in the calculation. Otherwise, the shortest distance will be found between two points, where the center line divides the bone into two halves and

not in the middle of the bone. For this reason, there is an empirically chosen magic offset in the algorithm, which arranges that the distance starts to be calculated only after omitting a certain number of pixels in the y-axis, and this applies to head and base as well. This magic constant is chosen to be 150, which works well for all the bones of the third metacarpus. In the image 3.8, it is possible to see the division of the bone into right and left side. In images 3.8a and 3.8b, the offset of 150 pixels is neglected for clarity.

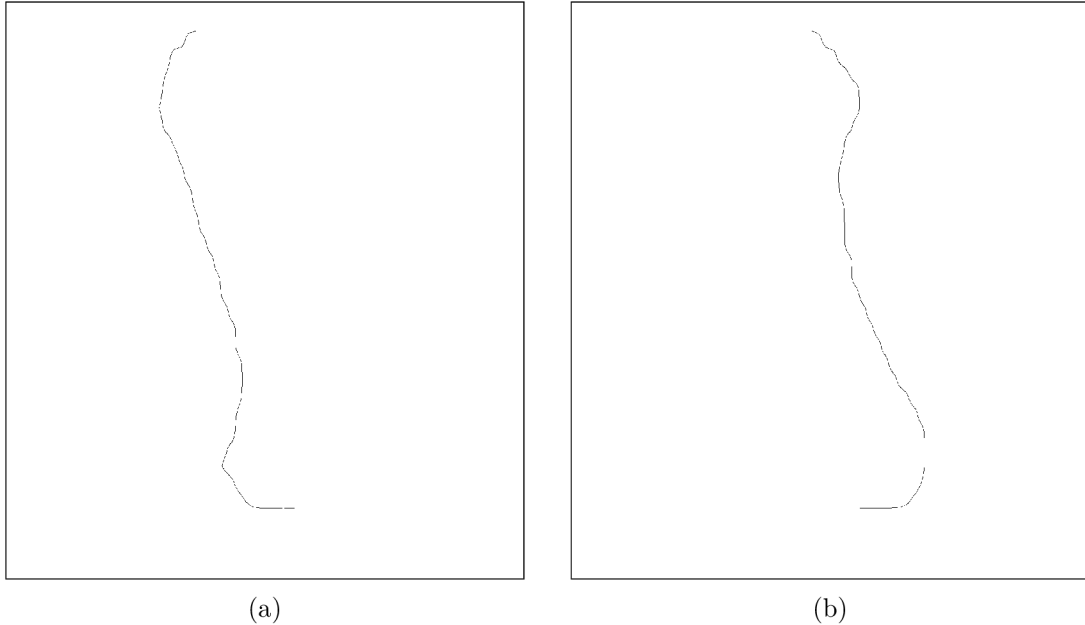


Figure 3.8: **Images show the bone division according to center line.** (a) Left side of the bone. (b) Right side of the bone.

Finding Minimal Distance Using Pythagoras Theorem

The final measurement of the distance between two points A and B is no longer a difficult task. Sequentially, each distance of the left side of the bone will be compared with the right side of the bone. To find this distance, Pythagoras theorem is used, where the following holds:

$$\begin{aligned}
 A &= (x_1, y_1) \\
 B &= (x_2, y_2) \\
 \text{distance} &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}
 \end{aligned}
 \tag{3.2}$$

If the new distance is less than the distance measured for the previous two points, it becomes a new minimum distance. The image 3.9 shows the final result of the analytical approach, which includes the center line, minimal length that was found, and the bone contour itself, which does not include the head and base of the bone.

Pixels Conversion to Millimeters

Since the measured distances work unequivocally with pixel coordinates, such output would be irrelevant for real research from anthropological point of view. That is why it is necessary

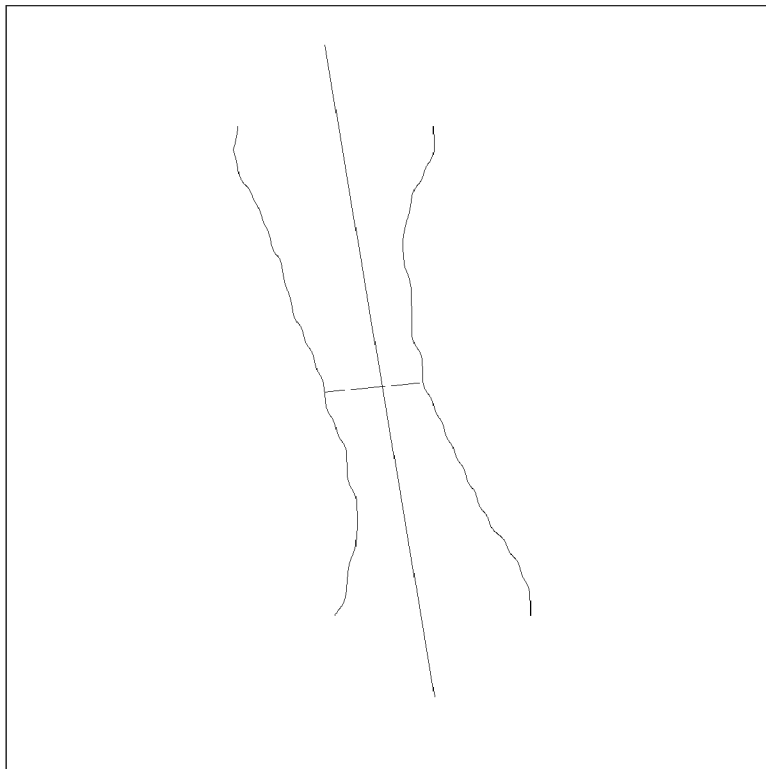


Figure 3.9: **Analytical approach pictured.** Image shows the final minimal length found in the third metacarpal bone.

to convert such values into actual units of distance. The basic equation is used for the conversion, but it must be slightly modified. Since the images were made by an anthropological institute, and were rescanned from the material images, it was necessary to find out what DPI(Dots per inch) digital images disposed. The properties of the images themselves provided an indication of 150, which is a relatively standard DPI for higher resolution digital images. However, when the scanning of images was provided, zoom was used and applied to 300%. This makes the total DPI value a triple and that equals 450. This information was verified and confirmed during communication with anthropology department. The final equation is therefore as follows:

$$d = \frac{p * c}{DPI} \quad (3.3)$$

Variable d is the final distance in millimeters, p is the measured distance in pixels, c is a constant with a value of 25.4 which is the distance of one inch in millimeters and finally DPI is a constant indicating the number of points at a distance of one inch set to 450.

3.6.12 Bone Circumference Export

The results of the measured distances are exported to a file in CSV format, where the resulting distance in millimeters is given for each image. If the algorithm detected more than one metacarpal bone in the X-ray, these bones are incrementally labeled from the number one.

3.6.13 Collection of Resulting Data

Successfully achieved measurements must be collected in order to enable the interpretation of the algorithm output in the most comfortable way and also to allow more detailed analysis and future work with the collected data for Masaryk University staff, for whom the program output is valuable and created in accordance with their research. At the same time, it is necessary to consider that the measurement results are primarily intended for people who are not so familiar with all kinds of file formats. Therefore, two types of files are used - TPS and CSV.

In addition to text files with measurements or contour coordinates, the program also saves image outputs during all steps of the algorithm. These outputs include the mask of the detected bone, the binary mask, the contour acquired using the Canny edge detector and the resulting image in which a line is drawn in the area of the bone where the shortest distance was measured, as can be seen in the image 3.10.

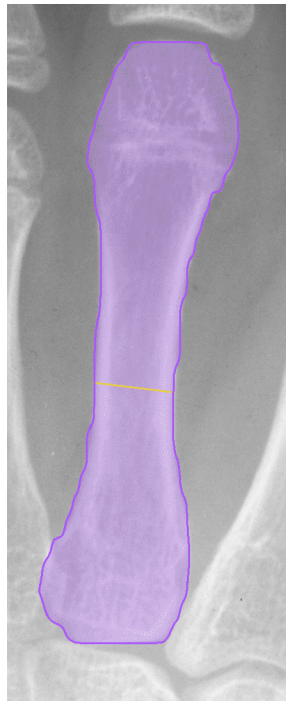


Figure 3.10: **Minimal length displayed in picture.** Yellow line in the image that shows the minimal length found.

Data Text Interpretation

Based upon communication with the team from Masaryk University, the most suitable way for them to interpret the text data of the obtained coordinates is to use TPS format, which they commonly use in their research. The CSV format will be used to export the measured distances for each image. This format is also one of the agreed forms of output with the anthropological institute.

3.7 Advanced Algorithm Extension

This section is an addition of the section 3.6, which describes the extension of the basic algorithm. During communication with the anthropological institute, the ultimate conclusion was made, saying that the algorithm could be improved, and thus the possibilities of examining metacarpal bones from X-ray images would be extended. The section 2.2, which thoroughly describes the structure of metacarpal bone, states that long bones are formed by compact bones, which form the edges of the bone, and the so-called medullary cavity that fills the inner bone.

The purpose of this extension is to measure the smallest distance in each of these parts separately, but it must still be in the part where the total distance of the bone width is the shortest. The approach in this extension will be very similar, despite a certain difference from the basic part of the algorithm. The Detectron2 library is again used to distinguish these three special parts to create a model in the same way as the model for detecting the mask of the third metacarpal bone. The image 3.11 shows the annotation of the region of interest. The created model consists of three individual classes for each part of the bone separately.

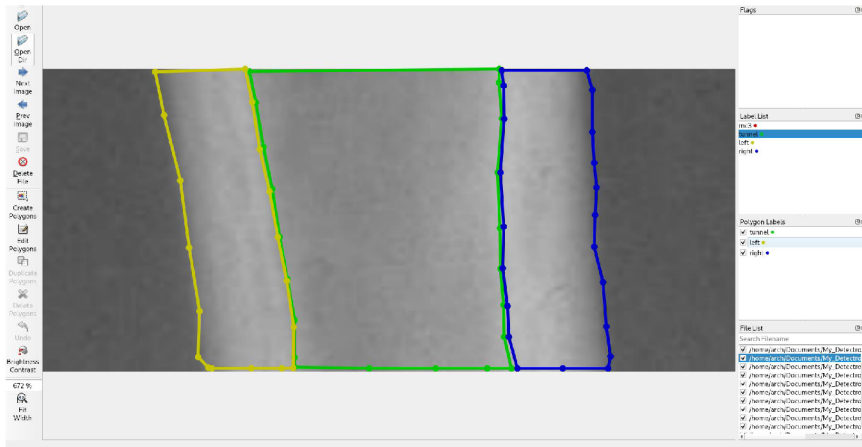


Figure 3.11: **Annotation of region of interest in LabelMe.** Image shows annotation process for three separate regions of metacarpal bone.

3.7.1 Model Creation for ROI

The training process was performed similarly as in the section 3.6.5. The chosen metrics for the resulting model are slightly different, but still very similar, thanks to which the functional model was created. Metrics for model training:

- Configuration file: **mask_rcnn_R_101_FPN_3x.yaml**
- Model weights: **mask_rcnn_R_101_FPN_3x.yaml**
- Number of iterations: **10000**
- Number of parallel data loading workers: **4**

- Number of images per batch across all machines: **2**
- Learning rate: **0.00025**
- Batch size per image: **128**
- Number of classes: **3**

3.7.2 Bone Measurement in Three Separate Regions

The created model is similarly utilized, and using the Detectron2 library, a prediction is applied, which creates a mask for each of the regions, which is shown in the image 3.12a. From these masks, a binary mask is then created using the function 3.1 for each region individually, as can be seen in the image 3.12c. The figure 3.12b is for illustrating the binary mask of all three regions in one image. Similarly, an edge detector is applied to these masks using the Canny edge method, where the final output is shown in the figure 3.12d.

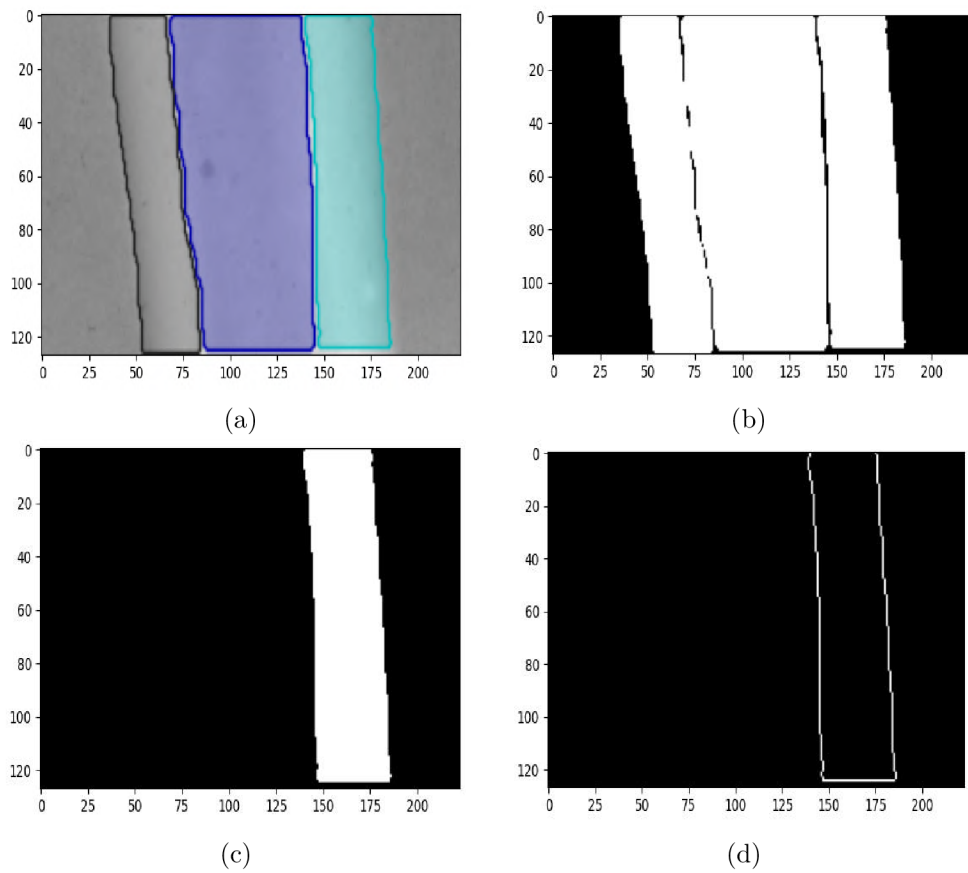


Figure 3.12: Images depicting outputs from extended algorithm for ROI (a) Detectron2 image mask prediction. (b) Binary mask of predicted image. (c) Binary mask for one of the three segments. (d) Canny edge detector output for applied for one of the three segments.

Analytical Measurement Approach

Measuring bone distance is identical to measuring width for the whole metacarpal bone. The difference is that each of the detected regions is measured individually, which means that the sum of these three distances may not be equivalent to the total distance. If the two points that determine the width of the bone do not lie at the same height, which means that the coordinates $y_1 \neq y_2$, then the range of coordinates in which the width will be measured in the individual regions is the difference of these coordinates. Therefore, it must hold from $range = |y_1 - y_2|$. The image 3.13 shows an example where the individual distances of the three separate regions are found in different locations and do not form one continuous distance as in the measurement of the total bone width. Each measured distance is color-coded by an automatically drawn line on the image.

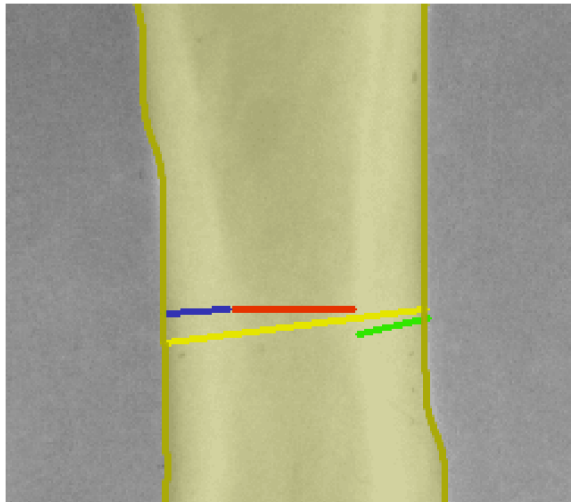


Figure 3.13: **Measurement comparison.** Image depicts difference in measurement of three separate regions and the full width of the bone.

3.8 Usage of the Algorithm

The algorithm is executable as a terminal application, which therefore does not have any GUI elements. This can act as a certain disadvantage for the usability of the application itself. However, from the usability point of view and the perspective of resulting data collection, the GUI elements are not very indispensable and could serve more as a possible improvement to a potential extension of the application for greater user comfort.

The program was developed by Python version 3.8.5 in the Linux environment maintained under the Archlinux distribution. The primary libraries used in this work are Detectron2 and OpenCV version 4.5.1, which are used to work with images. Most of the source files in the `Modules` directory contain a PYX extension, which is an extension of the PY extension, and is used to recompile source files using C language for hardware optimization, which is discussed in more detail in the section 4.4.1.

3.8.1 Directory Structure

The program has a directory structure, which consists of several scripts and other files. The directory structure of the program is as follows:

```
X-ray_Detector
├── Annotations_JSON
│   ├── Metacarpals_Train.json
│   └── ROI_Train.json
├── doc
│   ├── Doxyfile
│   ├── manual.pdf
│   ├── latex
│   └── html
├── README.md
├── Models
│   ├── metacarpal_model.pth
│   └── ROI_model.pth
├── ROI_Train
├── META_Train
├── Utils
│   └── labelme2coco.py
├── clear.sh
├── install.sh
├── run.py
├── src
│   ├── main.py
│   ├── setup.py
│   └── Modules
│       ├── canny.py
│       ├── config.py
│       ├── contour_coordinates.pyx
│       ├── distance.pyx
│       ├── mask.pyx
│       ├── roi.py
│       ├── test.pyx
│       ├── three_regions.pyx
│       └── train.py
```

The `Annotations_JSON` directory contains appropriately validated JSON COCO annotations that were made using the `labelme2coco.py` script. The folder `doc` contains PDF manual documentation for the entire source code, generated using Doxygen tool. Folder also has documentation configuration file `Doxyfile`. This folder also contains `html` and `latex` subdirectories with autogenerated documentation sources like TEX and HTML files. Program has also brief installation and functionality manual described in file `README.md`. The `Models` directory is made up of trained models using the Detectron2 library, which is used for prediction in the X-ray image database. The `ROI_Train` and `META_Train` directories consist of annotated images to create a model with the appropriate JSON files generated by the LabelMe annotation application. The `Utils` directory contains the downloaded

`labelme2coco.py` script, the meaning of which has already been described. The `clear.sh` script deletes the generated shared objects (.so) of the library, which are created by recompiling modules with the PYX extension. The `install.sh` script downloads all the necessary libraries and dependencies that are necessary to run the program. The `src` directory contains all the important source files of the program. The `setup.py` script is used to generate shared object libraries and C files for precompilation. The script `main.py` is the main module, the execution of which starts the algorithm. The `Modules` directory contains the scripts that are the core of the program. Finally, there is an executable script `run.py`, which starts the program.

3.8.2 Program Running Options

The program can be run relatively easily and it is possible to use several options for easier manipulation. These options can be combined between each other. If there are not used any of the optional flags, then the program processes dataset whose name is *hardcoded* in the source code. The overview of use is as follows:

- python **run.py**
- python **run.py -n** name of the image from where the program should start
- python **run.py -i** input image dataset directory path
- python **run.py -o** program output directory path

Chapter 4

Testing on Given Database

This chapter describes the measurement outputs and focuses mainly on algorithm deficiencies and incorrect outputs. It also statistically compares the success of the algorithm on the dataset, the speed of image processing and possible hardware optimization, which will allow faster processing of the database.

4.1 Used Database

The used database, provided by Masaryk University from the Department of Anthropology, follows on from the Wrocław study [21]. The *Material and methods* section in this study further describes the origin of the data, and what number of samples of men and women were examined in this work. The provided database contains a total of 4862 X-ray images, where the images of women's hands are 2329 and the male hands represent the number of 2533. The X-ray images were made mainly for the left hand, but there are cases where both hands were taken on one image.

4.2 Measurement of the Bone Full Width

The following table 4.1 shows the measurement success, including mask detection for the whole width of the third metacarpal bone. If the mask detection is not correct, the bone width measurement will not be correct either.

The table consists of the following parameters, which are expressed by abbreviations. *G* means gender, *FD* is faulty detection and vice versa *CD* is correct detection.

G	FD	FD(%)	CD	CD(%)
Males	0	0	2506	100
Females	2	0.08343	2395	99.91657
Both	2	0.0408	4901	99.9592

Table 4.1: Full width measurement comparison.

4.3 Measurement of the Three Separate Bone Regions

As in the previous part, this section will describe the accuracy of the measurement, which this time will concern the extension of the algorithm with three separate regions of the bone.

The 4.2 table shows the success of detection and measurement and contains abbreviations that are meaningfully identical to the previous section. The table contains one additional parameter, which describes the total measurement deviation if the bone was measured using two points across the width compared to three separate regions. Abbreviation *WD* stands for width deviation. This deviation was calculated as follows:

$$y = \sum_{n=1}^k x - (a + b + c) \quad (4.1)$$

$$d = \frac{y}{k} \quad (4.2)$$

Variable d is the total deviation, a , b and c are the individual distances of three separate regions, i.e. left compact, medullary cavity and right compact. The variable x is the measured distance of a bone for its entire width. The k variable is the number of detected bones. The resulting value of y is the total sum of the difference in distance between the total bone width and the sum of the distances of the three regions. The resulting deviation is finally calculated by dividing the resulting sum and the total number of bones detected.

G	FD	FD(%)	CD	CD(%)	WD(mm)
Males	10	0.399	2496	99.601	0.1736
Females	57	2.3779	2340	97.6221	0.1850
Both	67	1.3665	4836	98.6335	0.1791

Table 4.2: The table of three regions measurement comparison.

4.4 Issues of the Created Algorithm

Like many algorithms, this one is not error-free, and there are certainly a number of optimizations for processing speed, mask detection accuracy, and measurement accuracy. One of the major shortcomings of this algorithm is the mask prediction. The problem that caused the failure of traditional methods in image segmentation was the bone overlap. For instance, in the image 4.1b can be seen that the algorithm can very reasonably predict the mask of overlapping objects, which is the merit of the library Detectron2, which works with the method of instance segmentation. Nevertheless, the predicted mask is often imperfect and fails to accurately outline the boundaries of the bone itself. The image 4.1a shows how many parts of the mask are predicted to be inaccurate. The problem usually arises around the head of the bone, where this segmentation fails the most. Another inaccuracy in this mask is its *waviness*, which is clearly visible in the image 4.1a on the right side or at the top of the bone. The way to eliminate such problem is not entirely clear, but there is a high probability that an improvement could be accomplished by a new model, which

would be created using a large amount of annotated data. The most accurate model was created using training annotated data, the amount of which was 180, which is probably not enough to create a reliable model. Since the Department of Anthropology was notified of such an inaccuracy, which is also reflected in the resulting file of coordinates, their initiative was shown through the annotation of images, because the annotation of professional material requires a lot of time and effort. After the final communication, however, the conclusion was made, that the resulting contour is usable and will most likely find its application as stated by the Institute of Anthropology, such as the calculation of angles between the axes of bone and alike.

The second significant, although statistically negligible problem is the inaccurate prediction of all three regions while measuring individual bone widths. This problem arises again in the part of using the Detectron2 library, and it will probably be mainly related to the created model and the number of annotated images that were used to create this model (165 images). However, the images were made only after the creation of the basic part of the algorithm, which was able to measure the width of the whole bone. Using this width, it was then possible to create an ROI, which automated the cutting out of the important part of the bone. As can be seen in the figure 4.1b, the bone width for the left compact and the medullary cavity was detected successfully, and thus the measurements could be performed on these two segments. As for the right compact, the mask was not detected, and thus it was not possible to measure this segment. Statistically, these problems occur in percentage, which have been described in more detail in the table 4.2.

The program therefore has two marginal problems, where the need to annotate a larger amount of data should be a critical point, and at the same time it should be the primary step in the future to improve this work.

4.4.1 C Language Optimization

There are several different approaches on optimizing speed of the algorithm, specifically written in Python. This algorithm uses the Cython [4] library, which can be used to define, for example, the types of variables, as it is common in compiled languages. Cython generates shared object libraries and practically compiles Python into C. This optimization is chosen in this work mainly due to slow iterations in Python, where it is necessary to iterate several times through the image matrix.

4.5 Computation Time per Image

Usage of optimization in C language is possible to speed up the computation time by 40 seconds per one processed image which presents a significant difference. The measured time for processing was approximated and therefore is not always the precise number as stated in the table 4.3.

Time per image without optimization	Time per image with optimization
60s per image	20s per image

Table 4.3: Time comparison of one image processing without and with optimization.

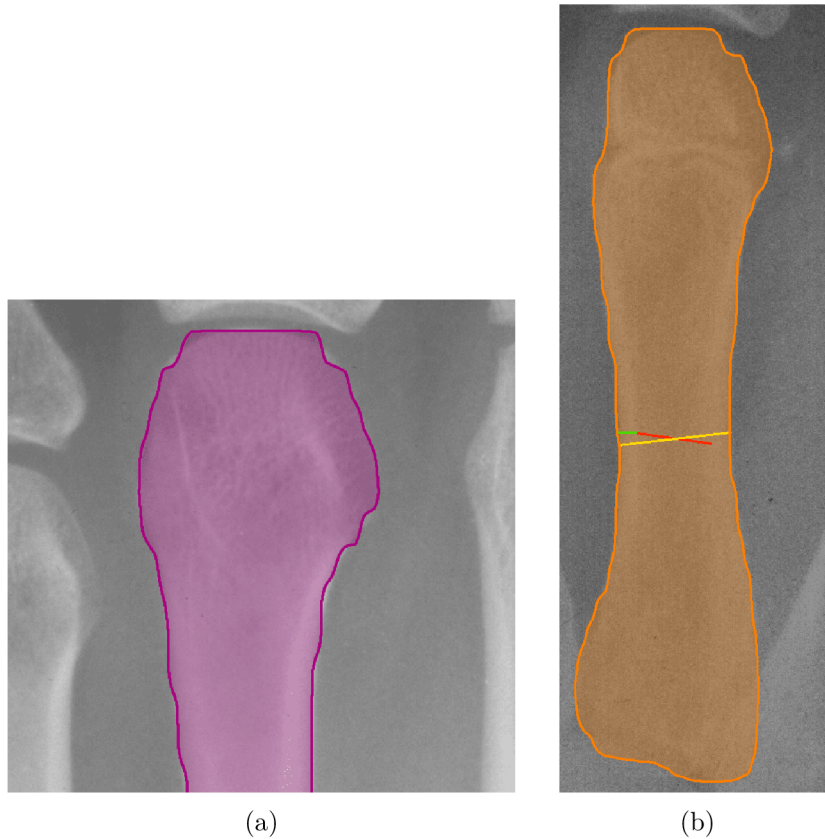


Figure 4.1: **Algorithm issues images.** (a) Detectron2 image mask prediction. (b) Binary mask of predicted image.

4.6 Hardware Resources of Testing Device

The overall processing of the database, besides creating models by the training process, took place on a machine with the following parameters:

- CPU: Intel® Core™ i5-8250U CPU @ 1.60GHz × 8
- GPU: Mesa Intel® UHD Graphics 620 (KBL GT2) but no GPU utilization could be used due to incompatibility with Detectron2 library (Detectron2 is constructed using PyTorch¹ that only supports CUDA Nvidia)
- RAM: 11.5 GiB

4.7 Comparison with Manual Measurements

A sample of manual measurements performed by the Institute of Anthropology in selected 39 figures was used to evaluate the accuracy of measurements. From this sample, the average deviation between manual and automated measurements was calculated. The resulting

¹<https://pytorch.org/>

deviation is on average **0.363572** millimeters, where the measured distances by automated method are mostly smaller than manual measurements, which can be caused by the imprecision of the detected contour. This may cause occasional inaccuracy in drawing the edges of the bone. The resulting measurement is included in the appendix [A](#) in table [A.1](#).

The table [A.2](#) compares the results of manual measurements with automated ones, where automated measurements represent the total sum of distances for measured three separate parts of bone, and thus left compact, right compact and meduallery cavity. The total value of the deviation in this case is **0.520193** millimeters.

Chapter 5

Conclusion

The role of this work could be divided into two large parts. The first task was to create an algorithm that will be able to detect the contour of the third metacarpal bone of the X-ray images available from the database provided by the anthropological institute. The second part, which the algorithm is designed to solve, is to measure the width of this detected bone in its narrowest diameter. Although a variety of types of images are not uniform, it was also necessary to find a solution that would be applicable to different-quality X-ray images containing a scanned hand.

For the introduction, it was necessary to get acquainted with the structure of metacarpal bone itself, including more specific terms such as compact of the bone, epiphysis, medullary cavity etc. Moreover, it was necessary to examine a certain sample of images in detail and identify the most significant problem for detecting bone contour. The marginal problem was the major overlap of metacarpal bones with each other, due to which it is often very difficult for the naked eye to determine exactly where the bone is located. On account of this problem, it was necessary to examine the current state of the art for pattern recognition and object detection in the computer vision department. This has seen several approaches and methods, of which object detection using Mask R-CNN for instance segmentation is the most appropriate and accurate. The Detectron2 library was used for this detection, with the help of which it is possible to use convolutional neural networks for various types of image segmentation. Using this method, it was possible to predict the mask of the third metacarpal bone, which is a key moment for the development of this work. Based on this mask, it was possible to create the resulting contour of the bone using traditional methods and obtain its coordinates for expert analysis of the staff of Masaryk University.

The second part, which deals with the analytical measurement of the bone width, was implemented practically without the use of any higher abstraction, which is provided by any of the available libraries in the Python language. Classic iterations are used through an array of images, which consist of pixels, and an individual comparison of the distance between these pixels.

The algorithm itself, in addition to the basic distance measurement for the whole width of the metacarpal bone, is extended by a functionality that allows measuring the bone width for three individual parts of the bone. These parts are the left and right compact bone, which are characterized by a light color in X-ray images. The third part is the medullary cavity, which forms the inside of the bone and its color is mostly darker. Detection of these

three parts is re-mediated using the Detectron2 library and the instance segmentation method is used.

The created algorithm will help the Department of Anthropology to measure metacarpal bones automatically, thanks to which they will be able to analyze the change of the bone width during the growth of individuals and investigate genetic anomalies, or to detect various adverse growth factors. Another useful element may be that they will be able to avoid manual measurements, which in most cases happens only for the length of the bone and also speeds up the observation process in some way.

Like many working algorithms, in this case it would be also possible to improve the functionality itself in some way. One of the fundamental steps for improvement would be to increase the accuracy of the predicted mask and thus to improve the resulting contour, which is often not completely precise and does not represent the exact shape of the bone. This step would also improve the measurement results, which are highly dependent on the detection of the mask. It would also be possible to improve the analytical approach for measuring three bone segments, where it would be appropriate for the resulting distance for each segment to form one imaginary solid line, as in the case of measuring the total bone width. Last of the possible improvement and extension for this algorithm would be the application for all the metacarpal bones in the hand. This extension was also considered to be quite useful, but at the same time the realization is not easy and requires a lot of time mainly due to time-consuming manual image annotations. This issue should be taken into consideration for the future work.

Bibliography

- [1] AHN, S. H. *Luminance*. 2006. Available at:
<https://www.songho.ca/dsp/luminance/luminance.html>.
- [2] ALGORITHMIA. *Introduction to Loss Functions*. Mar 2021. [Online; accessed 1-April-2021]. Available at:
<https://algorithmia.com/blog/introduction-to-loss-functions>.
- [3] AREECKAL, A. S., SAM, M. and DAVID, S. S. Computerized radiogrammetry of third metacarpal using watershed and active appearance model. In: IEEE, ed. *2018 IEEE International Conference on Industrial Technology (ICIT)*. 2018, p. 1490–1495. ISBN 978-1-5090-5949-2. Available at: [doi:10.1109/ICIT.2018.8352401](https://doi.org/10.1109/ICIT.2018.8352401).
- [4] BEHNEL, S., BRADSHAW, R., CITRO, C., DALCIN, L., SELJEBOTN, D. S. et al. Cython: The best of both worlds. *Computing in Science & Engineering*. 1st ed. Mar 2011, vol. 13, no. 2, p. 31–39.
- [5] COMMONS, W. *File:Metacarpal bones (left hand) 01 palmar view with label.png* — *Wikimedia Commons, the free media repository*. 2020. [Online; accessed 14-April-2021]. Available at: [https://commons.wikimedia.org/w/index.php?title=File:Metacarpal_bones_\(left_hand\)_01_palmar_view_with_label.png&oldid=457110514](https://commons.wikimedia.org/w/index.php?title=File:Metacarpal_bones_(left_hand)_01_palmar_view_with_label.png&oldid=457110514).
- [6] COMMONS, W. *File:Scheme human hand bones-en.svg* — *Wikimedia Commons, the free media repository*. 2020. [Online; accessed 22-December-2020]. Available at: https://commons.wikimedia.org/w/index.php?title=File:Scheme_human_hand_bones-en.svg&oldid=456537967.
- [7] COMMONS, W. *File:Structure of a Long Bone.png* — *Wikimedia Commons, the free media repository*. 2020. [Online; accessed 14-April-2021]. Available at: https://commons.wikimedia.org/w/index.php?title=File:Structure_of_a_Long_Bone.png&oldid=505625447.
- [8] DA SILVA, E. A. and MENDONÇA, G. V. 4 - Digital Image Processing. In: CHEN, W.-K., ed. *The Electrical Engineering Handbook*. Burlington: Academic Press, 2005, p. 891 – 910. DOI: <https://doi.org/10.1016/B978-012170960-0/50064-5>. ISBN 978-0-12-170960-0. Available at:
<https://www.sciencedirect.com/science/article/pii/B9780121709600500645>.
- [9] DEVICES, C. for and HEALTH, R. *Radiography: FDA* [FDA]. Apr. 28, 2020. [Online]. Available at: <https://www.fda.gov/radiation-emitting-products/medical-x-ray-imaging/radiography>.

- [10] EDUCATION, I. C. *What is Supervised Learning?* Aug 2020. [Online; accessed 10-April-2021]. Available at: <https://www.ibm.com/cloud/learn/supervised-learning>.
- [11] FENG, D. Segmentation of Bone Structures in X-ray Images. 1st ed. Jul 2006, no. 1, p. 1–66. Available at: <https://www.comp.nus.edu.sg/~leowwk/thesis/dingfeng-proposal.pdf>.
- [12] GHADERI, A. and ATHITSOS, V. Selective Unsupervised Feature Learning with Convolutional Neural Network (S-CNN). *CoRR*. 1st ed. 2016, abs/1606.02210, no. 1. Available at: <http://arxiv.org/abs/1606.02210>.
- [13] GOOGLE. *Colaboratory*. 2014. [Online; accessed 17-April-2021]. Available at: <https://research.google.com/colaboratory/faq.html>.
- [14] GRIEVE, G. *Machine learning (ML) is expanding the possibilities*. Nov 2020. [Online; accessed 5-April-2021]. Available at: <https://www.logpoint.com/sv/blogg-sv/explained-siemply-machine-learning/>.
- [15] HAFIZ, A. M. and BHAT, G. M. A survey on instance segmentation: state of the art. *International Journal of Multimedia Information Retrieval*. 1st ed. Springer Science and Business Media LLC. Jul 2020, vol. 9, no. 3, p. 171–189. DOI: 10.1007/s13735-020-00195-x. ISSN 2192-662X. Available at: <http://dx.doi.org/10.1007/s13735-020-00195-x>.
- [16] HE, K., GKIOXARI, G., DOLLÁR, P. and GIRSHICK, R. B. Mask R-CNN. *CoRR*. 3rd ed.th ed. 2017, abs/1703.06870, no. 1. Available at: <http://arxiv.org/abs/1703.06870>.
- [17] HOFESMANN, E. *How to work with object detection datasets in COCO format* [Towards Data Science]. Mar. 2021. [Online]. [Online; accessed 16-April-2021]. Available at: <https://towardsdatascience.com/how-to-work-with-object-detection-datasets-in-coco-format-9bf4fb5848a4>.
- [18] JIA XIN, C. and SEN, L. A medical image segmentation method based on watershed transform. In: IEEE, ed. *The Fifth International Conference on Computer and Information Technology (CIT'05)*. 2005, p. 634–638. DOI: 10.1109/CIT.2005.16. ISBN 0-7695-2432-X.
- [19] JORDAN, J. *An overview of semantic image segmentation*. Nov. 2020. [Online]. [Online; accessed 14-April-2021]. Available at: <https://www.jeremyjordan.me/semantic-segmentation/>.
- [20] K, S. *Region of Interest Pooling*. Towards Data Science, Apr 2021. Available at: <https://towardsdatascience.com/region-of-interest-pooling-f7c637f409af>.
- [21] KOZIEL, S. and MALINA, R. Variation in relative fat distribution associated with maturational timing: The Wrocław Growth study. *Annals of human biology*. 6th ed. november 2005, vol. 32, no. 1, p. 691–701. DOI: 10.1080/03014460500268531.
- [22] LAURENCIKAS, E., SÄVENDAHL, L. and JORULF, H. Metacarpophalangeal Pattern Profile Analysis: Useful Diagnostic Tool for Differentiating between Dyschondrosteosis, Turner Syndrome, and Hypochondroplasia. *Acta Radiologica*. 1st

- ed. Taylor & Francis. 2006, vol. 47, no. 5, p. 518–524. DOI: 10.1080/02841850600635939. Available at: <https://www.tandfonline.com/doi/abs/10.1080/02841850600635939>.
- [23] LEE, S., KWAK, S. and CHO, M. Universal Bounding Box Regression and Its Applications. *CoRR*. 1st ed. 2019, abs/1904.06805, no. 1. Available at: <http://arxiv.org/abs/1904.06805>.
- [24] LI, G., SONG, H., WANG, S. and KONG, J. Application of Image Processing and Three-Dimensional Data Reconstruction Algorithm Based on Traffic Video in Vehicle Component Detection. *Mathematical Problems in Engineering*. 1st ed. Nov 2017, vol. 2017, no. 1, p. 1–16. DOI: 10.1155/2017/4145625.
- [25] LIANG, J. *Canny Edge Detector*. Accessed: 20 April 2021. Available at: <http://justin-liang.com/tutorials/canny/>.
- [26] LIN, T.-Y., MAIRE, M., BELONGIE, S., BOURDEV, L., GIRSHICK, R. et al. *Microsoft COCO: Common Objects in Context*. 2015. [Online; accessed 18-April-2021].
- [27] LIU, X., DENG, Z. and YANG, Y. Recent progress in semantic image segmentation. *Artificial Intelligence Review*. 1st ed. 2018, vol. 52, no. 2, p. 1089–1106. DOI: 10.1007/s10462-018-9641-3.
- [28] MEDICINE, G. school of. *Factors Affecting Appearance*. Available at: https://www.dartmouth.edu/~anatomy/HAE/Radiology_Intro/rad/rad2/rad2a.html.
- [29] MOALLEM, P. and RAZMJOOY, N. Optimal Threshold Computing in Automatic Image Thresholding using Adaptive Particle Swarm Optimization. *Journal of Applied Research and Technology*. 1st ed. october 2012, vol. 703, no. 5. DOI: 10.22201/icat.16656423.2012.10.5.361.
- [30] NGO, X. *Python - Find whether a point is above or below a line*. Apr 2019. [Online; accessed 21-April-2021]. Available at: <https://openwritings.net/pg/python-find-whether-point-above-or-below-line>.
- [31] NOOBED. „*FUnderstanding Coordinate System*“. Available at: http://noobed.com/nb_coord_system.htm.
- [32] OLIVIER, M. *Metacarpal bones: Radiology Reference Article*. [Online; accessed 14-April-2021]. Available at: <https://radiopaedia.org/articles/metacarpal-bones-1>.
- [33] OPENCV. *Image Thresholding*. 2013. [Online; accessed 14-April-2021]. Available at: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html.
- [34] OPENCV. *Canny Edge Detector*. Dec. 2019. [Online; accessed 24-April-2021]. Available at: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html?ref=driverlayer.com/web.
- [35] OPENCV. *Canny Edge Detection*. Apr 2021. [Online; accessed 24-April-2021]. Available at: https://docs.opencv.org/master/da/d22/tutorial_py_canny.html.

- [36] PHYSIOPEdia. *Effects of Ageing on Bone* — *Physiopedia*,. 2019. [Online; accessed 23-December-2020]. Available at: https://www.physio-pedia.com/index.php?title=Effects_of_Ageing_on_Bone&oldid=226026.
- [37] PHYSIOPEdia. *Effects of Ageing on Hand Function* — *Physiopedia*,. 2020. [Online; accessed 22-December-2020]. Available at: https://www.physio-pedia.com/index.php?title=Effects_of_Ageing_on_Hand_Function&oldid=256694.
- [38] PREIM, B. and BOTHA, C. Chapter 4 - Image Analysis for Medical Visualization. In: PREIM, B. and BOTHA, C., ed. *Visual Computing for Medicine (Second Edition)*. Second Editionth ed. Boston: Morgan Kaufmann, 2014, p. 111 – 175. DOI: <https://doi.org/10.1016/B978-0-12-415873-3.00004-3>. ISBN 978-0-12-415873-3. Available at: <http://www.sciencedirect.com/science/article/pii/B9780124158733000043>.
- [39] SCIENCE, L. T. and 28, J. *Radiography & X-ray Imaging*. Jan 2020. Available at: <https://letstalkscience.ca/educational-resources/backgrounders/radiography-x-ray-imaging>.
- [40] SHARD. *Calculate on which side of a straight line is a given point located?* [Mathematics Stack Exchange]. 2020. Version: 2020-06-12, Online; accessed 21-April-2021. Available at: <https://math.stackexchange.com/q/274728>.
- [41] SHEETS, K. *1. What is a digital Image? - Learn ImageJ*. 2013. Available at: <https://sites.google.com/site/learnimagej/image-processing/what-is-a-digital-image>.
- [42] SHIPITKO, O. and GRIGORYEV, A. *Gaussian filtering for FPGA based image processing with High-Level Synthesis tools*. June 2018.
- [43] SPRAWLS, P. *INTRODUCTION AND OVERVIEW*. Available at: <http://www.sprawns.org/ppmi2/NOISE/>.
- [44] SRIDHAR, B. ApplicationsOf Digital Image Processing In Real Time World. *International journal of scientific & technology research*. 1st ed. Dec 2019, vol. 8, no. 1, p. 3354–3357. ISSN 2277-8616.
- [45] SWAIN, A. *Noise filtering in Digital Image Processing*. Jul. 2020. [Online]. Available at: <https://medium.com/image-vision/noise-filtering-in-digital-image-processing-d12b5266847c>.
- [46] TAN, Y. Chapter 11 - Applications. In: TAN, Y., ed. *Gpu-Based Parallel Implementation of Swarm Intelligence Algorithms*. Morgan Kaufmann, 2016, p. 167 – 177. DOI: <https://doi.org/10.1016/B978-0-12-809362-7.50011-X>. ISBN 978-0-12-809362-7. Available at: <http://www.sciencedirect.com/science/article/pii/B978012809362750011X>.
- [47] TECHOPEDIA. *What is Grayscale? - Definition from Techopedia*. Accessed: 23 December 2020. Available at: <https://www.techopedia.com/definition/7468/grayscale>.

- [48] THESKELETALSYSTEM. *Metacarpals: Definition, Location, Anatomy, Function, Diagram*. Apr 2019. Available at: <https://www.theskeletalsystem.net/metacarpal-bones>.
- [49] VARATHARASAN, V., SHIN, H.-S., TSOURDOS, A. and COLOSIMO, N. *Improving Learning Effectiveness For Object Detection and Classification in Cluttered Backgrounds*. February 2020.
- [50] WADA, K. *Labelme: Image Polygonal Annotation with Python*. 2016. [Online; accessed 20-April-2021]. Available at: <https://github.com/wkentaro/labelme>.
- [51] WU, Y., KIRILLOV, A., MASSA, F., LO, W.-Y. and GIRSHICK, R. *Detectron2*. 2019. Available at: <https://github.com/facebookresearch/detectron2>.
- [52] ZHANG, C. *How to create custom COCO data set for instance segmentation*. Aug 2019. [Online; accessed 16-April-2021]. Available at: <https://www.dlology.com/blog/how-to-create-custom-coco-data-set-for-instance-segmentation/>.

Appendix A

Comparison of Automated and Manual Measurements

Table has abbreviated head attributes where N stands for image name, MM means manual measurement, AM means automatized measurement and D is numerical difference between two measurements.

Table A.1: Measurement comparison for full width.

N	MM(mm)	AM(mm)	D(mm)
1003_1963L_F_upr	7.07382	6.886222	0.187598
1003_1964L_F_upr	7.377183	6.942667	0.434516
1004_1971L_M_upr	8.694094	8.353778	0.340316
1004_1972L_M_upr	8.741603	8.410222	0.331381
1005_1963L_F_upr	6.913465	6.716889	0.196576
1005_1963L_M_upr	7.324088	7.055556	0.268532
1005_1964L_F_upr	7.228416	6.716889	0.511527
1005_1964L_M_upr	7.395084	7.394222	0.000862
1005_1965L_M_upr	7.837449	7.394222	0.443227
1005_1966L_F_upr	7.451522	6.999111	0.452411
1005_1966L_M_upr	7.960468	7.845778	0.11469
1005_1967L_M_upr	8.356828	8.128	0.228828
1005_1968L_M_upr	9.011687	8.297333	0.714354
1005_1969L_M_upr	9.077558	8.579556	0.498002
1005_1970L_M_upr	9.250347	8.579556	0.670791
1005_1972L_M_orig	9.258438	8.974667	0.283771
1005_1972L_M_upr	9.144174	8.974667	0.169507
1006_1963L_M_upr	7.565451	6.208889	1.356562
1006_1964L_M_upr	7.693235	7.789333	0.096098
1006_1965L_M_upr	7.90726	7.676444	0.230816
1006_1967L_F_upr	7.829314	7.337778	0.491536

1006_1968L_F_upr	7.926577	7.394222	0.532355
1006_1969L_F_upr	8.009743	7.62	0.389743
1007a_1963L_F_upr	6.515606	6.491111	0.024495
1007_1967L_F_upr	7.211868	7.168444	0.043424
1007_1970L_F_upr	7.524279	7.055556	0.468723
1009_1965L_F_upr	7.697168	7.281333	0.415835
1009_1966L_F_upr	7.871926	7.055556	0.81637
1011_1964L_F_upr	6.2694	5.926667	0.342733
1011_1965L_F_upr	6.532454	6.152444	0.38001
1013_1963L_M_upr	6.434914	6.434667	0.000247
1013_1964L_M_upr	6.781795	6.660444	0.121351
1013_1966La_M_upr	6.680743	6.434667	0.246076
1013_1966L_M_upr	7.112896	6.942667	0.170229
1013_1967L_M_upr	7.235685	6.829778	0.405907
1015_1969L_F_upr	7.689714	7.450667	0.239047
1016_1963L_F_upr	6.352195	6.096	0.256195
1017_1969L_F_upr	7.351659	6.660444	0.691215
1018_1964L_F_upr	7.676652	7.450667	0.225985

Table A.2: Measurement comparison for three regions of extended algorithm.

N	MM(mm)	AM(mm)	D(mm)
1002_1962LP_F_upr	6.282599	5.588	0.694599
1003_1963L_F_upr	7.07382	5.531555	1.542265
1003_1964L_F_upr	7.377183	6.886223	0.49096
1004_1971L_M_upr	8.694094	8.184444	0.50965
1004_1972L_M_upr	8.741603	8.240889	0.500714
1005_1963L_F_upr	6.913465	6.321777	0.591688
1005_1963L_M_upr	7.324088	6.942667	0.381421
1005_1964L_F_upr	7.228416	6.716889	0.511527
1005_1964L_M_upr	7.395084	7.168444	0.22664
1005_1965L_M_upr	7.837449	7.168445	0.669004
1005_1966L_F_upr	7.451522	6.942667	0.508855
1005_1966L_M_upr	7.960468	7.845778	0.11469
1005_1967L_M_upr	8.356828	8.071555	0.285273
1005_1968L_M_upr	9.011687	8.353777	0.65791
1005_1969L_M_upr	9.077558	8.523111	0.554447
1005_1970L_M_upr	9.250347	8.579556	0.670791
1005_1972L_M_orig	9.258438	8.974667	0.283771
1005_1972L_M_upr	9.144174	8.974666	0.169508

1006_1963L_M_upr	7.565451	6.378223	1.187228
1006_1964L_M_upr	7.693235	7.789333	0.096098
1006_1965L_M_upr	7.90726	7.224889	0.682371
1006_1967L_F_upr	7.829314	7.507111	0.322203
1006_1968L_F_upr	7.926577	6.999111	0.927466
1006_1969L_F_upr	8.009743	7.676444	0.333299
1007a_1963L_F_upr	6.515606	6.491111	0.024495
1007_1967L_F_upr	7.211868	5.305778	1.90609
1007_1970L_F_upr	7.524279	7.168444	0.355835
1009_1965L_F_upr	7.697168	6.999111	0.698057
1009_1966L_F_upr	7.871926	7.281333	0.590593
1011_1964L_F_upr	6.2694	5.926667	0.342733
1011_1965L_F_upr	6.532454	6.039555	0.492899
1013_1963L_M_upr	6.434914	6.491112	0.056198
1013_1964L_M_upr	6.781795	6.604001	0.177794
1013_1966La_M_upr	6.680743	6.321777	0.358966
1013_1966L_M_upr	7.112896	6.547556	0.56534
1013_1967L_M_upr	7.235685	6.829777	0.405908
1015_1969L_F_upr	7.689714	7.337778	0.351936
1016_1963L_F_upr	6.352195	5.926667	0.425528
1017_1969L_F_upr	7.351659	6.604	0.747659
1018_1964L_F_upr	7.676652	7.281334	0.395318

Appendix B

Attached DVD Content

- **program**
 - source files including training images and code documentation
- **images**
 - small sample of test images from X-ray database
- **text**
 - L^AT_EX source files
 - L^AT_EX generated thesis in PDF format