



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**DETEKCE, EXTRAKCE A MĚŘENÍ DÉLKY ZÁPRSTNÍCH
KOSTÍ VE SNÍMCÍCH**

DETECTION, EXTRACTION AND MEASUREMENT OF THE LENGTH OF THE METACARPAL BONES

IN IMAGES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VOJTĚCH ULEJ

VEDOUCÍ PRÁCE

SUPERVISOR

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

BRNO 2022

Zadání bakalářské práce



Student: **Ulej Vojtěch**
Program: Informační technologie
Název: **Detekce, extrakce a měření délky záprstních kostí ve snímcích**
Detection, Extraction and Measurement of the Length of the Metacarpal Bones in Images
Kategorie: Zpracování obrazu

Zadání:

1. Prostudujte literaturu týkající se snímání a zpracování snímků kostí.
2. Navrhněte algoritmus pro detekci, extrakci a měření délky záprstních kostí ve snímcích.
3. Implementujte navržený algoritmus a otestujte na dostupné databázi dat Ústavu antropologie Přírodovědecké fakulty Masarykovy univerzity.
4. Zhodnoťte dosažené výsledky a diskutujte možná rozšíření.

Literatura:

- LIU, Zhi-Qiang; AUSTIN, Timothy J.; MOORE, Daniel. Image processing techniques for bone image analysis. In: *Proceedings., International Conference on Image Processing*. IEEE, 1995. p. 458-461.
- CHALECHALE, A.; BAHRI, A.; VATANCHIAN, M. Vision-based bone image recognition using geometric properties. *Iranian Journal of Science and Technology*, 2010.
- DROZDOVÁ, Eva. *Základy osteometrie*. 1. vyd. Brno: *Nadace Universitas Masarykiana*, 2005. 196 s. Panoráma biol. a sociokulturní antropologie 18. ISBN 80-7204-291-2.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Drahanský Martin, prof. Ing., Dipl.-Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 4. listopadu 2021

Abstrakt

Cílem této bakalářské práce bylo prostudovat literaturu zabývající se tématikou rozpoznávání objektů (přesněji kostí) na snímcích. Dále shrnout existující řešení pro detekci a měření objektů ze snímků a navrhnout vhodný algoritmus pro detekci a měření metakarpálních kostí na snímcích. Pro první návrh algoritmu byly využity metody detekce hran. Avšak tento přístup se ukázal jako nespolehlivý. Druhý návrh algoritmu je založen na strojovém učení s využitím konvoluční neuronové sítě Mask RCNN.

Abstract

The aim of this bachelor thesis is to study the literature dealing with the topic of object (specifically bones) recognition in images. Another task is to summarize the existing solutions for detection and measurement of object's length and to design a suitable algorithm for detection and measurement of metacarpal bones in images. First proposed algorithm is based on edge detecting methods. However, this approach has proved unreliable. The second algorithm design is based on machine learning using the convolutional neural network Mask RCNN.

Klíčová slova

počítačové vidění, zpracování obrazu, detekce objektů, Mask RCNN, OpenCV, metakarpální kost

Keywords

computer vision, image processing, object detection, Mask RCNN, OpenCV, metacarpal bone

Citace

ULEJ, Vojtěch. *Detekce, extrakce a měření délky záprstních kostí ve snímcích*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Dipl.-Ing. Martin Dražanský, Ph.D.

Detekce, extrakce a měření délky záprstních kostí ve snímcích

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana prof. Ing. Dipl.-Ing. Martina Drahanského Ph.D. Další informace mi poskytl pan doc. RNDr. Miroslav Králík Ph.D. z Ústavu Antropologie PřF MU. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Vojtěch Ulej
11. května 2022

Obsah

1	Úvod	3
2	Teorie detekce a extrakce kostí v obraze	4
2.1	Historie a vývoj metod pro detekci objektů	4
2.1.1	Metody založené na strojovém učení	5
2.1.2	Metody založené na hlubokém učení	6
2.2	Metody zpracování snímku kostí	7
2.2.1	Detekce kostí na snímcích	7
2.3	Metody měření kostí a anatomický popis	9
2.3.1	Anatomický popis záprstních kostí	9
2.4	Metody snímání kostí	10
2.5	Datová sada	12
2.6	Nástroje a návrh postupu řešení	13
2.6.1	Nástroje	13
2.6.2	Nabízený návrh řešení	14
3	Segmentace metakarpálních kostí pomocí konvolučních neuronových sítí	17
3.1	Metody segmentace obrazu pomocí KNN	17
3.1.1	U-Net	18
3.1.2	YOLACT	18
3.1.3	Mask RCNN	18
3.2	Trénování Mask RCNN	19
3.2.1	Nástroje	19
3.2.2	Anotace dat	21
3.2.3	Trénování modelu na vlastním datové sadě	21
3.3	Segmentace a měření délky metakarpálních kostí	22
3.3.1	Segmentace metakarpálních kostí	22
3.3.2	Zpracování dat segmentace	23
3.3.3	Testování a shrnutí algoritmu pro detekci, extrakci a měření délky záprstních kostí s využitím Mask RCNN	25
3.4	Možná rozšíření algoritmu	26
3.4.1	Zvýšení přesnosti segmentace	27
3.4.2	Segmentace několika instancí kostí najednou	27
3.5	Struktura projektu	29
3.5.1	Instalace závislostí	29
3.5.2	Spuštění skriptu pro trénování	29
3.5.3	Spuštění skriptu pro detekci a segmentaci	30
3.5.4	Spuštění skriptu pro zpracování segmentačních masek	30

4 Závěr	31
Literatura	32

Kapitola 1

Úvod

Člověk dokáže detekovat a rozpoznávat objekty ve snímcích velmi jednoduše. Lidský zrak je velmi rychlý, přesný a dokáže zpracovat komplexní úlohy, jako je detekce více objektů, s malou námahou. Když dojde na měření objektů, musíme k tomu použít nástroje, jako je pravítko, metr. V 60. letech 20. století se začaly objevovat nápady, jak naučit počítač rozumět obrazu. V dnešní době toto téma není zapomenuto, ale naopak, díky rostoucímu výpočetnímu výkonu a různým nástrojům jsme schopni naučit počítače zpracovávat čím dál tím více komplexní úlohy. Zpracování obrazu pomocí počítače je velmi užitečné pro zpracování velkého objemu dat v kratším čase, než by dokázal jediný člověk a v některých případech i celý tým. Cílem této bakalářské práce je využít počítač jako nástroj pro zpracování snímků a měření objektů (přesněji záprstních kostí) na snímcích.

Rozpoznávání objektů ve snímcích pomocí počítače a automatizace úloh je v dnešní době velmi užitečná a každodenně využívána, od odemykání telefonu pomocí obličeje až po automatizaci výrobních linek s kontrolou kvality pomocí kamer. Pro detekci objektů jsou dnes běžně používané konvoluční neuronové sítě. V této bakalářské práci bych rád otestoval využití konvolučních neuronových sítí pro detekci a segmentaci metakarpálních kostí na snímcích.

Hlavním cílem této práce je navrhnout algoritmus pro detekci a měření délky metakarpálních kostí na snímcích. Tento algoritmus bude otestován na databázi snímků metakarpálních kostí, kterou poskytl Antropologický ústav PřF MU. Hlavním důvodem vzniku tohoto algoritmu by měla být automatizace získávání užitečných dat z nasnímaných kostí, které mohou být využity při dalších výzkumech.

Kapitola 2

Teorie detekce a extrakce kostí v obraze

V první části této kapitoly je popsán historický vývoj[32] a dnes používané metody pro detekci objektů ve snímcích s hlavním zaměřením na rozpoznání kostí. Hlavním účelem detekce objektů ve zpracování obrazu je určit, zda se hledaný objekt nachází v daném obraze, případně klasifikovat dané objekty do různých tříd. Dalšími důležitými informacemi jsou pozice a ohraničení objektu. Pro účely této bakalářské práce je důležité kromě detekce objektu i extrakce tvaru, případně i textury kosti. Proto se v další části zaměřím na metody použité pro zpracování snímků kostí.

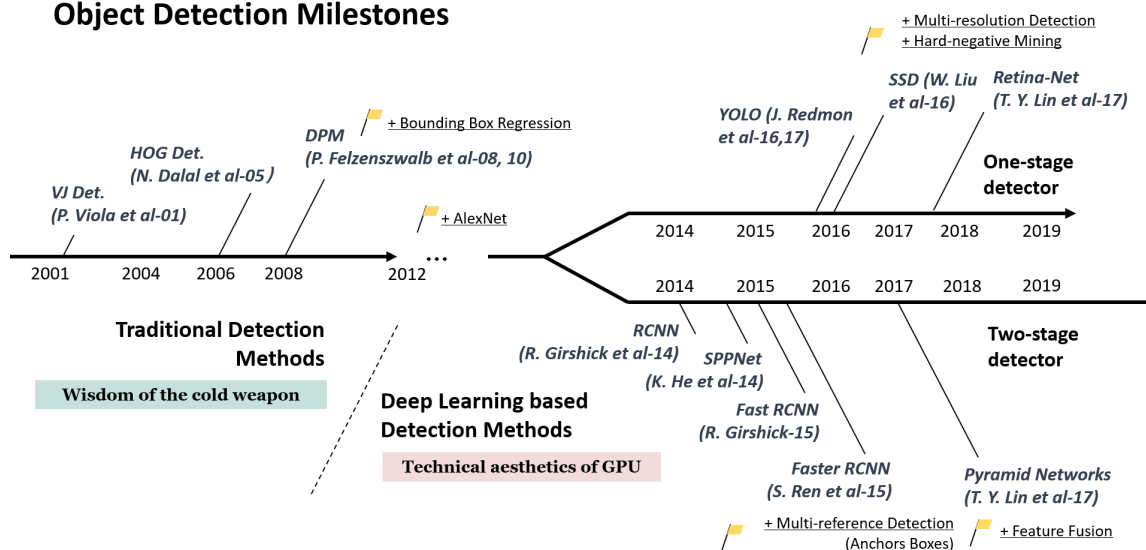
Měření záprstních kostí je součástí vědy antropometrie. V další části se budu věnovat metodám měření záprstních kostí a jejich anatomickému popisu. Dalším důležitým tématem jsou metody a technologie pro snímání kostí. V této části popíši různé způsoby, jak je možné snímat kosti, k těmto způsobům budou uvedeny i výhody, nevýhody a dostupnost potřebných technologií.

Při zpracovávání datové sady snímků můžeme narazit na snímky s různými chybami nebo artefakty, které mohou jeho zpracování ztížit, proto je důležité znát datovou sadu, kterou bude výsledný program zpracovávat. V neposlední řadě zde bude představen návrh postupu řešení a nástroje, které mohou být pro implementaci řešení využity.

2.1 Historie a vývoj metod pro detekci objektů

V této části představím metody pro detekci objektů a jejich vývoj. Časová osa na obrázku 2.1 ukazuje historický vývoj a důležité milníky detekce objektů za posledních 20 let.

Object Detection Milestones



Obrázek 2.1: Časová osa vývoje metod pro detekci objektů.[32]

2.1.1 Metody založené na strojovém učení

Níže vyjmenované metody jsou založeny na strojovém učení. Ve strojové učení, jak jeho název napovídá, se algoritmy zdokonalují (učí) na základě trénovacích dat, se kterými pracují v trénovací fázi.

Viola Jones Detectors

Tato metoda byla představena v roce 2001. Za jejím vznikem stojí Paul Viola a Michael Jones, po kterých se metoda jmenuje. Jedná se o nejrychlejší metodu své doby, která byla schopna detekovat objekty (hlavně lidskou tvář) téměř v reálném čase (až 15 snímků za sekundu, na tehdejší dostupném hardwaru), bez jakýchkoliv omezení[26][27](jako je barva pleti, atd.). Tento algoritmus byl desetkrát až stokrát rychlejší než jiné algoritmy tehdejší doby.

Algoritmus využívá k detekci klouzavého okna, které prochází všechna místa, ve kterých by se mohl vyskytnout objekt. I když tento způsob vypadá velmi jednoduše, tehdejší počítače neměly dostatečný výpočetní výkon na provedení tohoto výpočtu v reálném čase. VJ algoritmus proto používal 3 klíčové techniky, které zpracování obrazu dramaticky urychlily, jedná se o:

- **Integrální obraz** - Výpočetní metoda pro urychlení konvolučního procesu.
- **Výběr příznaků** - Využití Adaboost algoritmu[32] pro výběr malého množství příznaků užitečných při detekci obrazu.
- **Detekční kaskády** - Vícetupňové paradigma pro detekci objektů. Toto paradigma snižuje výpočetní nároky tak, že klasifikátory na nižších stupních detekční kaskády neprohledávají okénka, ve kterých jeden z klasifikátorů vyššího stupně nedetekoval objekt.

HOG Detector

V roce 2005 N. Dalal a B. Triggs představili metodu *Histogram orientovaných gradientů*[32]. Tato metoda je založená na histogramech orientace hran v obraze.

Deformable Part-based Model (DPM)

Tato metoda byla publikována v roce 2008 autory P. Felzenszwalb, D. McAllester a D. Ramanan[32]. Principem metody je rozdělení detekovaného objektu na jednodušší části. Tento algoritmus byl několikrát vylepšený R. Girshickem. Jedním ze zásadních vylepšení byla technika pro kompilaci modelů s použitím kaskádové architektury detekce, tato technika zvýšila rychlost detekce až desetkrát bez snížení přesnosti.

2.1.2 Metody založené na hlubokém učení

V této části se zaměřím na metody založené na hlubokém učení. Hluboké učení je podkapitola strojového učení. Metody jsou založené na umělých neuronových sítích. Přídavné jméno “hluboké” značí, že se v neuronové síti nachází více skrytých vrstev. Od vydání metody DPM, zmíněné v předchozí části 2.1.1, mezi lety 2010 až 2012 se vývoj detekce objektů zpomalil a docházelo pouze k vylepšování již existujících metod. V roce 2012 došlo k převratu v přístupu k detekci objektů, když tvůrci algoritmu AlexNet [16] zvítězili s velikým náskokem v soutěži ILSVRC2012 [22].

AlexNet je založený na hlubokých konvolučních neuronových sítích (dále jen CNN). Výhodou tohoto přístupu je, že CNN jsou schopny se učit složité reprezentace objektů. Nevýhodou je požadavek obrovského výpočetního výkonu, který roste s velikostí neuronové sítě. Výpočty jsou proto urychlovány na grafických kartách, programovatelných polích FPGA, nebo na specializovaném hardwaru jako je TPU od společnosti Google[20].

RCNN (AlexNet)

Zkratka RCNN stojí za názvem (přeloženo do češtiny) Regionální konvoluční síť. Přístup této metody k detekci objektů je následující:

- **Výběr kandidátních pozic ve snímku** - Dojde k ohraničení částí obrázku, ve kterých by se mohl vyskytovat objekt.
- Vybrané části jsou upraveny na fixní rozměr a zpracovány pomocí CNN.
- Natrénovaný CNN model vytáhne z části obrázku příznaky.
- **SVM** - Lineární klasifikátor klasifikuje objekty do jednotlivých kategorií (pokud jsou objekty nalezeny)

Tato metoda je, v porovnání s tehdejšími metodami velmi přesná, avšak velmi výpočetně náročná, a tudíž pomalá (14 sekund na jeden obrázek). Problém s rychlostí detekce vyřešilo vylepšení SPPNet[32].

Fast RCNN/Faster RCNN

Jedná se o dvě vylepšení metody RCNN tak, aby detekce probíhala rychleji. **Fast RCNN**[32] byla představena roku 2015, rychlost tohoto přístupu byla až 200krát větší než u předchozí metody.

Následující rok byla představena metoda **Faster RCNN**[32], tato metoda dosahovala rychlosti zpracování až 17 snímků za sekundu. Tento algoritmus představil tzv. Regional

Proposal Network (RPN), který nahrazuje dosud používané algoritmy pro generování *regionů zájmu*.

You Only Look Once (YOLO)

V roce 2015 byl dosažen významný milník pro jednorůchodové detekční metody. Algoritmus YOLO[32], jak jeho název napovídá, prochází obraz jen jednou a při průchodu detekuje všechny objekty, které se na něm nacházejí. Autoři této metody opustili princip generování *oblastí zájmu* a následnou verifikaci a místo toho použili jedinou neuronovou síť na celý obraz. Tato síť rozdělí obraz na buňky, v každé buňce je vytvořeno několik ohraničujících oblastí (bounding box), skóre oblastí určuje, zda oblast obsahuje objekt.

Tento přístup se ukázal jako velice rychlý, jelikož metoda je schopna zpracovávat velké množství obrazů v krátkém čase, až 155 snímků za sekundu s úspěšností mAp (mean average precision) 52.7 %.

Single Shot MultiBox Detector (SSD)

SSD[32] metoda byla publikována v roce 2016, stejně jako metoda YOLO. Jedná se o jednorůchodový detekční algoritmus. Výhodou tohoto algoritmu je jeho úspěšnost při zachování vysoké rychlosti detekce (až mAp=76.8%; až 59 snímků za sekundu).

2.2 Metody zpracování snímku kostí

V podkapitole 2.1 jsme si představili metody pro detekci a klasifikaci objektů vyvinuté v tomto století. Tato část kapitoly je zaměřena na metody využití při zpracování snímků kostí. Při extrakci objektu z obrazu je naším cílem detekovat hrany relevantního objektu, plochu nacházející se vně těchto hran můžeme považovat za pozadí, tedy oblast, která nás nezajímá. Zpracování snímků kostí úzce souvisí s pojmem *rozpoznávání vzoru* (z anglického Pattern recognition)[2]. Tento princip má široké použití při zpracování signálů, analýze dat, zpracování obrazu a v dalších disciplínách.

2.2.1 Detekce kostí na snímcích

Analýza snímků kostí má veliký přínos pro zpracování lékařských snímků (rentgenové snímky, CT, atd.), pro detekci nemocí spojených s kostmi. Dalším využitím je získání statistických údajů, jako je například délka kosti ze snímků pořízených archeology nebo antropology. Dalším užitečným údajem je tvar kosti, tato informace nám může prozradit, kterému živočichovi kost patří a kde se na těle nachází.

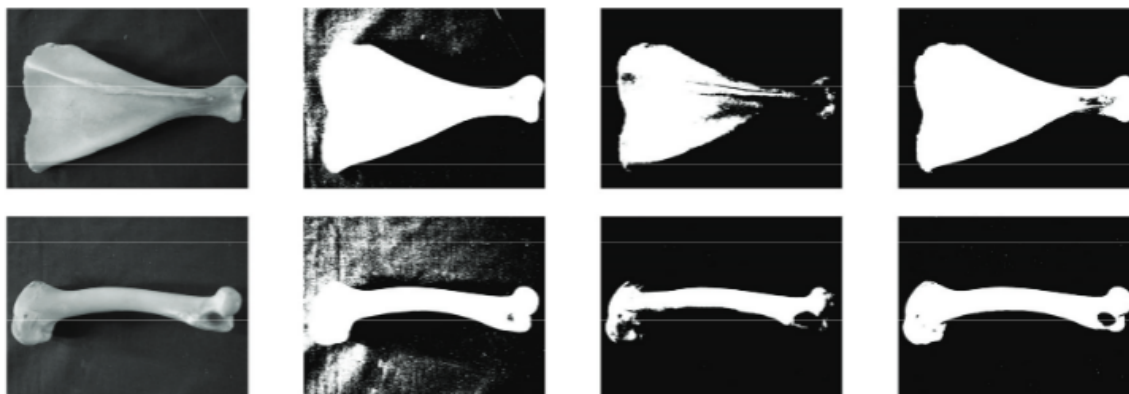
Rozpoznání kosti na snímku použitím geometrických vlastností

Tato část je převzata z [2]. Tento postup je založen na extrakci příznaků a statistické klasifikaci. Prvním krokem této metody je předzpracování obrazu. Barevný obraz je převeden na odstíny šedi tak, aby došlo k eliminaci odstínu a sytosti barev, ale byl zachován jas. Dále se provede binarizace obrazu na bílou a černou barvu. Na obrázku 2.2 jsou vidět různé metody binarizace obrazu z obrazu převedeného na odstíny šedi. Dále došlo k normalizaci velikosti obrazu na velikost $w \times h$ pixelů (256×256 pixelů).

Dalším krokem je extrakce příznaků. Autoři článku k tomuto účelu využívali 3 různé metody[2]. Pro každý předzpracovaný snímek jsou extrahovány následující geometrické vlastnosti:

- obsah
- obvod
- délka hlavní poloosy
- délka vedlejší poloosy
- excentricita

Tyto vlastnosti byly předány klasifikátorům, kteří určili druh živočicha a kosti.



Obrázek 2.2: Obrázek převedený do odstínů šedi (první sloupec), nízký práh vytvoří mnoho nechtěných zarušených oblastí (druhý sloupec) a vyšší práh zničí oblast kosti (třetí sloupec). Použití shlukovací metody nejbližších středů (K-means clustering) správně oddělí oblast kosti od pozadí (čtvrtý sloupec).[2]

Active shape models (ASM)

ASM je velmi populární rodina algoritmů pro segmentaci lékařských obrazů. Výhodou této metody je rychlost zpracování obrazu. Nevýhodou je limitovaná možnost zobecnění tvarů. Tato limitace je způsobena použitím SSM (statical shape model), který nesprávně určí zdeformované objekty.[15]

Modely jsou geometrické popisy objektů. Tvary objektů stejné třídy se však mohou lišit, například tvar kosti se mění s věkem osoby. Proto se musí dané modely přizpůsobovat. Flexibilní modely jsou složeny z geometrických tvarů, například body, úsečky, kruh, nebo křivky. Každému tvaru je přidělen stupeň volnosti (degree of freedom), který určuje, jak mohou tvary měnit měřítko nebo i pozici[4].

Active appearance models (AAM)

AAM[7] je podobný předchozí metodě ASM. Hlavním rozdílem je, že k tvaru objektu přibyla ještě textura. Jedná se o vysoce flexibilní model. K trénování tohoto algoritmu je potřeba využít anotovaná data. Trénovací data by měla obsahovat modelová řešení daného problému. Když dojde k detekci objektu pomocí AAM, vzniklý model je schopný popsat reálné vlastnosti detekovaného objektu, jako je tvar, rozměry i barva povrchu[24]. Nevýhodou tohoto přístupu je složité vytváření trénovacích dat.

2.3 Metody měření kostí a anatomický popis

Metody měření kostí jsou součástí vědy antropometrie[6]. Antropometrie je věda zaměřena na měření lidského těla. Její metodika je velmi široká. V antropometrii je velmi důležitá co nejpřesnější definice antropologických bodů, rozměrů a technik měření. Tato standardizace používaných bodů a měř je výhodná při srovnávání antropometrických výzkumů populací po celém světě. Pro měření kostí se používají normované antropometrické nástroje.

Antropometrie se dělí na somatometrii a osteometrii[6]. Somatometrie se zabývá měřením živého člověka. V této bakalářské práci se budu zabývat osteometrií, která se zaměřuje na měření kosterních pozůstatků. V osteometrii dělíme měření na kranimetrii (měření lebky) a osteometrii postkraniálního skeletu, do kterého spadají záprstní kosti (metakarpý).

V osteometrii je velmi důležitá přesná poloha antropometrických bodů, proto je potřebné se správně orientovat v anatomické poloze těla. Anatomické postavení těla vypadá takto: tělo je vzpřímené, hlava hledí přímo vpřed, horní končetiny jsou svěšeny podél boků a dlaně jsou obráceny vpřed s palci směřujícími zevně. Dolní končetiny jsou nataženy. Vždy se řídíme podle stran těla nebo kostry (popř. částí), nikoliv však podle své polohy. Důležité je zachovat správné strany i při vyobrazení těla či částí. Například pravá strana je vždy tam, kde je pravá strana i u vyobrazeného těla. Tato část byla převzata z publikace Horáckové a Maliny[14]. Pro správné určení antropometrických bodů je nutné znát roviny na těle. Roviny rozlišujeme tři a jsou to:

- **Sagitální rovina** - taktéž šípová. Míří kolmo na podélnou osu těla a prochází tělem zepředu dozadu (od břicha na záda)
- **Frontální rovina** - je svislá, je rovnoběžná s čelem, dělí tělo na přední a zadní část
- **Transverzální rovina** - prochází tělem příčně a tvoří horní a dolní část.

Abychom se správně zorientovali v poloze metakarpálních kostí, je potřebné uvést označení směrů a poloh na horní končetině:

- **Proximalis** - Označení pro polohu, značící nejbližší připojení končetiny k trupu.
- **Distalis** - Značí polohu pro vzdálenější nejbližší připojení končetiny k trupu.
- **Ulnaris** - Vnitřní směr, též malíkový – uložený na malíkové straně předloktí.
- **Radialis** - Vnější směr, taktéž palcový – uložený na palcové straně předloktí.
- **Palmaris** - Dlaňový směr, směr do dlaně.
- **Dorsalis** - Hřbetní směr, směr k hřbetu ruky.

2.3.1 Anatomický popis záprstních kostí

V lidské ruce se nachází pět záprstních kostí, které se latinsky nazývají ossa metacarpi I.-V.[14]. Záprstní kosti se číslují od palcové strany. Distálně se zakončují hlavicí (caput ossis metacarpi), na proximální straně nalezneme bázi (basis ossis metacarpi) s kloubními ploškami pro spojení s karpálními kostmi. Na bázích metakarpů (kromě I. a V. metakarpu) nalezneme na mediální a laterální straně také kloubní plošky pro spojení s vedlejšími metakarpálními kostmi. Mezi hlavicí a bází se nachází tělo (corpus ossis metacarpi), které je směrem k hřbetu ruky mírně konvexní (vypouklé).

Os metacarpale I.

Je palcová záprstní kost. Má nejkratší a nejsilnější vzhled ze všech metakarpálních kostí.

Os metacarpale II.

Tato metakarpální kost se nachází jako druhá od palcové strany ruky, je nejdelší metakarpální kostí.

Os metacarpale III.

Je třetí záprstní kostí od palcové strany. Na této záprstní kosti můžeme v místě báze naleznout strukturu jménem processus styloideus.

Os metacarpale IV.

Tato metakarpální kost je v porovnání s II. a III. metakarpální kostí poněkud slabší a kratší.

Os metacarpale V.

Jedná se o poslední z metakarpálních kostí, nachází se na malíkové straně ruky.

Jelikož tématem mé bakalářské práce je měření záprstních kostí, pro správné antropologické měření je potřebné zmínit standardizované vzdálenosti. Jsou jimi:

- **M2 Délka jednotlivých záprstních kostí** - „Přímočará vzdálenost středu plochy báze od vrcholového bodu hlavice. Susman (1979) měří největší délku od nejproximálněji položeného bodu báze.[14]“
- **M6 Radioulnární průměr diafýzy** - „Největší projektivní vzdálenost radiální od ulnární strany diafýzy, měřená v jejím středu, kolmo k podélné ose kosti.[14]“
- **M7 Dorzopalmární průměr diafýzy** - „Největší projektivní vzdálenost dorzální strany diafýzy od palmární, měřená kolmo k podélné ose kosti a v jejím středu.[14]“
- **M8 Největší proximální šířka** - „Největší projektivní šířka proximálního konce kosti měřená kolmo k podélné ose kosti.[14]“
- **M9 Největší distální šířka** - „Největší projektivní šířka distálního konce kloubu, měřená kolmo k podélné ose kosti.[14]“

2.4 Metody snímání kostí

V této části kapitoly se zaměřím na postup a technologie snímání kosterních pozůstatků. Tato část je převzata z publikace Garvina a Stocka[9]. Na počátku 20. století obdržel německý fyzik Wilhelm Conrad Röntgen Nobelovu cenu za objev krátkovlnného záření známé jako rentgenové záření (pojmenované po svém objeviteli). Tento objev zapříčinil velký technologický vývoj, hlavně v oblastech lékařství a inženýrství. Výsledkem tohoto vývoje je široké spektrum pokročilých technologií pro snímání lidského těla. Od rentgenových snímků až po trojrozměrné zobrazení lidských orgánů a skeletu pomocí výpočetní tomografie. Díky vyšší dostupnosti těchto technologií se stále častěji můžeme setkat se snímky pořízenými novými technologiemi.

Dále popíši konkrétní metody pořizování a uchovávání snímků, jejich principy, shrnu jejich výhody a nevýhody.

Fotografické snímky

Způsob, jak uchovávat fotografické snímky, je více jak 100 let starý, avšak v kontextu se strojovým viděním bylo dosaženo průlomu v 50. letech 20. století, když byl poprvé vytvořen digitální snímek. K hojnému používání digitálních fotoaparátů došlo ke konci 20. století, kdy se začaly vyrábět fotoaparáty, které byly finančně dostupnější širší veřejnosti. V dnešní době jsou digitální fotoaparáty nejrozšířenější technologií pro snímání fotek, jsou velmi levné a poskytují vysokou kvalitu fotografií.

Fotografování je nejjednodušší a nejlevnější způsob zdokumentování kosterních pozůstatků. Fotografie pomáhají zachytit kosterní pozůstatky v širším kontextu, jako je například poloha, ve které byla kostra nalezena. Pro kompletní dokumentaci jsou na fotografii zachyceny i okolní nálezy. Tyto informace jsou důležité pro antropology a archeology, kteří z nich například mohou odhadnout dobu, ze které kostra a s ní spojené hrobové místo pochází.

Po exkavaci a očištění nálezu jsou kosti znovu vyfoceny k účelům budoucího zkoumání. Pro účely detailnějšího výzkumu mohou být fotografie pořízeny pod mikroskopem. Kromě fotoaparátů a mikroskopů mohou být digitální fotografie zachyceny pomocí skeneru. Výhodou tohoto postupu jsou vysoká kvalita nebo zachování rozměrů pro snadnější měření prováděné na počítači.

Rentgenový snímek

Rentgenové snímky jsou užitečné nejen pro lékařské účely, ale i pro snímání kosterních pozůstatků. Rentgenové záření prostoupí tkání a částečně i povrchem kosti a zachytí poškození, které není na normálních snímcích vidět. Přístroj pro snímání vydává z jedné strany rentgenové záření, které je na druhé straně zachyceno buď na rentgenový film, nebo u novějších přístrojů detektorem, který vytvoří digitální obraz.

Výhodou tohoto snímkování je, že můžeme zachytit snímek kosti ze živého organismu, nebo i z neočištěného kosterního pozůstatku. Další výhodou je, že rentgenové paprsky proniknou částečně i povrchem kosti a měkkými tkáněmi, na snímcích mohou být vidět nálezy, které na klasických fotografiích nejsou viditelné. Mezi takové nálezy mohou patřit různá zranění, nebo také šrapnely (např. kovové předměty).

Výpočetní tomografie

Výpočetní tomografie je spíše známá pod svou zkratkou CT z anglického *Computed Tomography*. Tato technologie se používá primárně v lékařství k neinvazivnímu zobrazení orgánů a tkání. Principem této technologie je získání více snímků z více úhlů, pomocí rentgenového záření. Tyto snímky jsou pomocí počítače "*spojeny*" do tomografických snímků. Tomografické snímky mohou být dále "*poskládány*" do virtuální trojrozměrné rekonstrukce objektu. Snímky pořízené touto technologií nejsou velmi časté kvůli vysoké ceně CT přístrojů. Výhodou je i zachycení vnitřní struktury objektu.

Trojrozměrné povrchové modely

Tyto modely jsou pořízeny 3D skenery, které generují buďto mračno bodů (body na povrchu objektu se souřadnicemi (x, y, z)), nebo trojúhelníkovou síť, která tvoří povrch objektu. Na rozdíl od CT, které fungují na principu rentgenových snímků, 3D skenery zachycují pouze povrch kosti za podmínky, že je tento povrch v přímé viditelnosti 3D skeneru. Jelikož 3D skenery snímají pouze povrch kosti, je nutné před snímáním kosterní pozůstatky očistit.

Skenery dělíme na kontaktní a bezkontaktní. Kontaktní skenery procházejí zadanou oblast a při kontaktu s objektem si zaznamenají polohu bodu, ve kterém došlo ke kontaktu. Bezkontaktní skenery používají pro skenování povrchu světlo (většinou laser), ultrazvuk,

nebo rentgenové záření. Dále můžeme skenery dělit na ruční a statické. Statické skenery vyžadují, aby se při skenování objekt otáčel, zatímco ručními skenery pohybuje obsluha a skenovaný objekt se musí nacházet ve statické poloze. Většina skenerů pořizuje v průběhu skenování fotografie objektu, díky kterým je zachycena původní barva objektu. Výhodou této metody je nízká pořizovací cena ve srovnání s výpočetní tomografií.

Fotogrammetrie

Další cenově dostupná metoda pro tvorbu trojrozměrného modelu kosti. Fotogrammetrie byla vyvinuta před 3D skenery a výpočetní tomografií. Tato technologie vytváří z klasických digitálních snímků pozici, orientaci, tvar a velikost objektu, tudíž není potřeba žádného specializovaného a drahého zařízení. Principem této metody je, že pomocí specializovaného softwaru vytvoříme z více digitálních fotografií, pořízených z různých směrů a úhlů, trojrozměrný model objektu. Hlavní nevýhodou této technologie je její nepřesnost v porovnání s 3D skenery, které používají vyspělejší technologie k změření vzdálenosti od objektu. Další nevýhodou je složitost focení snímků. Pro přesnou rekonstrukci trojrozměrného modelu je nutné, aby byly snímky vytvořeny v přesném systematickém pořadí.

2.5 Datová sada

Datovou sadu poskytl Ústav Antropologie Přírodovědecké fakulty Masarykovy univerzity. Snímky byly pořízeny doc. RNDr. Miroslavem Králíkem, Ph.D. na základě athénské skeletální sbírky uložené v Department of Animal and Human Physiology – University of Athens[8].

Datová sada obsahuje snímky metakarpálních (záprstních) kostí. Každá kost byla nasnímána ze dvou směrů dorsálního (hřbetní směr) a radiálního (palcový směr). Při snímání kostí z radiálního směru byly některé kosti připevněny modelovací hmotou tak, že je část kosti hmotou překryta (jak je vidět na obrázku 2.3). Toto překrytí ztěžuje detekci a extrakci tvaru kosti ze snímků. V následující kapitole 2.6 se budu této problematice věnovat více.



Obrázek 2.3: Snímek metakarpální kosti z radiálního směru. Část kosti na snímku je překryta modelovací hmotou.



Obrázek 2.4: Snímek metakarpální kosti z dorsálního směru.

Dvourozměrné snímky byly pořízeny klasickým plošným skenerem v rozlišení 600 dpi v měřítku 1:1, s barevnou hloubkou 24 bit RGB a uloženy ve formátu **TIFF**. Údaje uložené v hlavičce formátu **TIFF** nám pomohou při zpracování snímku a měření délky kostí. Informace jsou anotovány tagy, mezi takové informace patří:

- **ResolutionUnit** - jednotka vzdálenosti pro (X/Y)Resolution (hodnoty nespécifkováno, palce, centimetry)

- **XResolution** - počet pixelů za jednotku určenou v tagu **ResolutionUnit** ve směru šířky obrázku
- **YResolution** - počet pixelů za jednotku určenou v tagu **ResolutionUnit** ve směru výšky obrázku
- **ImageLength** - počet řádků pixelů ve snímku, neboli počet pixelů v jednom sloupci
- **ImageWidth** - počet pixelů v jednom řádku snímku, neboli počet sloupců pixelů ve snímku

2.6 Nástroje a návrh postupu řešení

V této podkapitole představím návrh řešení detekce hran a extrakce tvaru kosti ze snímku. Zaměřím se také na možné řešení problému překrytí části kosti. Zmíním zde různé nástroje, vhodné pro vývoj programu pro zpracování databáze snímků kostí.

2.6.1 Nástroje

Hlavním nástrojem pro tvorbu programů je programovací jazyk. Jako programovací jazyk jsem si vybral **Python**[19], kvůli jeho široké nabídce balíčků pro zpracování obrazu. V této podkapitole představím knihovny a balíčky dostupné pro jazyk Python, které se používají pro zpracování obrazu, strojové učení a další užitečné nástroje.

NumPy

Digitální snímek nahraný v počítači je vlastně číselné pole, kde je jeden pixel reprezentován například 3 hodnotami (v případě RGB). Knihovna NumPy[12] je standardizovaná a většina dále zmíněných balíčků tuto knihovnu na pozadí používá. Tato knihovna poskytuje rozhraní pro efektivní práci s jednorozměrnými, ale i vícerozměrnými poli. Knihovna poskytuje vysoce optimalizované funkce, jako jsou sčítání, násobení matic nebo i složitější funkce jako je diskrétní Fourierova transformace.

SciPy

Tato knihovna[28] se využívá na vědecké a matematické výpočty, na pozadí je využita knihovna NumPy. Pro zpracování snímků tato knihovna poskytuje různé funkce pro filtrování nebo konvoluci.

Sci-kit Image

Tento balíček[30] nabízí veliké množství funkcí pro zpracování obrazu. Mezi tyto funkce patří různé geometrické transformace, manipulace s barevným prostorem snímku, segmentace obrazu (oddělení objektů v obraze), filtrační funkce a další.

OpenCV

Knihovna OpenCV[17] obsahuje funkce a algoritmy pro zpracování obrazu, podobně jako Sci-kit Image. Tato knihovna je velmi používaná pro detekci objektů, segmentaci snímků, s možností akcelerace výpočtů na specializovaném hardwaru.

Pillow/PIL

Tato knihovna[3] není určena pro strojové učení ani detekci objektů. Její funkce se zabývají spíše úpravami snímků, jako je rotace, změnění měřítka nebo převedení snímku do jiného formátu.

Google Colaboratory

Jedná se o webovou aplikaci, která umožňuje psát a spouštět kód napsaný v jazyce Python. Tato aplikace je určena pro vzdělávání a výzkum v oborech jako je strojové učení, uživatelé tak dostanou zdarma časově omezený přístup k výpočetním zdrojům, jako jsou výkonné grafické karty nebo TPU. [10]

2.6.2 Nabízený návrh řešení

Všechny metody zde popsané jsou funkce knihovny OpenCV, které byly při implementaci tohoto řešení použity.

Převod na odstín šedi

Nejprve je potřeba snímek převést do odstínů šedi. Pro převod z barevného modelu RGB do odstínů šedi je použit následující vzorec:

$$G = 0.299 * R + 0.587 * G + 0.114 * B \quad (2.1)$$

Pro každý pixel ve snímku je vypočítána hodnota G (dle vzorce 2.1), kde R je hodnota červené barvy v pixelu, G hodnota zelené a B hodnota modré.

Filtrace snímku

Aplikace filtrů je důležitá pro odstranění šumu na snímcích. V této kapitole shrnu několik filtrů.

Gaussův filtr

Jedná se o konvoluční filtr, jehož matice je složena z prvků určených Gaussovou funkcí (viz vzorec 2.2).

$$G(x, y) = \frac{1}{2 * \pi * \sigma^2} * e^{-\frac{x^2+y^2}{2*\sigma^2}} \quad (2.2)$$

Aplikací tohoto filtru dochází k “rozmazání” obrazu, tudíž detekce hran je složitější.

Mediánový filtr

Mediánový filtr je jeden z příkladů nelineárního filtru. Tento filtr z hodnot všech pixelů v okolí vybere medián.

Bilaterální filtr

Dalším případem nelineárního filtru je bilaterální filtr. Hlavní výhodou tohoto filtru je zachování ostrých hran. Díky zachování hran je tento filtr nejvhodnější, a proto byl při implementaci použit.

Binarizace obrazu

Binarizace obrazu je převedení objektů, nebo jejich hran na bílou barvu a pozadí na barvu černou (nebo naopak). Pro binarizaci obrazu existuje mnoho metod, některé z nich v této podkapitole uvedu.

Prahování

Pro obraz je určena hodnota “prahu”, každý pixel je porovnán s hodnotou “prahu”, pokud je hodnota pixelu nižší, pak je pixelu nastavena hodnota 0 (černá barva). Jestli je hodnota pixelu vyšší, pak je mu přidělena hodnota 255 (bílá barva).

Adaptivní prahování

Tato metoda je podobná té předchozí, avšak není zde určena jednotná hodnota “prahu” pro celý obraz, nýbrž hodnota je určena dle okolních pixelů. Knihovna OpenCV nabízí 2 možnosti určování hodnot prahu. První možností je průměr okolních pixelů po odečtení konstanty C . Druhou možností je součet hodnot vážených Gaussovou funkcí a odečtení konstanty C .

Metoda Otsu

Metoda Otsu je založená na předpokladu, že na histogramu hodnot pixelů obrazu se nachází 2 skupiny intenzit. Jedna skupina reprezentuje pozadí a druhá objekt. Hodnota “prahu” je určena mezi těmito skupinami.

Cannyho detekční metoda hran

Tento algoritmus se používá k detekci hran. Každému pixelu hrany je přidělena hodnota 255 a všem ostatním hodnota 0. Algoritmus se skládá z několika kroků. Prvním krokem je odstranění šumu pomocí Gaussova filtru, který byl popsán v kapitole 2.6.2.

Druhým krokem je určení gradientů pomocí Sobelova operátoru, horizontálně (G_x) i vertikálně (G_y) z těchto hodnot jsou vypočítány gradienty a směry hran pro každý pixel v obrazu.

Ve třetím kroku algoritmus projde každý pixel obrazu a odstraní ty pixely, jejichž hodnota gradientu není lokálním maximem okolních pixelů (non-maximum suppression).

Posledním krokem je prahování s hysterezí. Určíme 2 hodnoty “prahů”, minimální a maximální. Každý pixel s hodnotou gradientu pod minimální hodnotou prahu není považován za hranu a každý pixel nad maximální hodnotou je považován za hranový pixel. Pixely s hodnotou mezi minimální a maximální hodnotou jsou považovány za hrany pouze pokud jsou navázány na pixel, který byl dříve označen jako hranový.

Algoritmus segmentace a měření kostí

Pro segmentaci snímku jsem použil následující postup. Nejprve je snímek převeden do odstínů šedi. Na snímek je poté aplikován bilaterální filtr, který měl ze všech výše uvedených filtrů nejlepší účinek při vyhlazení snímku a zachování hran.

Na binarizaci obrazu byl použit Cannyho detekční algoritmus. Pro nalezení kontury objektu je na binarizovaný snímek použita funkce z knihovny OpenCV *cv.findContour*. Z bodů kontury kosti můžeme určit délku kosti výpočtem euklidovské vzdálenosti D_E (viz. vzorec 2.3).

$$D_E = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.3)$$

Kde (x_1, y_1) a (x_2, y_2) jsou body, mezi kterými počítáme vzdálenost.

Výsledkem tohoto výpočtu je vzdálenost mezi dvěma body v pixelech. Z metadat souboru snímku můžeme zjistit počet pixelů na jeden centimetr. Díky této hodnotě můžeme vypočítat délku kosti následujícím vzorcem:

$$D = \frac{D_{pixely}}{P_{perunit}} \quad (2.4)$$

Kde D je výsledná vzdálenost v centimetrech, D_{pixely} je vzdálenost v pixelech a $P_{perunit}$ je počet pixelů na jeden centimetr.

Tento postup však neposkytoval uspokojující výsledky. Tato metoda nesprávně segmentovala snímky, které byly světlejší, nebo jejichž hrany byly rozmazané, případně snímky, na nichž byla kost překryta modelovací hmotou. Implementace tohoto řešení se nachází ve skriptu *edge_based_segm.py*. V následující kapitole popíši metodu segmentace kostí založené na konvolučních neuronových sítích.

Kapitola 3

Segmentace metakarpálních kostí pomocí konvolučních neuronových sítí

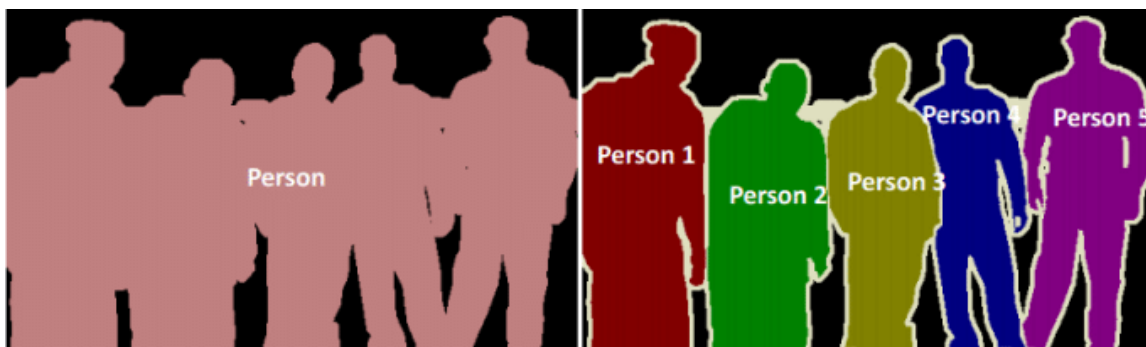
V první části této kapitoly popíšu segmentaci obrazu pomocí konvolučních neuronových sítí. Dále se budu zabývat praktickým využitím této metody pro tvorbu algoritmu na segmentaci metakarpálních kostí ze snímků. Popíšu použité nástroje a implementaci tohoto algoritmu a jeho použití na datové sadě popsané v části 2.5.

3.1 Metody segmentace obrazu pomocí KNN

V sekci 2.1.1 jsem již pár konvolučních neuronových sítí popisoval. Tyto metody se používají spíše k detekci a klasifikaci objektů, pro segmentaci jsou proto nevhodné. Existují neuronové sítě jako třeba plně konvoluční síť (FCN), U-net, nebo Mask-RCNN, které vytvářejí segmentační masky. Tyto masky přiřazují každému pixelu značku třídy, u segmentačních masek většinou rozlišujeme 3 třídy. První třídou je pixel patřící segmentovanému objektu, další třída označuje pixely, které ohraničují objekt, a poslední třídou jsou označeny pixely “pozadí” (mohou patřit jinému objektu).

Sémantická segmentace a segmentace instancí

Při sémantické segmentaci je každému pixelu v celém obraze přiřazena značka třídy, takto vytvořené masky tudíž nerozlišují mezi jednotlivými instancemi objektů. Tento typ segmentace je vhodný například pro klasifikace snímků ze satelitů. Nevýhodou tohoto typu je, že pro každou detekovanou třídu generují pouze jednu masku. Toto řeší segmentace instancí, která generuje segmentační masku pro každou instanci objektu. Rozdíly můžeme vidět na obrázku 3.1. Při sémantické segmentaci obrazu 3.1 vidíme, že většinu obrazu je tvořena lidmi (třídou Person), avšak už nejsme schopni jednotlivé masky lidí od sebe oddělit, a proto například nemůžeme určit počet lidí na snímku, jelikož se masky mohou překrývat. Při segmentaci instancí můžeme určit počet osob na obrázku 3.1 vpravo a případně každou osobu zvlášť oddělit od ostatních.



Obrázek 3.1: Sémantická segmentace vlevo a segmentace instancí vpravo[25]

3.1.1 U-Net

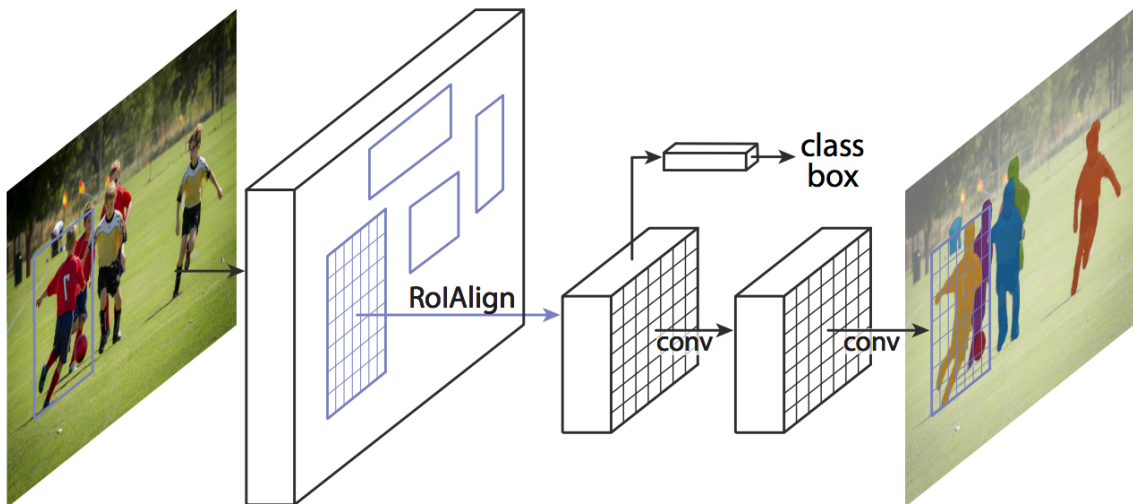
Jedná se konvoluční neuronovou síť, představenou v roce 2015[21]. Tato metoda se zaměřovala na segmentaci snímků v lékařství (např. segmentace buněk z mikroskopických snímků). Tato neuronová síť vznikla na základě plně konvolučních sítí a byla navržena pro trénování s malým počtem snímků v trénovací datové sadě. Autoři publikace také zdůrazňují pozitivní přínos augmentace snímků při trénování sítě U-Net.

3.1.2 YOLACT

Tato metoda se využívá k segmentaci obrazu v reálném čase. V době vydání byla považována za nejrychlejší metodu segmentace, která dosahovala rychlosti zpracování až 33 snímků za vteřinu. Této rychlosti zpracování autoři metody dosáhli rozdělením segmentace snímku na dva paralelní úlohy. První úlohou je tvorba prototypů masek a druhou je predikce koeficientů pro každou masku instance. Dále jsou masky instancí vytvořeny lineární kombinací prototypů masek s koeficienty masek.[11]

3.1.3 Mask RCNN

Tato metoda byla představena v roce 2017 skupinou vědců z FAIR (Facebook AI Research).[13] Jedná se o rozšíření rychlých regionální konvoluční sítí zmiňovaných v sekci 2.1.2. Jedná se o spojení 2 spolehlivých systémů Faster RCNN pro detekci objektů a FCN pro sémantickou segmentaci. Cílem Mask RCNN je dosáhnout segmentace všech instancí v obrazu. Faster RCNN detekuje jednotlivé objekty v obrazu a vytváří oblasti zájmu (region of interest), na kterých je dále prováděna klasifikace a tvorba ohraničujícího rámečku (z anglického bounding box). Mask RCNN tuto metodu rozšířila o paralelní větev, která současně s klasifikací a tvorbou rámečku na každém regionu zájmu generuje segmentační masku (obrázek 3.2). Tato větev pro generování segmentační masky je složena z jednoduché plně konvoluční sítě FCN.



Obrázek 3.2: Mask RCNN [13]

Tato metoda je implementována a můžeme ji nalézt jako součást platformy Detectron2[31] napsané ve frameworku PyTorch, nebo Matterport Mask RCNN[1] implementovaná pomocí frameworků Tensorflow a Keras.

3.2 Trénování Mask RCNN

Abychom mohli použít Mask RCNN pro segmentaci metakarpálních kostí, musíme nejprve naučit (natrénovat) model, který je schopný tento úkol splnit. Rozlišujeme více druhů učení neuronových sítí. Prvním druhem je učení bez učitele (unsupervised learning), při tomto učení jsou neuronovým sítím poskytnuta data bez anotace, neboli bez očekávaného výsledku. Dalším druhem je učení se zpětnou vazbou, při kterém jsou opět trénovací data bez anotace, ale výsledku neuronové sítě je přiděleno skóre, které určuje, jak se výsledek přiblížil očekávanému výstupu.

Posledním zmíněným druhem je učení s učitelem. Při učení neuronových sítí s učitelem jsou neuronovým sítím poskytnuty anotovaná data. V případě Mask RCNN je zapotřebí na snímku označit všechny detekované objekty, jejich ohraničující rámečky a jejich segmentační masky. Takto označené snímky nazýváme jako trénovací datová sada. Pro ověření správnosti výstupů neuronové sítě je zapotřebí označit i část dat, která nebudou při trénování sítě poskytnuta, tato data označíme jako validační. V následujících částech této podkapitoly popíšu nástroje pro anotaci dat a pro trénování Mask RCNN. Dále se zaměřím na samotný proces učení Mask RCNN pro segmentaci metakarpálních kostí.

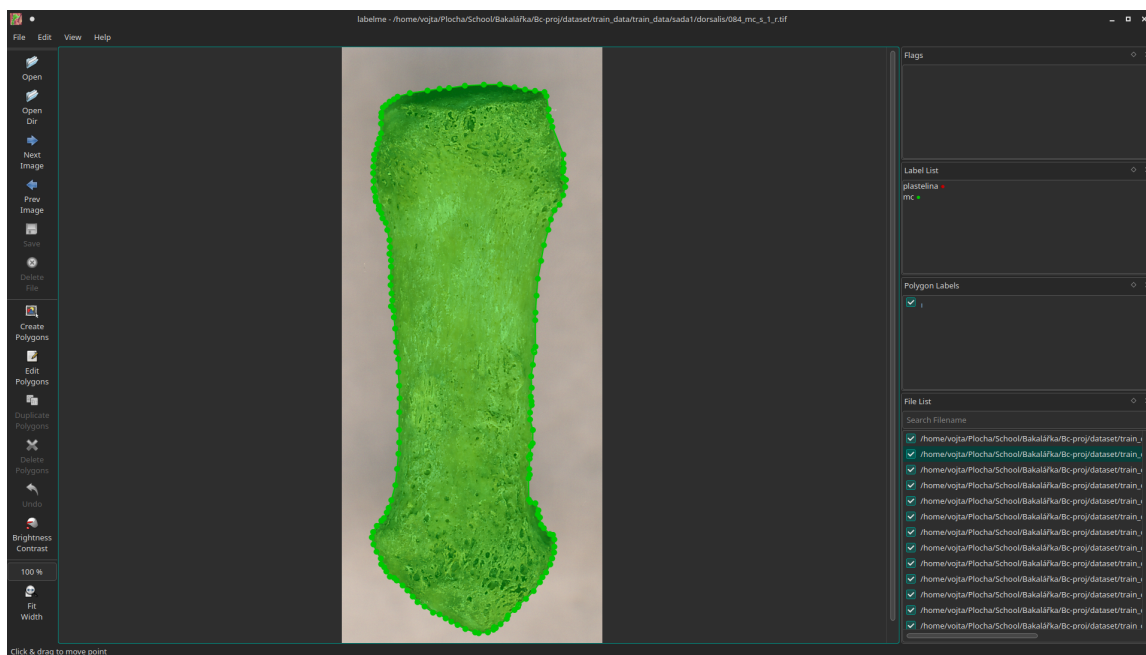
3.2.1 Nástroje

Pro anotaci dat existuje mnoho nástrojů a formátů, ve kterých jsou anotace uloženy. Při výběru nástroje pro anotaci dat jsem se zaměřil na projekty s otevřeným zdrojovým kódem (opensource), s přehledným a jednoduchým formátem výstupu a dále dle schopností anotovat snímky a vytvářet segmentační masky. Tyto parametry splňují například nástroje LabelMe, COCO Annotator, nebo CVAT. Pro účely této práce jsem si vybral a použil nástroj LabelMe[29].

V podkapitole 3.1.3 jsem se zabýval, jak Mask RCNN funguje, a zmiňoval jsem 2 implementace. Jednou implementací je Matterport Mask RCNN[1]. Tuto implementaci jsem původně chtěl využít pro natrénování modelu a následnou segmentaci. Avšak po prvotních testech jsem se rozhodl přejít na platformu Detectron2[31], kvůli kratšímu času na trénování a nižším nárokům na velikost úložiště. Matterport Mask RCNN při trénování ukládá po každé epoše natrénovaný model, který zabírá cca 200 megabytů, tudíž při vyšším počtu epoch uložené modely zabírají mnoho místa.

LabelMe

Nástroj LabelMe je volně dostupný v repozitáři na službě github[29], psaný v jazyce Python. Tento repozitář obsahuje mimo kódu také příklady využití a skripty pro převod na jiné formáty. Grafické prostředí nástroje LabelMe je velmi jednoduché (obrázek 3.3). Na pravé straně můžeme spravovat naše značky (Label List) a přepínat mezi soubory (File List). Kliknutím pravého tlačítka na obrázek vybereme, jaký druh anotace chceme vytvořit. Nástroj nám nabízí několik geometrických tvarů, jako je obdélník, kružnice, ale také jednotlivé body, úsečky a polygony. Po vybrání a vytvoření tvaru mu přidělíme značku určující jeho třídu, do které patří. Výsledná anotace je uložena pro každý snímek zvlášť ve formátu JSON, který uchovává mimo jiné relativní cestu ke snímku a vytvořené tvary s přidělenou značkou třídy. Pro převod tohoto formátu nalezneme v repozitáři dva skripty, jedním je labelme2voc.py pro převod do formátu VOC a druhým je labelme2coco.py, který převede anotace do formátu coco. Tento skript také konvertuje snímky do formátu JPEG.



Obrázek 3.3: Anotace metakarpální kosti nástrojem LabelMe.

Detectron2

Platforma Detectron2[31] obsahuje knihovnu, ve které nalezneme mnoho algoritmů pro detekci a segmentaci objektů. Mimo to knihovna nabízí “Model Zoo”, ve které nalezneme předtrénované modely, které lze upravit a využít pro vlastní specifické úlohy detekce a

segmentace. Díky této knihovně není potřeba vytvořit novou architekturu KNN, ale použít jednu z již vytvořených a naučit ji na vlastní datové sadě. Pro urychlení trénování modelu můžeme využít akceleraci pomocí grafické karty, ke které získáme přístup, například na Google Colaboratory[10].

3.2.2 Anotace dat

Pro anotaci dat jsem použil nástroj LabelMe popsany v sekci 3.2.1. Z datové sady bylo vybráno 208 obrázků, přibližně polovina z dorsálního pohledu a další část z radiálního pohledu. Pomocí nástroje byly jednotlivé kosti obtaženy polygonem a byla jim přidělena značka třídy. Tímto jsme vytvořili segmentační masku pro účely trénování. Aby bylo možné trénovat Mask RCNN na platformě Detectron2, je zapotřebí, aby každá anotace obsahovala i ohraničující rámeček, ten můžeme přidat v nástroji LabelMe ručně, nebo jej vytvořit později při načítání datové sady. Platforma Detectron2 neumí pracovat s anotacemi ve formátu LabelMe, toto se dá vyřešit dvěma různými způsoby. Prvním způsobem je vytvořit vlastní funkci, která načte takto anotovaná data a vytvoří rámeček (pokud jsme jej nevytvořili ručně). Pro účely této práce jsem zvolil možnost druhou, a to je převedení anotovaných dat z formátu LabelMe pomocí skriptu zmiňovaného v podkapitole 3.2.1 na formát coco. Tento skript vygeneruje chybějící rámečky, konvertuje snímky do formátu JPEG. Formát coco je uložen v jednom souboru JSON a jednotlivé anotace jsou od sebe odděleny názvem snímku. Platforma Detectron2 dokáže datové sady ve formátu coco načítat a pracovat, tudíž nemusíme nic upravovat.

3.2.3 Trénování modelu na vlastním datové sadě

Po vytvoření trénovací datové sady a jejího převedení do formátu coco, můžeme začít s přípravou modelu pro trénování. Nejprve je nutné mít správně nainstalovanou knihovnu Detectron2 pro jazyk Python. Po instalaci a úspěšném načtení knihovny můžeme začít s načtením trénovací datové sady. Načtené datové sadě přiřadíme název, cestu k souboru s anotacemi a cestu ke složce, kde se anotace nacházejí. Pro dosažení co nejpřesnějšího výsledku je třeba při učení neuronových sítí upravovat různé parametry. Pro získání co nejlepšího modelu je nutné trénování několikrát opakovat s různým nastavením parametrů. Tato úloha je velmi výpočetně náročná a pro rychlejší průběh je potřeba využít grafické karty (jedné, nebo i více). Pro přístup ke grafické kartě určené pro trénování neuronových sítí jsem využil výše zmíněnou službu Google Colaboratory[10].

Konfigurace modelu pro trénování

Nejprve je nutné načíst konfiguraci modelu z Model ZOO. Pro účely této práce jsem vybral konfiguraci modelu `mask_rcnn_R_101_FPN_3x.yaml`, protože tento model nabízí nejlepší kompromis mezi přesností a rychlostí. Tento konfigurační soubor obsahuje nastavení modelu Mask RCNN s využitím spojení ResNetu a FPN[31]. Po načtení konfiguračního souboru je třeba načíst váhy předtrénovaného modelu. Cesta k váhám je uložena v konfiguračním souboru.

Dále nastavíme do konfigurace jméno naší načtené datové sady a počet vláken, které budou data načítat. Vzhledem k malému počtu trénovacích dat jsem zmenšil výchozí hodnotu počtu vláken ze 4 na 2. Dále je potřeba upravit, kolik tříd budeme na snímcích detekovat a segmentovat, tuto hodnotu nastavíme na 2 třídy. První třídou je pozadí, toto není nijak anotované. A druhou třídou je metakarpální kost. Jako další parametr konfigurace je

potřeba nastavit, kolik regionů zájmu bude během trénování vytvořeno pro jeden obrázek. Jelikož kost na snímku zabírá velkou část prostoru, není nutné vytvářet mnoho regionů zájmu. Výchozí hodnota 512 byla snížena na 64.

Další parametr nám bude určovat rychlost učení tzv. learning rate. Nastavením tohoto parametru určujeme, jak moc jsou upraveny váhy neuronové sítě během učení. Tento parametr byl snížen na 0.00025. Vysoce ovlivňuje výsledek, například pokud je nastaven na příliš nízkou hodnotu, učení potrvá déle a může se zaseknout. Dalším parametrem nám určuje maximální počet iterací pro trénování. Tento parametr jsem nastavil na 20 000 iterací. A v neposlední řadě je potřeba nastavit, kolik obrázků bude během jedné iterace načteno. Tento parametr závisí na počtu grafických karet, ale také na velikosti paměti grafických karet. Shrnutí nastavení parametrů:

- Konfigurační soubor - **mask_rcnn_R_101_FPN_3x.yaml**
- Počet vláken pro načítání - **2**
- Počet tříd pro detekci - **2** (pozadí + metakarpální kost)
- Počet regionů zájmu (pro jeden obrázek) - **64**
- Learning rate - **0.00025**
- Počet iterací - **20000**
- “Velikost várky” (Batch size) - **2**

Trénování

Na základě konfigurace vytvoříme “trenéra” (trainer), který nejprve načte předtrénované váhy a trénovací datovou sadu a poté můžeme spustit trénování. Samotný proces trénování probíhá automaticky bez nutnosti zásahů. Trénování trvá od několika desítek minut až po několik hodin. Výsledný natrénovaný model je uložen a můžeme jej využít pro segmentaci. Zdrojový kód pro trénování se nachází ve složce *src/mask_rcnn*, jedná se hlavně o soubory *config.py* a *train.py*. Při tvorbě zdrojových kódů jsem vycházel z oficiálního návodu[5], jehož odkaz můžeme nalézt v oficiálním repozitáři projektu Detectron2[31].

3.3 Segmentace a měření délky metakarpálních kostí

V této podkapitole popíšeme algoritmus segmentace a měření metakarpálních kostí a jeho účinnost na datové sadě popsané v sekci 2.5. Tato část algoritmu byla rozdělena do dvou kroků, které budou podrobněji popsány v následujících podkapitolách.

3.3.1 Segmentace metakarpálních kostí

Pro segmentaci a extrakci kontury metakarpální kosti byl použit natrénovaný model popsaný v podkapitole 3.2. Nejprve je nutné upravit konfiguraci popsanou v sekci 3.2.3, tak aby se využil natrénovaný model pro segmentaci metakarpálních kostí. Platforma Detectron2 nabízí více možností jak pomocí natrénovaného modelu detekovat a segmentovat objekty na snímcích. Pro účely této práce nám stačí využít základní wrapper nazvaný DefaultPredictor s upravenou konfigurací. Objekt vytvořený pomocí DefaultPredictoru načte natrénovaný model.

Dále je nutné načíst snímky z datové sady. Snímky jsou roztrženy dle sad a pohledů, ze kterých jsou nasnímány, v podadresářích. Vytvořil jsem funkci `get_files`, která načte cesty všech souborů do pole a odfiltruje soubory, které nemají příponu `tif`. Poté stačí polem procházet a jednotlivé snímky načítat a zpracovat.

Pro načtení snímku jsem využil funkcionality z knihovny OpenCV. Data načteného snímku jsou následovně předány načtenému modelu Mask RCNN, který provede segmentaci. Výstupem segmentace je objekt, který obsahuje segmentační masku a ohraničující rámeček. Objekt obsahuje všechny detekované instance objektu. Segmentační maska jednotlivých objektů je dále načtena jako pole knihovny Numpy a uložena pomocí funkce `np.save` do souborů. Tyto soubory jsou pojmenovány dle názvu snímku, ze kterého byly vytvořeny a je jim přidána přípona `.NPY`, která popisuje formát, ve kterém jsou uloženy. Jedná se o binární formát, který ukládá jedno Numpy pole spolu s popisem datového typu.

Při segmentaci již není nutné používat akceleraci výpočtů na grafické kartě a můžeme využít běžné cpu. Avšak je nutné, aby architektura cpu byla podporována knihovnou Detectron2.

3.3.2 Zpracování dat segmentace

V předchozí podkapitole jsem popsal segmentaci snímků a extrakci kontury, získaná data jsou uložena v binárním formátu. Pro zpracování dat je použit skript `process_output.py` uložený ve složce `src`. Tento skript načítá uložené segmentační masky ze souborů, následně z nich vytvoří binarizovaný snímek, dále je extrahována kontura kosti a provedeno měření délky kosti. Výsledná data jsou uložena ve formátech JSON a TPS.

Pro veškeré výpočty a zpracování dat byly využity funkce z knihoven Numpy a OpenCV. V jazyce Python je sice možné vytvořit funkce, které by dosáhly stejného výsledku, avšak pro co nejrychlejší zpracování je vhodnější využít optimalizované funkce z knihoven, které jsou pro tento účel vytvořené.

Extrakce kontury a měření kosti

Pro načtení uložené segmentační masky jsem použil funkci `numpy.load`. Segmentační maska má stejný rozměr jako snímek, ze kterého byla vytvořena a je složena z booleovských hodnot. Souřadnice, na kterých jsou hodnoty pravdivé určují souřadnice pixelů, které náležejí metakarpální kosti. Pomocí funkcí Numpy vytvoříme dvourozměrnou matici, která odpovídá výšce a šířce snímku kosti, tato matice bude představovat náš binarizovaný obraz. Dále s pomocí knihovny Numpy naplníme tuto matici hodnotami 255 (odpovídá černé barvě) na souřadnicích, kde jsou hodnoty segmentační masky pravdivé. Vytvoření binarizovaného obrazu je provedeno ve funkci `bin_img` v souboru `process_output.py`.

Z binarizovaného obrazu extrahujeme konturu metakarpální kosti pomocí funkce z knihovny OpenCV. Kontura je obvod kosti, tedy pixely, které hraničí s pozadím snímku. Funkce `findContours` z knihovny OpenCV nabízí také možnost konturu aproximovat. Aproximovaná kontura neukládá každý bod ohraničující objekt, ale u bodů, které se nacházejí v přímce, ukládá pouze počáteční a konečný bod. Aproximovaná kontura nám značně snižuje počet bodů v kontuře, a tudíž i paměťové nároky. Aproximace kontury se spíše využívá pro jednoduché geometrické tvary jako jsou například obdélníky. Avšak tvar kosti je nepravidelný a spíše zaoblený, z tohoto důvodu jsem se rozhodl aproximaci kontury nevyužít. Kontura je extrahována spolu s výpočtem délky kosti ve funkci `measure` v souboru `process_output.py`.

Na obrázcích 3.4 a 3.5 vidíme různé rozměry, které se dají na metakarpálních kostech měřit. Pro změření délky kosti je nutné vypočítat vzdálenost mezi body 1 a 5 na obrázku

3.4 a body 1 a 2 na obrázku 3.5. Tyto body se nacházejí na nejvyšší a nejnižší položených bodech kontury. Na tuto skutečnost se můžeme v datové sadě poskytnuté Antropologickým ústavem spolehnout, jelikož kosti byly pečlivě skenovány tak, aby na snímcích byly zřetelné všechny měřitelné délky a šířky.



Obrázek 3.4: Metakarpální kost z radiálního směru s vyznačenými rozměry.[18] Obrázek 3.5: Metakarpální kost z dorsálního směru s vyznačenými rozměry.[18]

S využitím funkcí knihovny *Numpy argmin* a *argmax* určíme nejvyšší a nejnižší bod kontury. Dále vypočítáme vzdálenost stejným způsobem, který byl popsán v podkapitole 2.6.2 s využitím rovnice 2.3 pro výpočet vzdálenosti a rovnice 2.4 pro převod vzdálenosti na centimetry. Pro převod vzdálenosti v pixelech potřebujeme zjistit měřítko, neboli kolik pixelů odpovídá jedné jednotce. Využijeme knihovnu *Pillow*, která obsahuje funkce načítající metadata ze souborů snímků. Z těchto metadat můžeme zjistit měřítko, ale také jednotku, které měřítko odpovídá. Formát snímků *.tif* může obsahovat 2 druhy jednotek. Prvním druhem jsou centimetry a druhým palce, ale také obsahuje možnost, že snímek nemá žádnou absolutní jednotku míry.

Pro účely této práce jsem se rozhodl využít metrický systém a vzdálenosti budou vypočítány a uloženy v centimetrech. Pokud snímek bude obsahovat měřítko v jednotce palců, bude tato vzdálenost převedena na centimetry, vynásobením konstantou 2.54 (1 in = 2.54 cm). Pokud snímek nebude obsahovat žádnou absolutní jednotku míry, pak bude vzdálenost ponechána v pixelech. Při ukládání dat bude uložena i informace, zda-li je vzdálenost v centimetrech, nebo pixelech.

Uložení výsledných dat

Po konzultaci s Antropologickým ústavem bylo domluveno, že extrahované kontury kostí budou uloženy v souborech ve formátu TPS. Datová sada je rozdělena na 3 podadresáře, každý podadresář je dále rozdělen dle směru, ze kterého byly kosti nasnímány (dorsalis a radialis), na další dva podadresáře tyto podadresáře obsahují snímky kostí.

Pro každý podadresář, který obsahuje snímky kostí, je vytvořen jeden soubor ve formátu TPS pojmenován dle názvů adresářů, ve kterých se kosti vyskytují. Data jsou v souboru ve

formátu TPS uložena v textové podobě a jednotlivé snímky jsou od sebe odděleny názvem souboru. Pro práci se soubory ve formátu TPS byla použita knihovna *py_tps*. Do tohoto souboru jsou zapsány jednotlivé body kontury a název snímku.

Jako druhý formát pro zpracovaná data jsem zvolil formát JSON, který je více používaný a pro většinu programovacích jazyků existují knihovny, které jej umí načítat. Do tohoto formátu je možné uložit více dat, která jsou pojmenována, aby bylo možné určit jejich význam. Data jsou stejně jako v případě předchozího formátu ukládána v textové podobě, což zvyšuje požadované místo pro jejich uložení. Data jsou v souboru uložena jako pole objektů, kde každý objekt reprezentuje jeden snímek kosti. V jednotlivých objektech je uložen název snímku pod klíčem **Image**. Souřadnice vrcholného a spodního bodu jsou uloženy jako pole dvou hodnot, pod klíči **TopPoint** a **BottomPoint**. Změřená a případně převedená délka kosti je uložena pod klíčem **Length**, jednotka délky je uložena pod klíčem **LengthUnit**. V neposlední řadě je uložena detekovaná kontura kosti pod klíčem **Contour** v dvourozměrném poli.

3.3.3 Testování a shrnutí algoritmu pro detekci, extrakci a měření délky zápěstních kostí s využitím Mask RCNN

Při vývoji a testování výše popsaného algoritmu jsem došel k následujícímu závěru. Po otestování na poskytnutém datasetu byly správně detekovány všechny metakarpální kosti na snímcích, extrahovaná kontura kosti však nebyla příliš přesná. Od přesnosti extrahované kontury se odvíjí i následná přesnost měření délky metakarpálních kostí. Rychlost algoritmu při využití grafické karty pro detekci a vytvoření segmentační masky je dostatečně rychlá i pro zpracování většího počtu snímků. Při využití pouze procesoru je rychlost dostačující pro menší počet snímků. Díky mezikroku uložení segmentačních masek je možné segmentaci většího množství dat na procesoru provést v čase, kdy nebudeme počítač využívat a s uloženými segmentačními maskami poté můžeme pracovat rychleji. Testování probíhalo v prostředí Google Colaboratory, při testovacím sezení mi byly přiděleny následující hardwarové prostředky:

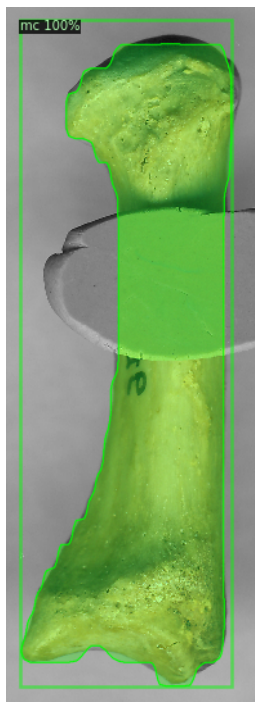
- RAM - **12GB**
- Disk - **150GB**
- Procesor - Intel® Xeon® CPU @ 2.20GHz
- Grafická karta - NVidia Tesla T4 15GB

Trénování neuronové sítě zabralo cca **3** hodiny času při konfiguraci popsané v podkapitole **3.2.3**, s využitím grafické karty. Pro segmentaci a uložení segmentační masky jsem otestoval obě možnosti - s a bez využití grafické karty. Při využití grafické karty načtení jednoho snímku, segmentace a uložení masky zabralo přibližně **0.132** sekund. Při využití jen procesoru zpracování jednoho snímku zabralo v průměru **5.09** sekund.

Přesnost segmentace metakarpální kosti můžeme pozorovat vlastním okem, pokud použijeme funkci *segment* ze souboru *process_output.py*, která načte originální obrázek a na základě segmentační masky odstraní pozadí a obrázek opět uloží (na specifikované místo). Vytvořené segmentační masky nejsou dokonale přesné, v podkapitole **3.4** se budu více věnovat tomu, jak je možné přesnost zvýšit. Od přesnosti segmentace se také odvíjí přesnost měření. Měření provedená algoritmem byla v průměru o **0.02** cm kratší. Avšak ručně naměřené hodnoty byly poskytnuty pouze u 1. sady snímků nasnímaných ze směru dorsalis.

U snímků, na kterých jsou kosti nasnímány ze směru radialis, jsou segmentační masky viditelně méně přesné. Pro výpočet průměrného rozdílu délek naměřených algoritmem oproti ručnímu měření jsem vytvořil skript *measurement.py*.

Výhodou využití Mask RCNN pro segmentaci je generování segmentačních masek pro kosti překryté modelovací hmotou. Masky překrytých kostí předpovídají, jak by kost vypadala, kdyby překryta nebyla, tudíž i s překrytou částí můžeme pracovat. Na obrázku 3.6 můžeme pozorovat, jak taková maska vypadá na překryté kosti. Další výhodou je použitelnost natrénovaného modelu bez nutnosti grafické karty. Čas potřebný pro segmentaci je však několikanásobně vyšší.



Obrázek 3.6: Vizualizace masky vytvořené natrénovaným modelem na snímku kosti překryté modelovací hmotou.

Další částí algoritmu je zpracování segmentačních masek a změření délky kostí. Díky využití funkcí z knihoven OpenCV a numpy, samotná úprava masek, extrakce kontury a měření probíhají rychle (cca **0.005** sekund na jeden snímek). Nejvíce času zabírá uložení výsledků do souborů typu JSON a TPS (cca **23** vteřin), jelikož extrahovaná kontura obsahuje mnoho souřadnic bodů. Pro rychlejší a úspornější možnost uložení kontur je možné ukládat konturu do souborů ve formátu NPY. Ukládání bodů kontury v tomto formátu jsem neimplementoval.

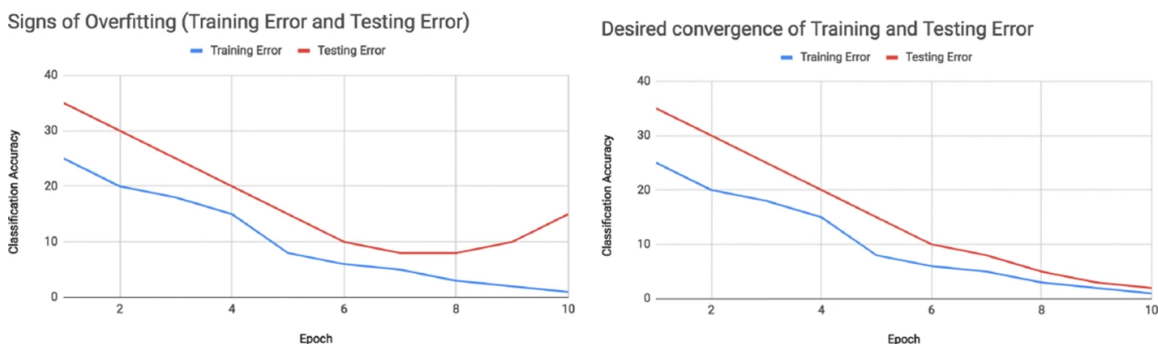
3.4 Možná rozšíření algoritmu

V této podkapitole se budu věnovat možnostem, jak by se dal výše popsáný algoritmus vylepšit, nebo rozšířit. V první části se zaměřím na to, jak by se dala vylepšit přesnost segmentace kostí a jak by se dal algoritmus upravit, aby bylo možné zpracovat více instancí detekovaných objektů.

3.4.1 Zvýšení přesnosti segmentace

Základem celého algoritmu je správná detekce a vytvoření segmentační masky. Detekce kostí na poskytnuté datové sadě byla bezchybná. Avšak segmentační masky nebyly příliš přesné. Pro zvýšení přesnosti segmentačních masek je nutné se vrátit k trénování modelu konvoluční neuronové sítě Mask RCNN, případně zvolit jinou metodu segmentace kostí. Přesnost segmentace s využitím Mask RCNN se dá vylepšit zvětšením trénovací datové sady. Pro vytvoření trénovací datové sady bylo použito 208 snímků.

Při trénování neuronových sítí s malou trénovací datovou sadou může dojít k přetrénování neuronové sítě. Přetrénovaný model dokáže detekovat a vytvářet velmi přesné segmentační masky pro snímky, které se nacházejí v trénovací datové sadě. Avšak ztratí schopnost generalizace a na snímcích, které nejsou součástí trénovací datové sady, nejsou výsledné masky přesné. [23] Abychom zabránili přetrénování neuronové sítě, je nutné zvětšit trénovací datovou sadu a zavést validační datovou sadu, na které budeme v průběhu trénování testovat, zda-li nedošlo k přetrénování (viz obr. 3.7). Přetrénování neuronových sítí se dá předejít i jinými způsoby, například dropout (vyřazení náhodně vybraného neuronu), nebo třeba využití modelů předtrénovaných na velkém vzorku dat.[23]



Obrázek 3.7: Přetrénovaný model vlevo. Požadovaný průběh trénování vpravo.[23]

Zvětšení trénovací datové sady je možné provést dvěma způsoby. Prvním způsobem je vybráním většího vzorku snímků z datové sady a provedení anotace na těchto datech. Toto je časově velmi náročné. Dalším způsobem, jak trénovací datovou sadu rozšířit, je využít augmentace snímků. Jedná se o umělé zvětšení trénovací datové sady za pomoci například geometrických transformací (rotace, posun, ...), tvorba šumu, použití filtrů, nebo náhodným vymazáním části snímku.[23]. Knihovna *Detectron2* obsahuje funkce, které nám umožňují rozšířit dataset pomocí augmentací, nebo vytvářet vlastní funkce, které augmentaci provádějí. Nebo je možné využít dalších knihoven, jako například *Imgaug*. Při augmentaci snímků je nutné, aby provedené transformace snímků byly provedeny i na anotacích snímků.

Dalším způsobem, kterým by se zvýšila přesnost detekce a segmentace kostí, je rozdělení kostí do tříd podle pohledu, ze kterého jsou nasnímany. Metakarpální kosti nasnímané ze stejného pohledu mají podobnější tvar, i když se jedná o jinou kost (např. palcová záprstní kost a záprstní kost prostředníku).

3.4.2 Segmentace několika instancí kostí najednou

Délky metakarpálních kostí se pohybují přibližně mezi 4 cm a 7 cm a jejich šířka mezi 1 cm a 2 cm.[18] Snímky kostí v datové sadě jsou pořízeny na skeneru dokumentů. Na plochu skeneru je možné umístit více metakarpálních kostí. Díky využití Mask RCNN pro detekci

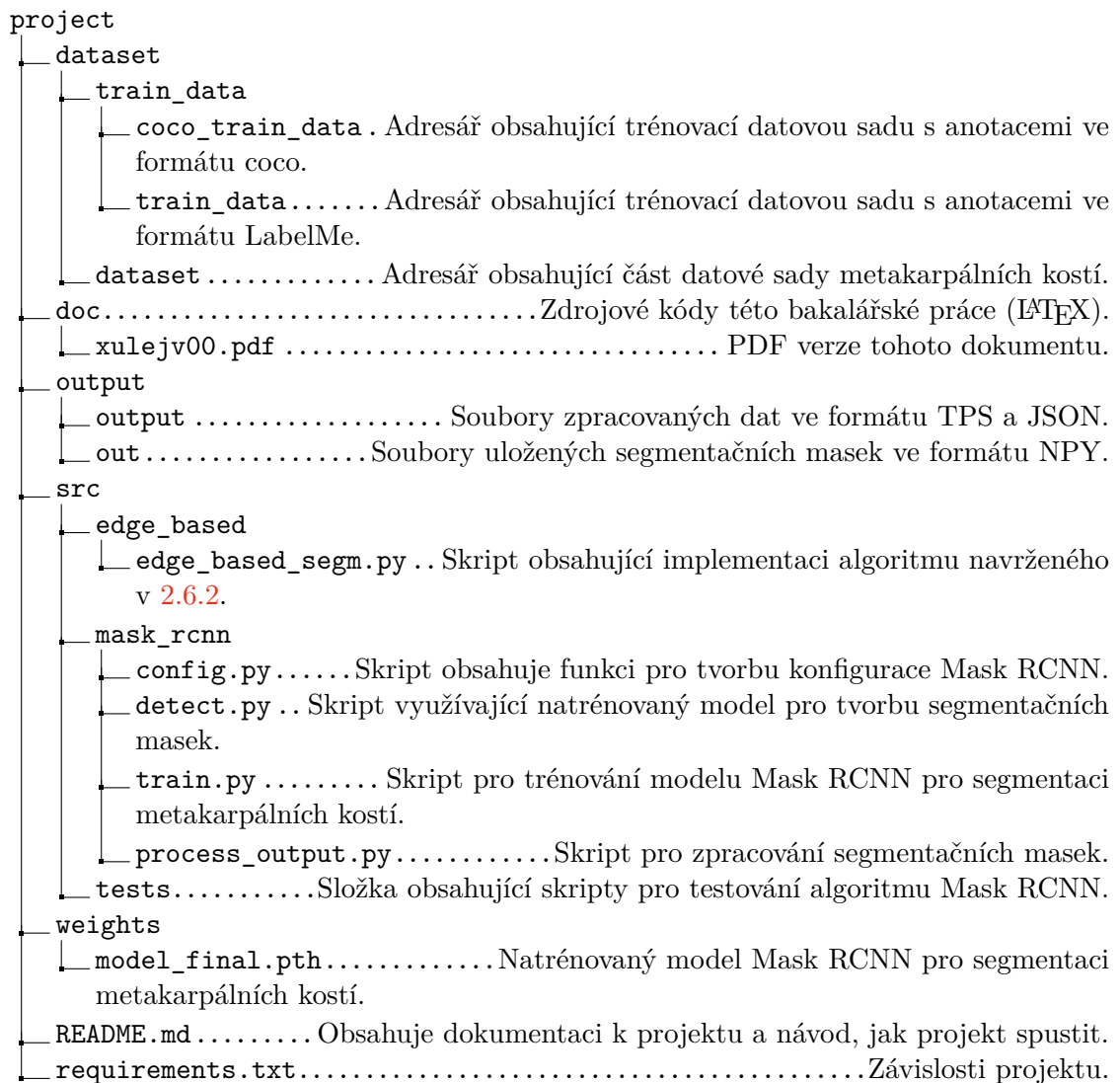
a segmentaci kostí je možné detekovat více instancí objektu na jednom snímku. V nynější podobě algoritmus počítá pouze s jedním výskytem metakarpální kosti na snímku.

Při zpracování snímku, který by obsahoval více instancí metakarpálních kostí, by bylo potřeba upravit algoritmus tak, aby segmentační masky pro každou instanci metakarpální kosti byly ukládány do zvláštního souboru označeného názvem snímku a identifikačním číslem (od 1 do x, pro x detekovaných instancí). Dále by bylo potřeba upravit zpracování segmentačních masek a ukládání výsledných dat.

Výsledná data jsou ukládána do souborů ve formátu JSON a TPS a rozdělena dle názvu snímku. Pro soubory ve formátu JSON by bylo možné přidat klíč **ID**, do kterého by se ukládalo identifikační číslo detekované instance objektu. Podobně by bylo možné upravit ukládání dat pro soubory ve formátu TPS, který umožňuje vkládat k jednotlivým záznamům kromě bodů a názvu souboru také identifikační číslo záznamu.

3.5 Struktura projektu

Implementace projektu má následující adresářovou strukturu:



3.5.1 Instalace závislostí

Projekt je napsán v jazyce Python3 a byl testován na Linuxovém operačním systému. Závislosti jsou uvedeny v souboru *requirements.txt*, kromě knihovny Detectron2. Pro instalaci této knihovny postupujte dle návodu v repozitáři.[31] Pro instalaci ostatních závislostí stačí spustit následující příkaz.

```
pip3 install -r requirements.txt
```

3.5.2 Spuštění skriptu pro trénování

Skript pro trénování *train.py* má následující argumenty:

- `-help` - vypíše nápovědu
- `-t` - cesta k trénovacímu dataset s anotacemi coco

- -l - learning rate
- -o - cesta k adresáři, do kterého bude uložen výsledný model
- -i - počet iterací

Příklad spuštění trénování Mask RCNN ze složky *projekt/src/mask_rcnn*:

```
python3 train.py -t /path/coco\_train\_data -o /path/logs
```

3.5.3 Spuštění skriptu pro detekci a segmentaci

Skript pro vytvoření a uložení segmentačních masek *detect.py* má následující argumenty:

- -help - vypíše nápovědu
- -t - práh pro detekci
- -o - cesta k adresáři, do kterého budou uloženy soubory se segmentačními maskami
- -i - cesta k adresáři obsahující snímky metakarpálních kostí

Příklad spuštění segmentace snímků:

```
python3 detect.py -o /path/out -i /path/dataset -m /path/model_final.pth
```

3.5.4 Spuštění skriptu pro zpracování segmentačních masek

Skript pro zpracování segmentačních masek *process_output.py* má následující argumenty:

- -help - vypíše nápovědu
- -d - cesta k adresáři obsahující soubory segmentačních masek
- -o - cesta k adresáři, do kterého budou uloženy soubory s extrahovanou konturou a délkou kostí
- -i - cesta k adresáři obsahující snímky metakarpálních kostí

Příklad spuštění zpracování souborů se segmentačními maskami:

```
python3 process\_output.py -o /path/out -i /path/dataset -d /path/output
```

Kapitola 4

Závěr

Cílem této bakalářské práce bylo prostudovat literaturu zabývající se detekcí a segmentací objektů, hlavně metakarpálních kostí, ze snímků. Na základě získaných poznatků navrhnout a implementovat algoritmus pro detekci kosti na snímku a následnou extrakci kontury kosti a změření její délky. Implementovaný algoritmus otestovat na datové sadě snímků metakarpálních kostí poskytnuté Antropologickým ústavem PŘF MU.

V teoretické části textu jsem se věnoval historickému vývoji metod pro detekci objektů na snímcích, dále jsem shrnul anatomický popis metakarpálních kostí a metody zabývající se zpracováním snímků kostí. Krátce jsem popsal nástroje vhodné pro zpracování obrazu. V posledních podkapitolách teoretické části práce jsem se věnoval návrhu řešení založeného na detekci hran v obraze.

Navrhovaný algoritmus s využitím detekce hran v obraze nebyl příliš spolehlivý, hlavně při zpracování snímků, na kterých byly hrany kostí rozmazané. V další části bakalářské práce jsem se proto zaměřil na tvorbu algoritmu, který pro detekci kostí a segmentaci obrazu využívá modernější přístup založený na konvolučních neuronových sítích. Pro implementaci tohoto algoritmu jsem si vybral neuronovou síť Mask RCNN implementované v knihovně Detectron2. V práci se věnuji tvorbě trénovací datové sady a učení modelu pro detekci a segmentaci metakarpálních kostí. Při testování tohoto řešení jsem zjistil, že algoritmus správně detekoval kosti na snímcích, avšak segmentační masky nebyly příliš přesné. V další části této práce jsem se proto zaměřil na různé způsoby, jak by se dala přesnost segmentace vylepšit, například využitím augmentace snímků trénovací datové sady. Do budoucna bych se zaměřil na otestování jiných architektur neuronových sítí pro detekci a segmentaci metakarpálních kostí. Jako další možnost rozšíření přichází v úvahu úprava stávajícího algoritmu tak, aby bylo možné zpracovávat více kostí na jednom snímku.

Literatura

- [1] ABDULLA, W. *Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow* [https://github.com/matterport/Mask_RCNN]. Github, 2017 [cit. 2022-01-10].
- [2] CHALECHALE, A., BAHRI, A. a VATANCHIAN, M. Vision-based bone image recognition using geometric properties. *Iranian Journal of Science & Technology, Transaction B: Engineering*. Prosinec 2010, sv. 34, s. 597–604.
- [3] CLARK, A. *Pillow (PIL Fork) Documentation*. readthedocs, 2015. Dostupné z: <https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf>.
- [4] COOTES, T. F., TAYLOR, C. J., COOPER, D. H. a GRAHAM, J. Active shape models-their training and application. *Computer vision and image understanding*. Academic Press. 1995, sv. 61, č. 1, s. 38–59.
- [5] *Detectron2 Beginner's Tutorial: Google Colaboratory*. 2021. Dostupné z: https://colab.research.google.com/drive/16jcaJoc6bCFAQ96jDe2HwtXj7BMD_-m5.
- [6] DROZDOVÁ, E. a MALINA, J. *Základy osteometrie*. 1. vyd. Brno: Nadace Universitas Masarykiana, 2005. ISBN 80-7204-291-2.
- [7] EDWARDS, G., TAYLOR, C. a COOTES, T. Interpreting face images using active appearance models. In: Květen 1998, s. 300 – 305. DOI: 10.1109/AFGR.1998.670965. ISBN 0-8186-8344-9.
- [8] ELIOPOULOS, C., LAGIA, A. a MANOLIS, S. A modern, documented human skeletal collection from Greece. *Homo : internationale Zeitschrift für die vergleichende Forschung am Menschen*. Únor 2007, sv. 58, s. 221–8. DOI: 10.1016/j.jchb.2006.10.003.
- [9] GARVIN, H. M. a STOCK, M. K. The Utility of Advanced Imaging in Forensic Anthropology. *Academic Forensic Pathology*. 2016, sv. 6, č. 3, s. 499–516. DOI: 10.23907/2016.050. Dostupné z: <https://doi.org/10.23907/2016.050>.
- [10] *Google Colaboratory: FAQ*. 2021. Dostupné z: <https://research.google.com/colaboratory/faq.html>.
- [11] HAFIZ, A. M. a BHAT, G. M. A survey on instance segmentation: state of the art. *International journal of multimedia information retrieval*. Springer. 2020, sv. 9, č. 3, s. 171–189.

- [12] HARRIS, C. R., MILLMAN, K. J., WALT, S. J. van der, GOMMERS, R., VIRTANEN, P. et al. Array programming with NumPy. *Nature*. Springer Science and Business Media LLC. září 2020, sv. 585, č. 7825, s. 357–362. DOI: 10.1038/s41586-020-2649-2. Dostupné z: <https://doi.org/10.1038/s41586-020-2649-2>.
- [13] HE, K., GKIOXARI, G., DOLLÁR, P. a GIRSHICK, R. *Mask R-CNN*. arXiv, 2017. DOI: 10.48550/ARXIV.1703.06870. Dostupné z: <https://arxiv.org/abs/1703.06870>.
- [14] HORÁČKOVÁ, L. a MALINA, J. *Panoráma antropologie biologické - sociální - kulturní: modulové učební texty pro studenty antropologie a příbuzných oborů*. Brno: Nadace Universitas, 2007. ISBN 978-80-7204-558-7.
- [15] KIRSCHNER, M. a WESARG, S. Active Shape Models Unleashed. *Progress in Biomedical Optics and Imaging - Proceedings of SPIE*. Březen 2011, sv. 7962. DOI: 10.1117/12.877690.
- [16] KRIZHEVSKY, A., SUTSKEVER, I. a HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*. ACM New York, NY, USA. 2017, sv. 60, č. 6, s. 84–90.
- [17] *OpenCV: About*. 2021. Dostupné z: <https://opencv.org/about/>.
- [18] PRAŽÁK, O. *Metakarpální vzorec u člověka: sexuální dimorfismus a stranová asymetrie [online]*. 2014 [cit. 2022-04-19]. Diplomová práce. Masarykova univerzita, Přírodovědecká fakulta, Brno. SUPERVISOR : Miroslav Králík. Dostupné z: <https://is.muni.cz/th/srjz8/>.
- [19] *Python: About*. Python Software Foundation, 2021. Dostupné z: <https://www.python.org/about/>.
- [20] REDDY, N. *A Survey on Specialised Hardware for Machine Learning*. Červenec 2019. DOI: 10.13140/RG.2.2.20697.26725.
- [21] RONNEBERGER, O., FISCHER, P. a BROX, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv, 2015. DOI: 10.48550/ARXIV.1505.04597. Dostupné z: <https://arxiv.org/abs/1505.04597>.
- [22] RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S. et al. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*. 2015, sv. 115, č. 3, s. 211–252. DOI: 10.1007/s11263-015-0816-y.
- [23] SHORTEN, C. a KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of big data*. Springer. 2019, sv. 6, č. 1, s. 1–48.
- [24] STEGMANN, M. B. *Generative Interpretation of Medical Images*. Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2004. 248 s. Disertační práce. Informatics and Mathematical Modelling, Technical University of Denmark, DTU. Awarded the Nordic Award for the Best Ph.D. Thesis in Image Analysis and Pattern Recognition in the years 2003-2004 at SCIA'05. Dostupné z: <http://www2.compute.dtu.dk/pubdb/pubs/3126-full.html>.

- [25] VARATHARASAN, V., SHIN, H.-S., TSOURDOS, A. a COLOSIMO, N. Improving Learning Effectiveness For Object Detection and Classification in Cluttered Backgrounds. In: Listopad 2019, s. 78–85. DOI: 10.1109/REDUAS47371.2019.8999695.
- [26] VIOLA, P. a JONES, M. Rapid Object Detection using a Boosted Cascade of Simple Features. In: únor 2001, sv. 1, s. I–511. DOI: 10.1109/CVPR.2001.990517. ISBN 0-7695-1272-0.
- [27] VIOLA, P. a JONES, M. Robust Real-Time Face Detection. *International Journal of Computer Vision*. Květen 2004, sv. 57, s. 137–154. DOI: 10.1023/B:VISI.0000013087.49260.fb.
- [28] VIRTANEN, P., GOMMERS, R., OLIPHANT, T. E., HABERLAND, M., REDDY, T. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. 2020, sv. 17, s. 261–272. DOI: 10.1038/s41592-019-0686-2.
- [29] WADA, K. *labelme: Image Polygonal Annotation with Python* [<https://github.com/wkentaro/labelme>]. 2016.
- [30] WALT, S. van der, SCHÖNBERGER, J. L., NUNEZ-IGLESIAS, J., BOULOGNE, F., WARNER, J. D. et al. Scikit-image: image processing in Python. *PeerJ*. Červen 2014, sv. 2, s. e453. DOI: 10.7717/peerj.453. ISSN 2167-8359. Dostupné z: <https://doi.org/10.7717/peerj.453>.
- [31] WU, Y., KIRILLOV, A., MASSA, F., LO, W.-Y. a GIRSHICK, R. *Detectron2* [<https://github.com/facebookresearch/detectron2>]. 2019.
- [32] ZOU, Z., SHI, Z., GUO, Y. a YE, J. *Object Detection in 20 Years: A Survey*. 2019. Dostupné z: <https://arxiv.org/abs/1905.05055v2>.