



Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra kybernetiky

---

**Kalibrace kamery dle známého předmětu pro úlohu měření  
trajektorií během chirurgických operačních výkonů**  
Camera calibration according to a known object for the task  
of trajectory measurement during surgical operations

---

# BAKALÁŘSKÁ PRÁCE

Autor:  
Jiří Moravec

Vedoucí práce:  
Ing. Miroslav Jířík Ph.D.

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd  
Akademický rok: 2021/2022

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jiří MORAVEC**  
Osobní číslo: **A18B0457P**  
Studijní program: **B3902 Inženýrská informatika**  
Studijní obor: **Inteligentní komunikace člověk – stroj**  
Téma práce: **Kalibrace kamery dle známého předmětu pro úlohu měření trajektorií během chirurgických operačních výkonů**  
Zadávající katedra: **Katedra kybernetiky**

## Zásady pro vypracování

1. Seznamte se s problematikou kalibrace kamery a detekce objektů v obraze.
2. Navrhněte a implementujte metodu pro kalibraci kamery dle známého předmětu ve scéně s chirurgickým operačním výkonem.
3. Ověřte funkci navržené metody prostřednictvím experimentu a vhodným způsobem vyhodnoťte její funkčnost.
4. Pomocí vhodného experimentu navrženou metodu vyhodnoťte.

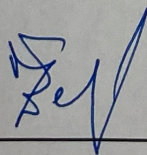
Rozsah bakalářské práce: **30 – 40 stránek A4**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

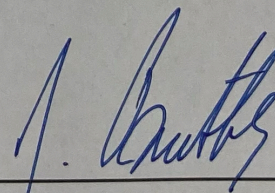
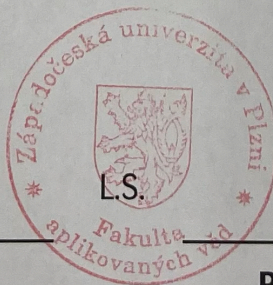
- Sonka, M., Hlavac, V., & Boyle, R. (2014). Image Processing, Analysis, and Machine Vision Second Edition. Thomson-Engineering.
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, 24(6), 381–395. <https://doi.org/10.1145/358669.358692>
- Hruďa, L., Dvořák, J., Váša, L.: On evaluating consensus in RANSAC surface registration, Computer Graphics Forum, Vol. 37(5), 2019.
- Wu, X., Sahoo, D., & Hoi, S. C. H. (2020). Recent advances in deep learning for object detection. Neurocomputing, 396, 39–64. <https://doi.org/10.1016/j.neucom.2020.01.085>
- Liška, V., Baxa, J., Beneš, J., Brůha, J., Ferda, J., Hošek, P., Jansová, M., Jiřík, M., Jonášová, A., Králíčková, M., Křečková, J., Křen, J., Lobovský, L., Lukeš, V., Mírka, H., Pálek, R., Pešta, M., Pitule, P., Rohan, E., ... Vyčítal, O. (2016). Experimental surgery (V. Liška (ed.); 1st ed.). NAVA, s.r.o.
- Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into Deep Learning.

Vedoucí bakalářské práce: **Ing. Miroslav Jiřík, Ph.D.**  
Výzkumný program 1

Datum zadání bakalářské práce: **15. října 2021**  
Termín odevzdání bakalářské práce: **23. května 2022**



**Doc. Ing. Miloš Železný, Ph.D.**  
děkan



**Prof. Ing. Josef Psutka, CSc.**  
vedoucí katedry

## Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 20. 5. 2022

.....  
Jiří Moravec

## Poděkování

Chtěl bych poděkovat panu doktoru Ing. Miroslavu Jiříkovi, Ph.D. za jeho podporu, cenné rady a trpělivost. Také i za možnost zpracovávat toto téma, které bylo mou první větší prací v oboru počítačového vidění. Za možnost použít jejich výpočetní kapacity děkuji MetaCentru. Dále bych chtěl poděkovat své rodině za podporu během studia.

Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

## **Anotace**

Tato práce se zabývá novými metodami kalibrace kamery dle známého předmětu pro úlohu měření trajektorií během chirurgických operačních výkonů. Za tímto účelem byly využity konvoluční neuronové sítě, hlavně konvoluční neuronová síť Mask R-CNN, i konvenční metoda počítačového vidění, kterou byla Houghova transformace. Byly navrženy nové metody, jejichž cílem bylo zpřesnit dosud používané řešení, které nebylo spolehlivé. V závěru práce byly nové metody porovnány mezi sebou i se stávajícími metodami s uspokojujícími výsledky.

## **Klíčová slova**

neuronové sítě, konvoluční neuronové sítě, detectron, kalibrace, kamera, operace, mask r-cnn, měření, umělá inteligence, počítačové vidění, výuka, zoom, coco, meta-centrum

## **Abstract**

This work deals with new methods of camera calibration according to the known subject for the task of trajectory measurement during surgical operations. For this purpose, convolutional neural networks, mainly the convolutional neural network Mask R-CNN, and the conventional method of computer vision, which was the Hough transform, were used. New methods were proposed to refine the solution used so far, which was not reliable. At the end of the work, the new methods were compared with each other and with existing methods with satisfactory results.

## **Keywords**

neural networks, convolutional neural networks, detectron, calibration, camera, operation, mask r-cnn, measurement, artificial intelligence, computer vision, teaching, zoom, coco, metacentrum

# Obsah

<b>1</b>	<b>Počítačová podpora v chirurgii a medicíně</b>	<b>1</b>
1.1	Lékařská informatika . . . . .	1
1.1.1	Aplikace lékařské informatiky . . . . .	1
1.2	Použití kamery v chirurgii . . . . .	3
1.2.1	Tradiční chirurgie . . . . .	3
1.2.2	Robotické a laparoskopické operace . . . . .	4
1.3	Digitální obraz . . . . .	5
1.3.1	Záznam digitálního obrazu . . . . .	6
1.3.2	Digitální zpracování obrazu . . . . .	7
1.4	Počítačové vidění . . . . .	9
1.4.1	Metody počítačového vidění k analýze obrazu . . . . .	10
1.4.2	Aplikace počítačového vidění . . . . .	10
1.4.3	Kalibrace kamery v počítačovém vidění . . . . .	11
1.5	Umělé neuronové sítě . . . . .	12
1.5.1	Perceptron . . . . .	13
1.5.2	Aktivační funkce . . . . .	14
1.5.3	Struktura umělých neuronových sítí . . . . .	17
1.5.4	Učení neuronových sítí . . . . .	19
1.5.5	Konvoluční neuronové sítě . . . . .	20
1.6	Podobné řešení dalších problémů v praxi . . . . .	21
<b>2</b>	<b>Návrh systému pro kalibraci</b>	<b>23</b>
2.1	Představení problému . . . . .	23
2.2	Příprava dat . . . . .	24
2.2.1	Typické video . . . . .	24
2.2.2	Anotace . . . . .	27
2.2.3	Soubor COCO . . . . .	29
2.3	Trénování neuronové sítě . . . . .	30
2.3.1	Mask R-CNN . . . . .	30
2.3.2	Metacentrum . . . . .	34
2.4	Navržený kalibrační algoritmus . . . . .	37
2.4.1	Měření stran desky . . . . .	37
2.4.2	Měření poloměru kruhových výřezů na desce . . . . .	39
2.5	Výsledky experimentů . . . . .	43
2.5.1	Přesnost a rychlost trénování neuronové sítě . . . . .	43
2.5.2	Porovnání kalibračních metod . . . . .	44



2.6	Diskuze výsledků . . . . .	45
<b>3</b>	<b>Závěr</b>	<b>50</b>
<b>4</b>	<b>Přílohy</b>	<b>51</b>
4.1	Obrázky . . . . .	51
4.2	Tabulky . . . . .	53

# 1 Počítačová podpora v chirurgii a medicíně

## 1.1 Lékařská informatika

Lékařská (nebo také zdravotnická) informatika je mezidisciplinární obor, který se zabývá využitím výpočetní techniky k organizaci, analýze, správě a využívání informací pro zlepšování zdravotní péče. Podle [1] je cílem je vypracovat standardy, formalizovat lékařskou terminologii, zvýšit použitelnost zdravotnických informačních systémů pro usnadnění komunikace a řízení toku informací od lékařů k pacientům. Lékařská informatika zahrnuje teoretické a praktické aspekty zpracování informací a komunikace založené na znalostech a zkušenostech odvozených z procesů v medicíně a zdravotnictví.

Konečným cílem lékařské informatiky je integrace dat, znalostí a nástrojů nezbytných k aplikaci těchto dat a znalostí v rozhodovacím procesu spojeném s péčí o pacienta.

### 1.1.1 Aplikace lékařské informatiky

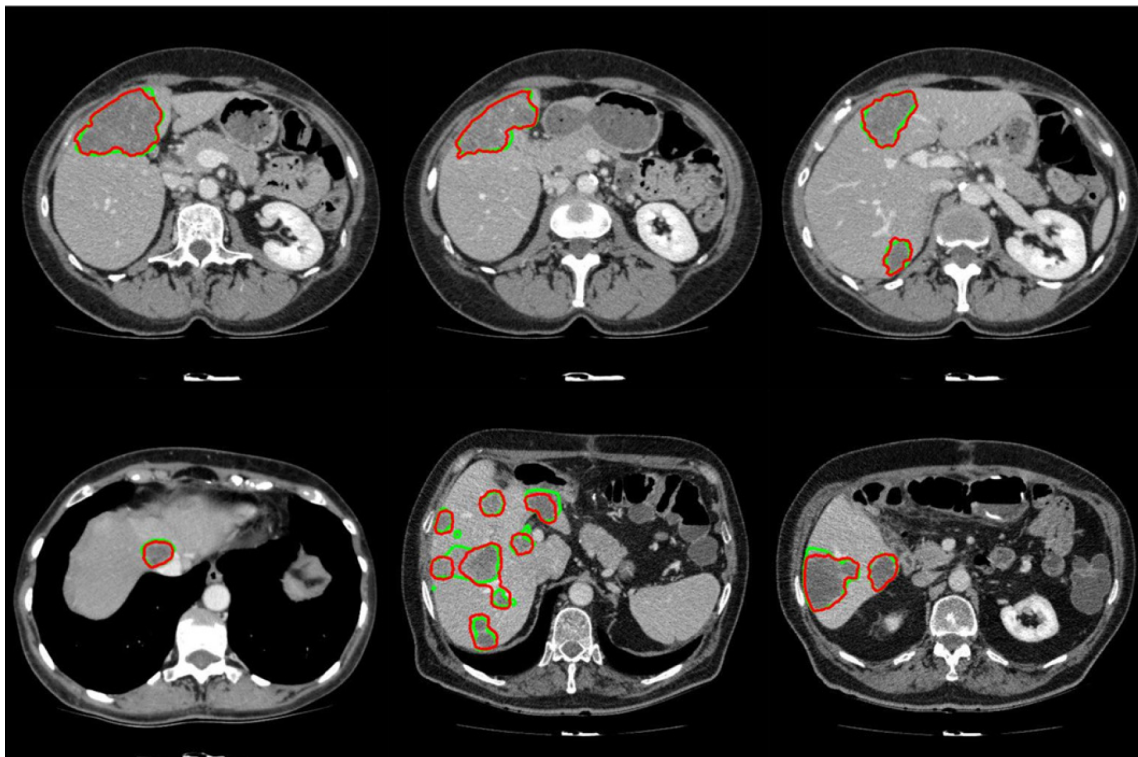
Aplikace počítačů ve zdravotnictví jsou velmi rozsáhlé, ale oblast lékařské informatiky lze dle [2] strukturovat nebo rozdělit do následujících oblastí:

- Zpracování signálů - elektrokardiografie (EKG), elektroencefalografie (EEG), elektromyografie (EMG)
- Zpracování obrazu - Radiografie, CT skenování, MRI/magnetická rezonanční angiografie (MRA)
- Počítačové záznamy pacientů
- Systémy pro podporu rozhodování
- Telemedicína
- Internetová a webová lékařská komunikace

Při práci s těmito systémy je potřeba počítat s tím, že mnoho z těchto systémů má vysokou míru falešně pozitivních výsledků, které nemusí být relevantní pro klinickou situaci. Stejně jako u každého automatizovaného systému, je pro rutinní klinickou aplikaci nutné po počítačovém vyhodnocení provést analýzu zkušeným lékařem [3].

Počítače jsou užitečná zařízení pro zpracování elektrických signálů z různých zdrojů, jako je EKG pro detekci srdečních arytmí a EEG pro analýzu a detekci

špiček a ostrých vln, které může lékař někdy přehlédnout, jak bylo zjištěno v [4]. Zpracování signálu může pomocí počítačů pomoci při analýze tvaru vlny, měření složených svalových akčních potenciálů (CMAP) a sensorických nervových akčních potenciálů (SNAP) a kvantitativních měření amplitudy a plochy. V kombinaci se srovnávacím systémem pro hodnoty referenčního rozsahu lze vytvořit zprávu na míru pro snadné rozpoznání abnormálních hodnot a konečnou diagnózu, jak je rozebráno v [5].



Obrázek 1: Snímky z CT skenu, na který byly použity metody počítačového vidění pro nalezení nádorů jater, převzato z [6]

Biomedicínské zobrazování se soustředí na zachycení obrazů pro diagnostické i terapeutické účely. Snímky fyziologie a fyziologických procesů lze získat pomocí pokročilých senzorů a počítačové technologie. Biomedicínské zobrazovací technologie

využívají buď rentgenové záření (CT skeny) viz obrázek 1, zvuk (ultrazvuk), magnetismus (MRI), radioaktivní léčiva (nukleární medicína: SPECT, PET) nebo světlo (endoskopie, OCT) k posouzení aktuálního stavu orgánu nebo tkáně. a může monitorovat pacienta v průběhu času pro diagnostiku a hodnocení léčby, jak je uvedeno v [7].

Systemy pro podporu rozhodování popsané v [8], jsou počítačové algoritmy, které pomáhají lékařům v jejich klinické praxi. Když například lékaři provádějí pomocí systému pro podporu rozhodování, jsou upozorněni, pokud se úkol jeví jako nevhodný na základě údajů o pacientovi. Systém zobrazuje toto varování automaticky pomocí pravidel pro podporu klinického rozhodování založených na konsenzu, která jsou odvozena z lékařských znalostí a informací specifických pro pacienta.

Telemedicína je distanční konzultace mezi zdravotníky nebo mezi zdravotníky a pacienty pomocí telekomunikačních technologií, jako jsou audio nebo vizuální systémy v reálném čase, zejména videokonference, jak je rozebráno v [9]. Využití je široké a rozmanité a zahrnuje přímý rozhovor a odběr anamnézy, pozorování fyzických příznaků a hlášení zobrazovacích postupů na dálku. Místo konzultace se liší od nemocničního lůžkového a ambulantního prostředí až po širší rezidenční a domácí prostředí a dokonce i ve vesmíru. Další pohodlné a nákladově efektivní využití telemedicíny bude ve výukových a poradenských situacích, zvláště když velké vzdálenosti oddělují pedagoga a studenta.

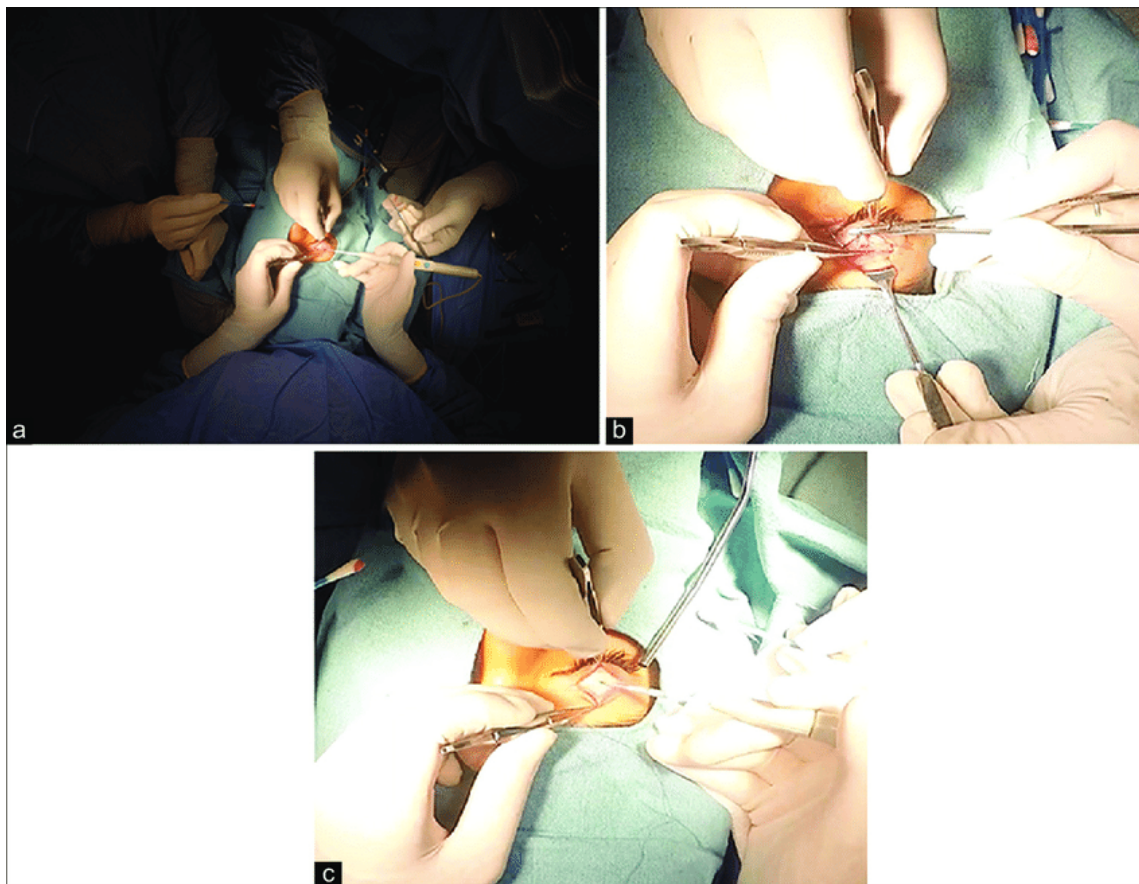
Dle [10] nebude telemedicína lékaři snadno přijata jako rutinní náhrada přímé péče o pacienty, protože postrádá intimitu rozhovorů tváří v tvář, ale je přínosným doplňkem do světa zdravotní péče.

## 1.2 Použití kamery v chirurgii

### 1.2.1 Tradiční chirurgie

Tradiční chirurgie vyžaduje řez, který umožňuje chirurgovi vidět do těla pacienta, které pak operuje rukama. Videozáznam se získává různými způsoby, např. zobrazovacím systémem mikroskopu, k záznamu externích postupů je často vyžadována externě umístěná videokamera a další kameraman, často jsou kamery také upevněny na stropní externí světelný zdroj tak, že má kamera pohled na operační výkon shora. I když tato technika může poskytovat dobré výsledky, chirurg musí často vstát a změnit polohu nebo se dokonce sklonit a ohnout, aby měl lepší přístup nebo manévrovatelnost, což brání ve výhledu kamery. V současné době probíhají i experimenty s umístěním kamery na hlavový držák, který má nasazený chirurg na hlavě. V [11] bylo zjištěno, že zachycený záznam má vysokou kvalitu a nabízí jedinečnou perspektivu. Chirurg se nemusí obávat, zda byl jakýkoliv nález nebo chirurgický krok

kompletně zaznamenán, protože zorné pole chirurga je stejné jako to, které je zachyceno kamerou. Zachycení snímků z pohledu chirurga je další výhodou jako nástroj výuky, protože studenti mohou vidět přesně to, co vidí chirurg, jak je popsáno v [12]. Příklad pohledu kamery na hlavě chirurga je na obrázku 2.



Obrázek 2: Záběry při operaci klasickým způsobem za použití kamery na hlavovém nástavci, převzato z [11]

### 1.2.2 Robotické a laparoskopické operace

Robotické a laparoskopické operace jsou tzv. minimálně invazivní operace využívající malé řezy k zavedení kamery a operačního zařízení nebo robota. Kamera tak nahrazuje zrak lékaře. Při laparoskopické operaci lékař používá nástroje ovládané rukama a řízené kamerou. V robotické chirurgii lékař používá počítačový ovladač k ovládnutí

robota během operace. Při laparoskopii se umísťují malé řezy do břicha, břišní stěna se prodlužuje oxidem uhličitým a zavádí se kamera na ohebné trubici pro zobrazení oblasti. Laparoskopie se často používá pro odběr vzorků biopsie a drobné chirurgické korekce. Laparoskopická chirurgie používá nástroje uvnitř břicha, někdy s pomocí rukou lékaře, k provádění chirurgických zákroků. Typickými laparoskopickými výkony jsou reparační kýly, bypass žaludku a resekce střeva, jak je popsáno v [13]. Pro chirurgy je nyní k dispozici celá řada robotických systémů, z nichž některé byly vyvinuty s ohledem na konkrétní typy operací. Výhodou robotické chirurgie je, že jakmile je lékař proškolen, je provádění těchto operací snazší než klasické nebo laparoskopické operace. Kamery používané robotickými systémy jsou navíc lepší a umožňují chirurgovi jasnější pohled na operaci.

Minimálně invazivní operace prováděné pomocí kamerových systémů mohou pacientům zkrátit pobyt v nemocnici a dobu zotavení. Menší řezy znamenají méně infekcí. V robotických ordinacích vykonávají obtížnou fyzickou práci chirurgie částečně roboti, což lékařům usnadňuje a zpřijemňuje práci. Počítač je navíc schopen korigovat jakékoli zakolísání rukou lékaře v důsledku únavy, což může zpřesnit operace. V [14] bylo zjištěno, že operace s kamerou sice mohou mít výrazně vyšší náklady na vybavení a školení lékařů, ale tyto nevýhody mohou být vyváženy kratší dobou hospitalizace a lepšími výsledky pacientů.

### 1.3 Digitální obraz

Digitální obraz je v zásadě dvojrozměrné pole (rastr), jehož každá položka obsahuje informaci o intenzitě (jasu) v jednom bodě rastru. Body obrazu se nazývají pixely (z anglického picture element), které je možné jednoznačně identifikovat pomocí souřadnic. Procesu získání číselných hodnot pro všechny pixely se říká vzorkování. Dalším důležitým pojmem je rozlišení. Rozlišení (anglicky resolution) monitoru nebo displeje je počet pixelů, které může být zobrazeno na obrazovce. Udává se jako počet bodů na šířku (souřadnice X) a počet bodů na výšku (souřadnice Y).

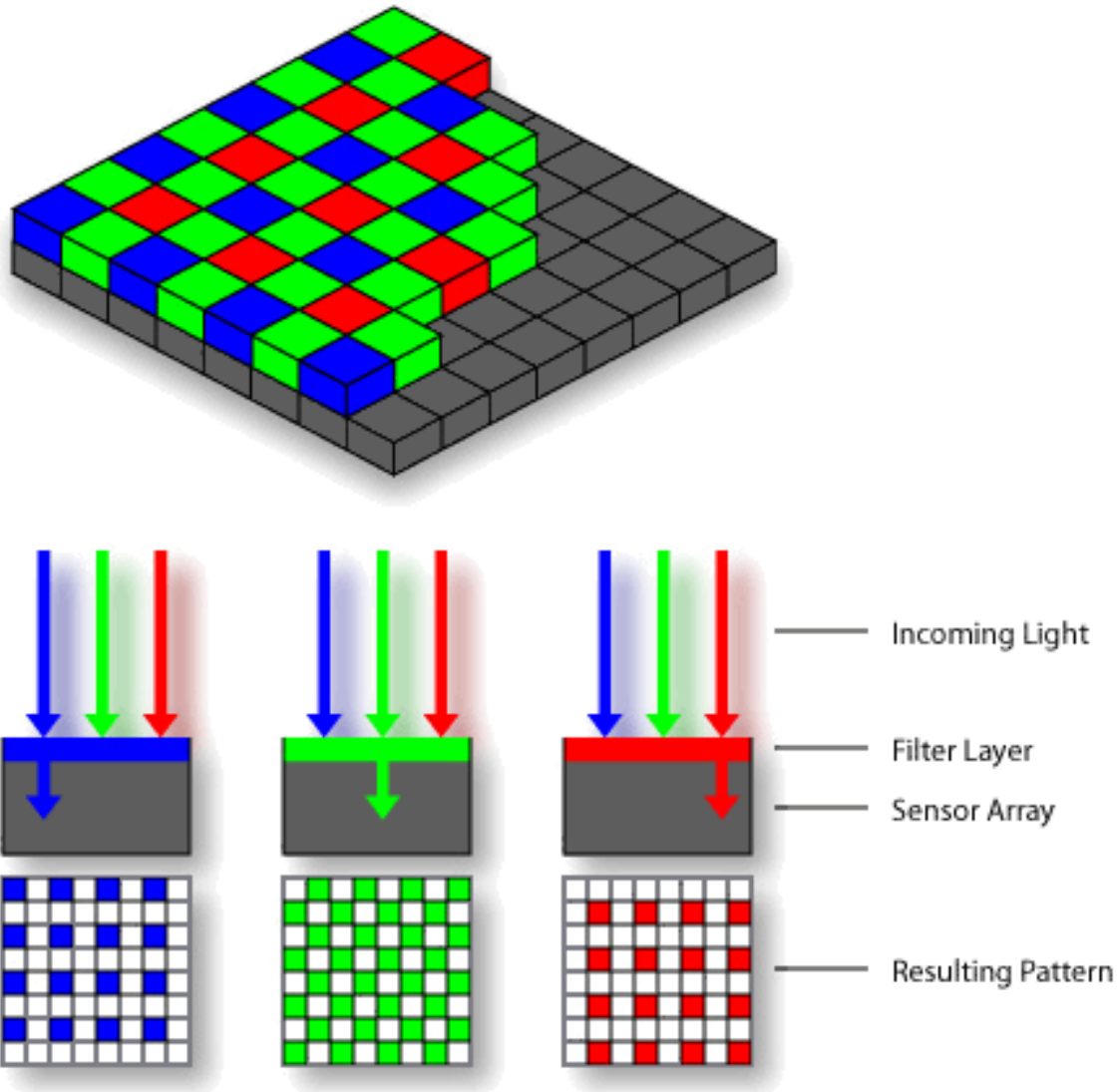
Pro černobílý obraz se zaznamenává intenzitu každého pixelu, hodnoty se pohybují od 0 do 255, kde je 0 černá a 255 bílá. Čísla od 1 do 254 jsou odstíny šedé. V případě zaznamenání barev pouze jedno číslo nestačí, je potřeba použít čísla tři v případě že se pro popis barev používají tři barvy které se spolu kombinují. Pokud jsou tyto tři barvy červená, zelená a modrá, nazýváme tento model RGB, podle anglických ekvivalentů red, green a blue, jak je popsáno v [15]. Lidské oko je citlivé na tři vlnové délky světla, odpovídající červené, modré a zelené barvě. Všechny barvy, které oko vnímá, vznikají skládáním těchto tří primárních barev. RGB systém je výhodný pro editování obrázků a jejich zobrazování na monitoru počítače a reprezentuje barvy

pomocí jejich složek v aditivním systému skládání primárních barev.

### 1.3.1 Záznam digitálního obrazu

K zachycení obrazu se dnes ve většině případů používají digitální záznamová zařízení. Může jít o digitální fotoaparáty, kamery a nebo také mobilní telefony které jsou osazeny kamerou. Tato zařízení přenášejí trojrozměrný prostor do dvojrozměrného prostoru ve formě fotografií a videí.

Princip snímání obrazu, který staví na základě fotoelektrického jevu nebo též fotoefektu, funguje na základě světla, které se odrazí od snímaného objektu, prochází objektivem a přes soustavu zrcadel dopadá na světlocitlivý snímací čip, na němž se světlo mění na elektrický náboj. Velikost náboje odpovídá intenzitě dopadajícího světelného záření. Podle [16] se aktuálně nejvíce používají snímače typu CMOS (Complementary Metal–Oxide–Semiconductor). Obrazové snímače s vestavěnými jednotkami pro strojové vidění jsou známé jako chytré obrazové snímače. U digitálních fotoaparátů a kamer jsou světlocitlivé senzory tvořeny zpravidla maticí čtvercových obrazových bodů, nad kterou jsou umístěny barevné filtry, které se nazývají Bayer filtry (příklad na obrázku 3). Jeho konkrétní uspořádání barevných filtrů se používá ve většině jednočipových digitálních obrazových snímačů používaných v digitálních fotoaparátech, videokamerách a skenerech k vytvoření barevného obrazu. Vzor filtru je z poloviny zelený, ze čtvrtiny červený a ze čtvrtiny modrý, proto se také nazývá BGGR, RGBG, GRBG, nebo RGGB. Toto uspořádání bylo zvoleno s ohledem na spektrální charakteristiky lidského oka, které má dominantní citlivost v oblasti vlnových délek 500 nm, jak je rozebráno v [15].



Obrázek 3: Bayerův filtr, převzato z [17]

### 1.3.2 Digitální zpracování obrazu

V digitálním zpracování obrazu a počítačovém vidění je segmentace obrazu proces rozdělování digitálního obrazu na více segmentů obrazu, také známé jako oblasti obrazu nebo obrazové objekty (sady pixelů). Cílem segmentace je zjednodušit nebo



změnit reprezentaci obrazu na něco, co je smysluplnější a snáze analyzovatelné.

Segmentace obrazu se obvykle používá k lokalizaci objektů a hranic (čáry, křivky atd.) v obrazech, jak je popsáno v [18]. Přesněji řečeno, segmentace obrazu je proces přiřazení třídy každému pixelu v obrázku, takže pixely se stejnou třídou sdílejí určité vlastnosti. Výsledkem segmentace obrazu je sada segmentů, které společně pokrývají celý obraz, nebo sada kontur extrahovaných z obrazu. Každý z pixelů v oblasti je podobný s ohledem na nějakou charakteristickou nebo vypočítanou vlastnost, jako je barva, intenzita nebo textura. Sousední oblasti mají výrazně odlišnou barvu s ohledem na stejnou charakteristiku, což je rozebráno v [19]. Na obrázku 4 je příklad segmentace břišní dutiny na CT skenu.

Prahování, které je popsáno v [19], je nejjednodušší metodou segmentace obrazů. Nejjednodušší metody prahování nahrazují každý pixel v obrázku černým pixelem, pokud je intenzita obrázku menší než nějaká pevná konstanta, nebo bílým pixelem, pokud je intenzita obrazu větší než tato konstanta.

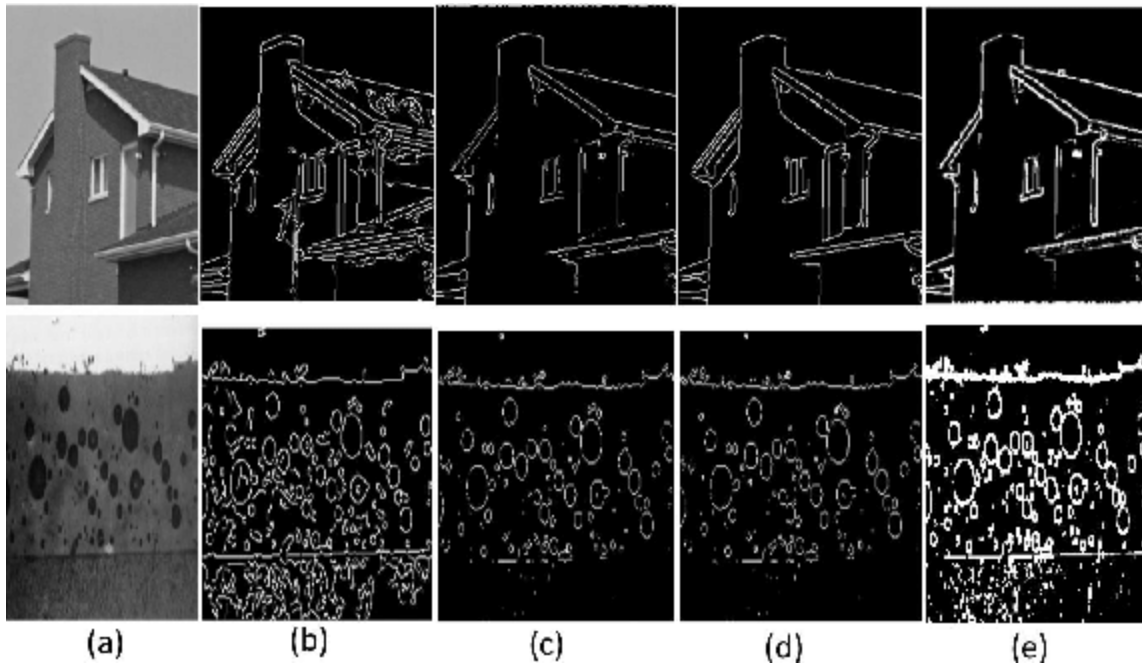


Obrázek 4: Segmentace ve snímku z CT břišní dutiny, červeně játra, zeleně žaludek, žlutě slezina, převzato z [6]

Detekce hran zahrnuje řadu matematických metod, které se zaměřují na identifikaci hran, křivek v digitálním obrazu, u kterých se jas obrazu ostře mění nebo, formálněji, má nespojitosti. Detekce hran je základním nástrojem ve zpracování obrazu a počítačovém vidění. Jeho ukázka je na obrázku 5.

Existuje mnoho metod pro detekci hran, ale většinu z nich lze seskupit do dvou kategorií, založené na vyhledávání a založené na zero-crossing(inflexní bod derivace),

jak je rozebráno v [20]. Metody založené na vyhledávání detekují hrany tak, že nejprve vypočítají míru síly hrany, obvykle derivační výraz prvního řádu, jako je velikost gradientu, a poté hledají místní směrová maxima velikosti gradientu pomocí vypočítaného odhadu místní orientace hrany, obvykle směr gradientu. Metody založené na zero-crossing hledají inflexní bod v derivačním výrazu druhého řádu vypočítaném z obrázku, aby našly hrany. Jako krok předzpracování k detekci hran se téměř vždy používá vyhlazovací fáze, typicky Gaussovské vyhlazování (redukce šumu).



Obrázek 5: Detekce hran, převzato z [21]

## 1.4 Počítačové vidění

Počítačové vidění je obor umělé inteligence, který umožňuje počítačům a systémům odvozovat smysluplné informace z digitálních obrázků, videí a dalších vizuálních vstupů – a na základě těchto informací podnikat akce nebo vydávat doporučení. Počítačové vidění funguje podobně jako lidské, lidský zrak má ale výhodu celoživotního tréninku toho, jak odlišit různé předměty od sebe. V počítačovém vidění je cílem pomocí dat, kamer a algoritmů natrénování stroje k provádění funkcí

lidského zraku. Specializované systémy nahrazující lidského pracovníka mohou analyzovat objekty rychleji a často postřehnou i běžnému oku skryté vlastnosti objektu, jak je rozebráno v [22].

Obrazová data mohou mít mnoho podob, jako jsou video sekvence, pohledy z více kamer, vícerozměrná data z 3D skeneru nebo lékařského skenovacího zařízení (například CT).

#### 1.4.1 Metody počítačového vidění k analýze obrazu

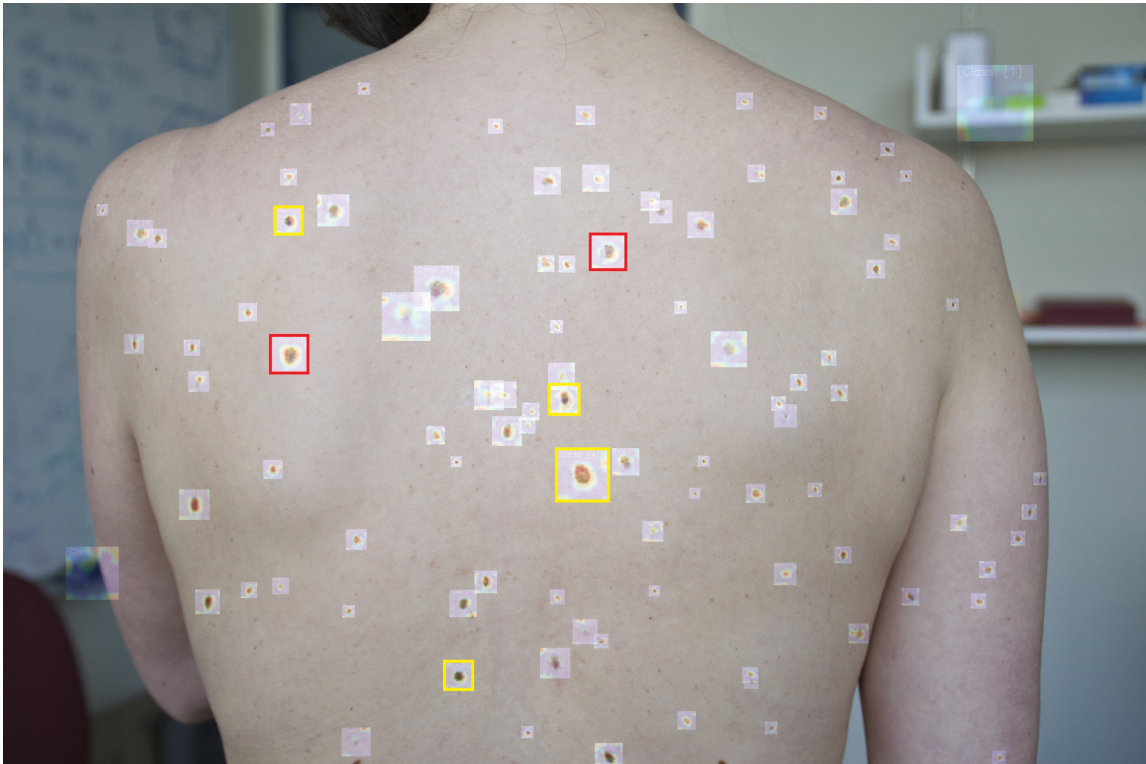
Podle [22] můžeme rozdělit základní úkoly počítačového vidění při zpracování obrazu na několik kategorií. Zde jsou uvedeny příklady:

- Klasifikace je úloha strojového učení pro určení, které objekty jsou na obrázku nebo videu.
- Lokalizace určuje polohu klasifikovaných objektů na snímku nebo videu.
- Detekce kombinuje klasifikaci a lokalizaci, poskytuje informace o poloze objektů.
- Segmentace obrazu rozděluje obraz do oblastí, z nichž každá má svůj konkrétní tvar a ohraničení.

#### 1.4.2 Aplikace počítačového vidění

Aplikace počítačového vidění sahají od průmyslových úloh, jako je třeba kontrola kvality výrobků na páse výrobní linky, až po vývoj umělé inteligence, počítačů a robotů, kteří dokážou porozumět světu kolem sebe. Základem ale je to, že k těmto činnostem by neměl být stroj explicitně naprogramován, ale měl by se učit z dat, která jsou mu dodávána.

Jednou z aplikačních oblastí počítačového vidění je zpracování obrazového materiálu pro diagnostiku pacienta. Používá se pro detekci nádorů, arteriosklerózy (která se projevuje ztrátou elasticity krevních tepen), měření rozměrů orgánů a nebo průtoku krve. Aplikace počítačového vidění v lékařské oblasti také zahrnují vylepšení obrazů interpretovaných lidmi – například ultrazvukové snímky nebo rentgenové snímky ke snížení vlivu šumu a lidské chyby, jak je popsáno v [23]. Jednou z dalších možných aplikací je detektor kožních nádorů, jeho ukázka je na obrázku 6.



Obrázek 6: Aplikace počítačového vidění pro detekci rakoviny kůže, převzato z [24]

### 1.4.3 Kalibrace kamery v počítačovém vidění

Kalibrace obrazu se v počítačovém vidění používá velmi často, pokud je zapotřebí geometrických charakteristik scény. Je nezbytná u úloh měření, pokud chceme aby aplikace dosahovala co nejpřesnějších výsledků. Po kalibraci je často jejím výstupem poměr pixel/mm. Tento parametr je kritický pro správný odhad reálné velikosti zkoumaného objektu, jak je popsáno v [25].

Kameru je možné popsat vnitřními a vnějšími parametry. Vnitřními parametry mohou být například ohnisková vzdálenost nebo skosení osy, vnější parametry jsou poloha a rotace kamery.

Kalibraci kamery lze definovat jako techniku odhadu parametrů kamery, aby bylo možné co nejpřesněji určit vztah mezi 3D objekty scény a 2D obrazem. Existuje matice kamery, která představuje specifikace fotoaparátu a mapuje 3-D scénu. Pomocí vnějších a vnitřních parametrů se vypočítají matice kamery, což je rozebráno v [26].

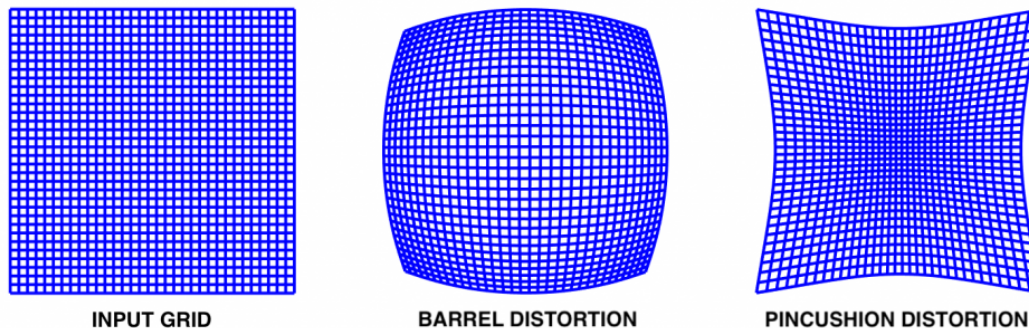
Jedním z modelů fotoaparátu je základní model fotoaparátu bez objektivu.

Světelné paprsky procházejí otvorem a promítají převrácený obraz na opačnou stranu fotoaparátu. V reálných aplikacích se ale objektiv používá a může způsobit různé druhy zkreslení. K radiálnímu zkreslení (na obrázku 7) dochází, když se světelné paprsky ohýbají více v blízkosti okrajů čočky než v jejím optickém středu.

Matice vnitřních parametrů kamery:

$$K = \begin{bmatrix} \alpha_x & \gamma & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Vnitřní matice  $K$  obsahuje 5 vnitřních parametrů konkrétního modelu fotoaparátu. Tyto parametry zahrnují ohniskovou vzdálenost, formát obrazového snímače a hlavní bod. Parametry  $\alpha_x$  a  $\alpha_y$  představují ohniskovou vzdálenost v pixelech.  $\gamma$  představuje koeficient šikmosti mezi osou  $x$  a  $y$  a často je 0.  $u_0$  a  $v_0$  představují hlavní bod, který by byl ideálně ve středu obrázku. Nelineární vnitřní parametry, jako je zkreslení čočky, jsou také důležité, i když je nelze zahrnout do lineárního modelu kamery popsaného maticí vnitřních parametrů. Mnoho moderních kalibračních algoritmů kamer odhaduje tyto vnitřní parametry také ve formě nelineárních optimalizačních technik, které jsou popsány v [27].



Obrázek 7: Radiální zkreslení - bez zkreslení, pozitivní radiální zkreslení, negativní radiální zkreslení, převzato z [28]

## 1.5 Umělé neuronové sítě

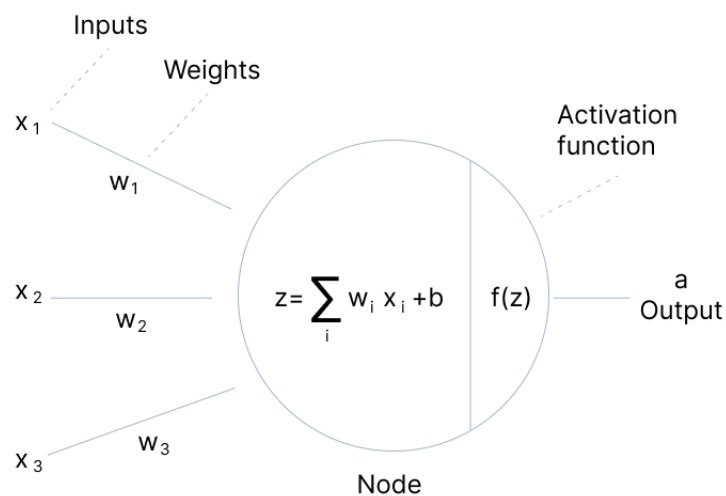
Umělá neuronová síť (v angličtině Artificial neural network) je jeden z výpočetních modelů používaný v oboru umělé inteligence. Je inspirovaný biologickými neuronovými sítěmi, které jsou jednou z hlavních součástí mozku živočichů.

Biologická neuronová síť, popsaná v [29], je tvořená z neuronů, jakožto hlavních buněk pro jejich stavbu, a spojením mezi nimi. Biologický neuron je tvořen tělem (soma), větším množstvím dendritů, které přivádí elektrické signály k tělu neuronu a jedním axonem, který vede výstupní signál z těla neuronu. Tyto neurony jsou mezi sebou spojeny synapsemi, které zajišťují předávání chemických a elektrických signálů. Pomocí excitátorů (budících látek) a inhibitorů (tlumících látek) synapse ovlivňují šanci neuronu vygenerovat impuls. Ten nastane v případě, že vybuzené a ztlumené signálu v součtu přesáhnou určitý práh, potřebný k vygenerování impulsu. Tento impuls se poté šíří axonem, které se větví k dalším synapsím. Po vygenerování impulsu nastává refrakterní doba, během níž není neuron schopen generovat další impuls, dochází k obnově mechanismů nutných pro generování impulsů.

Umělá neuronová síť, která je popsaná v [30], je složena z umělých neuronů, které se snaží matematicky modelovat biologické neurony. Tyto modely neuronů jsou spojeny do sítě a předávají si signály, které umělý neuron vyhodnocuje a případně sám nějaký signál vyšle. Signálem je reálné číslo a výstup umělého neuronu se získá po průchodu součtu vstupních signálů většinou nějakou nelineární funkcí. Vstupní signály mohou mít váhy, tyto hodnoty imitují excitátory a inhibitory biologického neuronu a svojí hodnotou zesilují či zeslabují signál. Umělé neurony, podobně jako biologické neurony, mohou mít práh, který musí všechny signály v součtu překonat, aby neuron generoval signál. Důležitou vlastností je to že umělý neuron může mít několik vstupů, ale vždy pouze jeden výstup.

### 1.5.1 Perceptron

Perceptron je diskrétní model umělého neuronu uvedený v roce 1958 F.Rosenblattem [31]. Jeho vstupem jsou výstupy předchozí vrstvy, které jsou přenásobené hodnotami vah. Také se do něj přivádí hodnota prahu, vyskytující se jako hodnota 1 přenásobená právě prahem. Všechny tyto hodnoty se nakonec sečtou a jsou argumentem vnitřní funkce neuronu, které se říká aktivační funkce. Výsledek aktivační funkce pro zadaný argument se stává výstupem perceptronu. Model perceptronu je na obrázku 8.



V7 Labs

Obrázek 8: Model perceptronu, převzatý z [32]

Model perceptronu se dá popsat vzorcem:

$$z = \sum_{i=1}^N \omega_i \cdot x_i + b$$

kde  $z$  je potenciál perceptronu,  $\omega_i$  je váha  $i$ -tého vstupu,  $x_i$  je jeden ze vstupů a  $b$  je práh. Konečným výstupem perceptronu je:

$$a = f(z)$$

kde  $a$  je výstup,  $f$  je aktivační funkce a  $z$  je potenciál neuronu.

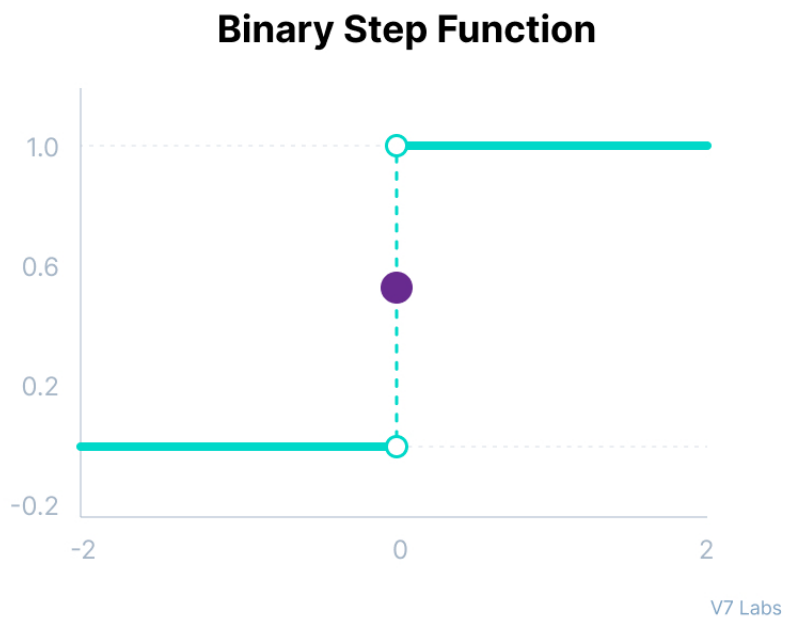
### 1.5.2 Aktivační funkce

Zde jsou uvedeny vybrané aktivační funkce a grafy jejich funkcí:

- Binární skoková funkce - závisí na prahové hodnotě, která rozhoduje, zda má být neuron aktivován nebo ne. Mezi nevýhody této funkce patří to že nemůže poskytovat výstupy s více hodnotami a má nulový gradient, což způsobuje

překážku v procesu zpětného šíření chyby (backpropagation) [32].

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



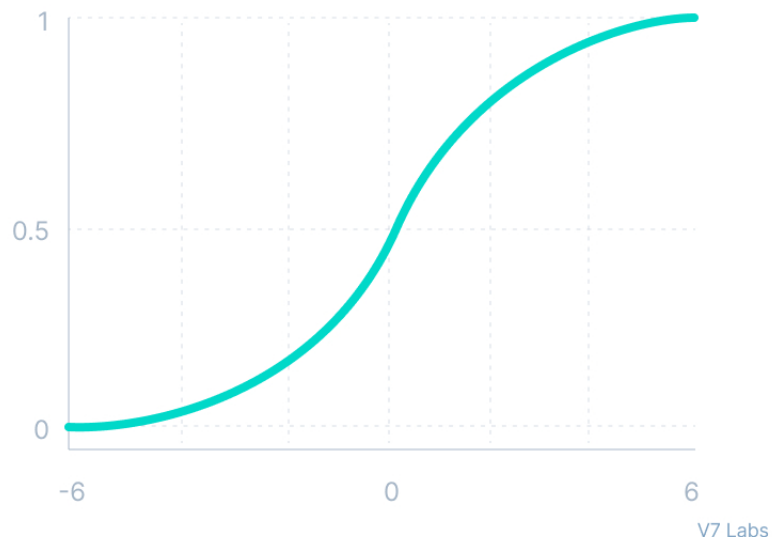
Obrázek 9: Binární skoková funkce [32]

- Sigmoidální funkce - přijímá jakoukoli reálnou hodnotu jako vstupní a výstupní hodnoty jsou v rozsahu 0 až 1. Běžně se používá pro modely, kde musíme předpovídat pravděpodobnost jako výstup. Protože pravděpodobnost čehokoli existuje pouze mezi rozsahem 0 a 1, je sigmoid správnou volbou kvůli svému rozsahu. Funkce je diferencovatelná a poskytuje hladký gradient, zabraňuje skokům ve výstupních hodnotách [32].

$$f(x) = \frac{1}{1 + e^{-x}}$$



## Sigmoid / Logistic



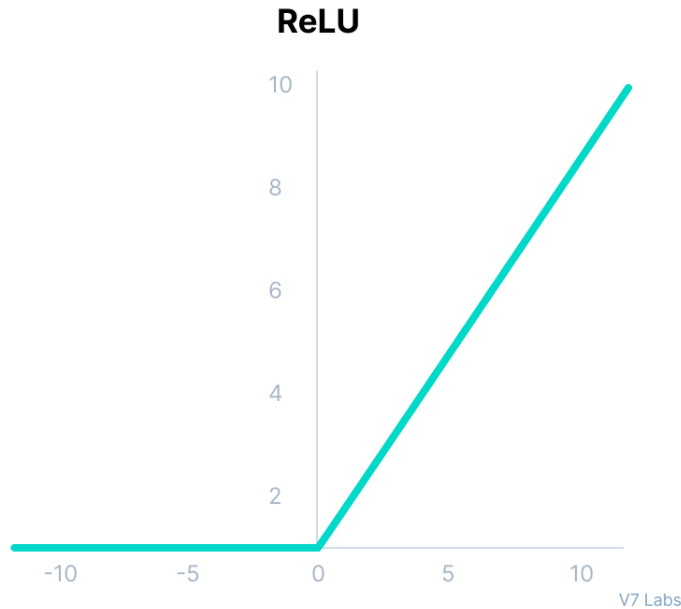
Obrázek 10: Sigmoidální funkce [32]

- Tanhova funkce - je velmi podobná funkci sigmoid a má stejný tvar s rozdílem ve výstupním rozsahu -1 až 1. Obvykle se používá ve skrytých vrstvách neuronové sítě, protože její hodnoty leží mezi -1 až; průměr pro skrytou vrstvu tedy vyjde na 0 nebo se jí velmi blíží. Pomáhá při centrování dat a usnadňuje učení pro další vrstvu [32].

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- ReLU funkce - má derivační funkci a umožňuje zpětné šíření a zároveň je výpočetně efektivní a neaktivuje všechny neurony současně. Protože je aktivován pouze určitý počet neuronů, je funkce ReLU mnohem efektivnější ve srovnání s funkcemi sigmoid a tanh. ReLU také urychluje konvergenci sestupu gradientu ke globálnímu minimu ztrátové funkce [32].

$$f(x) = \max(0, x)$$



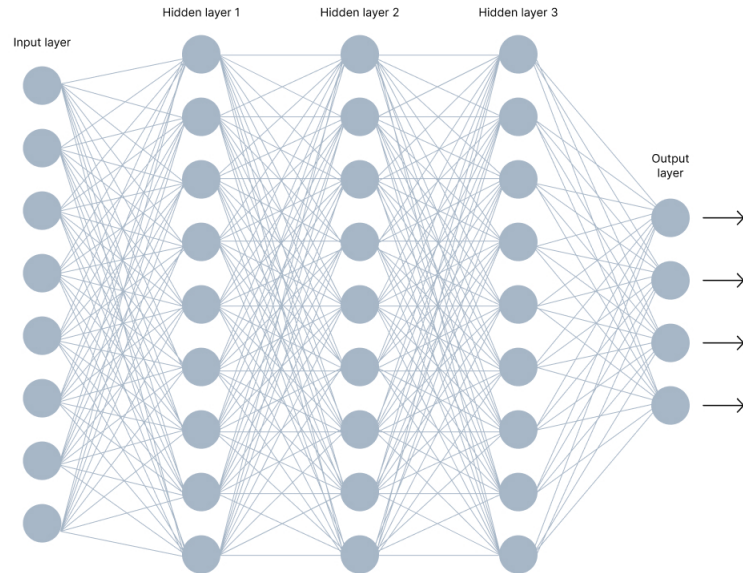
Obrázek 11: Funkce ReLU [32]

- Leaky ReLU funkce - Výhody Leaky ReLU jsou stejné jako u ReLU, kromě toho, že umožňuje zpětné šíření i pro záporné vstupní hodnoty [32].

$$f(x) = \max(0.1x, x)$$

### 1.5.3 Struktura umělých neuronových sítí

Umělá neuronová síť vznikne spojením jednotlivých neuronů. Neurony jsou obvykle složeny do několika vrstev, přičemž neurony v jedné vrstvě se mohou spojit pouze s neurony ve dvou sousedních vrstvách. Vrstva která přijímá externí data se nazývá vstupní vrstva. Vrstva která poskytuje konečný výsledek se nazývá výstupní vrstva. Mezi vstupní a výstupní vrstvou se zpravidla nachází různý počet vrstev, které se nazývají skryté vrstvy. Propojení vrstev je možné několika způsoby, pokud je výstup z každého neuronu jedné vrstvy přiveden na vstup všech neuronů z následující vrstvy jde o plně propojenou síť, v angličtině se setkáváme s označením dense nebo fully conected. Příklad dense sítě na obrázku 12.



V7 Labs

Obrázek 12: Struktura neuronové sítě [32]

Základními typy umělých neuronových sítí jsou vícevrstvé dopředné neuronové sítě a neuronové sítě s dopřednou vazbou. V dopředných sítích (feedforward network) se informace přenáší pouze od vstupu sítě na její výstup, nedochází k cyklům. U sítí se zpětnou vazbou (recurrent network) dochází na rozdíl od dopředných sítí také k šíření výstupu na vstup a dá se u ní pozorovat jisté dynamické chování. Chod sítě se ukončí poté co síť dosáhne rovnovážného stavu, kdy se již její výstup nemění.

Pro různé úlohy je potřeba navrhnou často úplně odlišné architektury sítí. Výslednou funkci určuje způsob propojení, použité aktivační funkce a nastavením prahů a vah. Rozšířením konvenčních umělých neuronových sítí jsou hluboké neuronové sítě. Ve srovnání s konvenčními neuronovými sítěmi existují rozdíly, které hluboké neuronové sítě mají. Pro konvenční neuronové sítě je běžné mít jednu nebo dvě skryté vrstvy. Na druhou stranu v hlubokých neuronových sítích je mnoho skrytých vrstev. Čím větší je síť a čím více vrstev má, tím složitější se síť stává a tím více zdrojů a více času potřebuje na trénování. Hluboké neuronové sítě fungují nejlépe s architekturami založenými na GPU, které vyžadují méně času na trénování než klasické CPU.

#### 1.5.4 Učení neuronových sítí

Jednou z hlavních vlastností neuronových sítí je jejich schopnost učit se. Při učení se síť adaptuje na řešený problém a snaží se ho lépe zvládat. Dochází k změnám vah a prahů k zlepšení přesnosti výstupních výsledků. Přesnost sítí zvyšuje minimalizováním chyby, která je typicky zjišťována jako rozdíl vypočítaného výstupu od požadovaného výstupu. Funkce která vypočítává přesnost se nazývá ztrátová nebo kritériální funkce (anglicky cost function).

Činnost neuronových sítí se dá rozdělit na dvě fáze, na fázi nastavování, při níž je cílem právě trénování, a na fázi pracovní, při které už natrénovaná síť reaguje na předložené vstupy podle svých nastavení. Při trénování je naším cílem nastavit váhy a prahy tak, aby síť dokázala odvádět svou práci co nejpřesněji. Většinou se setkáváme se dvěma typy učení: učení s učitelem (Supervised learning) a učení bez učitele (Unsupervised learning). V aplikacích neuronových sítí se ve většině případů používá učení s učitelem.

Algoritmus učení s učitelem používá dvojice dat, vstup a očekávaný výstup, přičemž cílem trénování je získat na zadaný vstup očekávaný výstup. Trénování probíhá v trénovacích cyklech, kdy jsou síti předloženy všechny dvojice právě jednou. Pořadí výběru trénovacích dvojic má vliv na průběh učení, proto se doporučuje vybírat dvojice pro trénování náhodně. Podle chyby na výstupu sítě, která se odvíjí od toho jak dobře síť rozpoznala předložený vstup se poté upravují váhy a prahy ve všech vrstvách sítě. Tento algoritmus úpravy vah a prahů se nazývá backpropagation. Toto trénování může probíhat dávkově, kdy se nejprve projde celá trénovací množina a až poté se upravují váhy a prahy, a nebo sekvenčně kdy k těmto úpravám dochází po každé trénovací dvojici. Úkoly hodící se pro učení s učitelem jsou například klasifikace nebo regrese (aproximace funkcí).

Algoritmus učení bez učitele, popsáný v [33], se snaží hledat souvislosti v nepopsaných datech. Cílem je aby si stroj vytvořil svojí vnitřní reprezentaci světa a byl schopen generovat výstupy. Ve fázi učení se síť snaží napodobovat data, která dostala jako vstup a poté si podle chyb upraví své váhy a prahy. V prostředí neuronových sítí se tento typ učení používá pouze u specifických úloh. Používá se pro shlukování, asociace závislostí mezi proměnnými a nebo také při redukci rozměrů datové sady (tato technika se často používá ve fázi předběžného zpracování dat, například když auto kodéry odstraňují šum z vizuálních dat, aby zlepšily kvalitu obrazu).

Obecně platí, že pokud síť obsahuje malý počet neuronů, její schopnost vystihnout a popsat závislosti v trénovacích datech je slabší. Pokud bude síť naopak obsahovat příliš velký počet neuronů, tato síť pravděpodobně nebude mít problém navést a reprezentovat závislosti v trénovacích datech, ale její schopnost generalizace, tedy vystihnout správný výsledek na nových datech, může být horší. Takovému jevu se

říká přeučení sítě (overfitting). K přeučení také může docházet ve chvíli, kdy model obsahuje velký počet vstupních parametrů a relativně málo pozorování. Cílem tedy není maximalizace výkonu sítě na trénovacích datech, ale rozumný kompromis mezi trénovacím výkonem a schopností zevšeobecňovat znalosti i na nových datech.

### 1.5.5 Konvoluční neuronové sítě

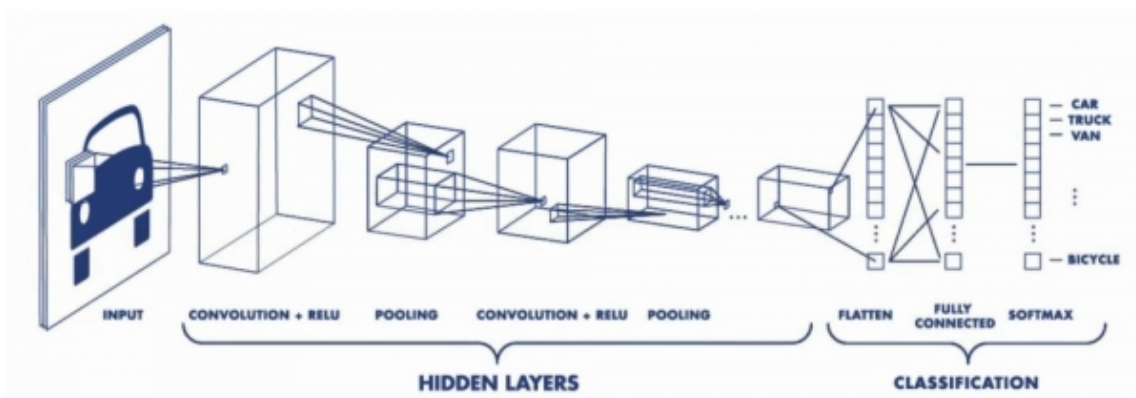
Konvoluční neuronové sítě (anglicky Convolutional neural network(CNN)) patří mezi hluboké neuronové sítě, které nejčastěji pracují s obrazem, videem a nebo rozpoznáváním řeči, jak je probráno v [34]. V oblasti obrazu konvoluční sítě nepotřebují tolik preprocessingu jako klasické metody počítačového vidění, protože mají schopnost naučení se preprocessingových filtrů.

Jednotlivé neurony reagují pouze na podněty oblasti které se říká receptivní pole. Tyto pole se překrývají aby pokryly celou vizuální oblast, dokáže tak redukovat obraz do podoby, kde se lépe zpracovává, bez ztráty informací důležitých k získání dobrých výsledků. Konvoluční sítě je schopna zachytit prostorové a časové závislosti v obraze pomocí použití příslušných filtrů. Filtr který je zapojený do konvoluce se nazývá Kernel. Filtr se pohybuje doprava s určitou hodnotou kroku, dokud nezanalyzuje celou šířku obrazu. Poté sestoupí dolů a pokračuje se stejnou hodnotou kroku, dokud tímto způsobem neprojde celý obraz. V případě barevných obrázků má kernel stejnou hloubku jako obrázek, každá barva má svůj vlastní filtr a výsledky se pak sčítají, čímž získáváme výstup. Cílem této operace je extrahovat ze vstupního souboru obrazu prvky s vysokou úrovní, jako jsou například hrany [34].

Konvoluční sítě ale nemusí být omezeny pouze na jednu konvoluční vrstvu. První konvoluční vrstva obvykle zachycuje nízkourovňové prvky, jako je barva, okraje, orientace přechodu. Pokud se připojí další vrstvy, architektura sítě se přizpůsobí na funkce na vyšší úrovni. Výstupu konvoluční vrstvy se také říká příznaková mapa.

Další vrstvou konvolučních neuronových sítí je slučovací vrstva (pooling), popsaná v [35]. Podobně jako konvoluční vrstva se snaží o zmenšení prostorové velikosti vstupu. Kromě zlepšení výkonu se také hodí k extrakci dominantních prvků. Používají se dva druhy poolingů, Max pooling, který vrací maximální hodnotu v oblasti pokryté filtrem, a Average pooling který vrací průměr hodnot z části obrazu pokryté filtrem.

Konvoluční a slučovací vrstvy tvoří pouze část celé sítě. Poté co vstup projde vrstvami konvoluce a poolingů je výstup dodáván do běžné neuronové sítě pro další zpracování, např. klasifikaci viz obrázek 13.



Obrázek 13: Model sítě s konvoluční a pooling vrstvou, převzato z [36]

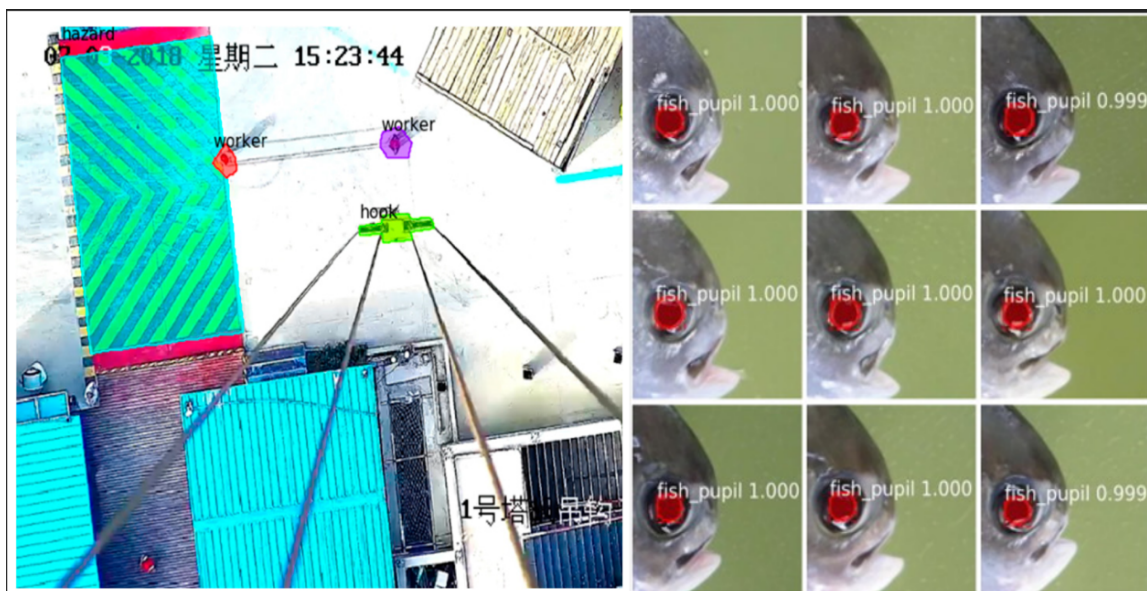
## 1.6 Podobné řešení dalších problémů v praxi

Návrhy řešení, který využívá konvolučních neuronových sítí pro měření (konkrétně sítě Mask R-CNN) již byly v minulosti použity v praxi.

Jedním z nich je aplikace navržená v [37] pro zlepšení bezpečnosti na stavbách či skladech na místech kde se využívají věžové jeřáby. Protože pracovníci nemohou vždy věnovat pozornost prostředí v horní části hlavy, je často obtížné vyhnout se nehodám při pádu těžkých předmětů. Výzkum zaznamenával videodata kamerou věžového jeřábu, označoval obrázky a provozoval rozpoznávání obrazu pomocí metody Mask R-CNN. Dále byla provedena extrakce barev RGB na identifikované vrstvě masky, aby se získaly pixelové souřadnice pracovníků a nebezpečné zóny. Nakonec byly použity metody převodu pixelů na skutečné vzdálenosti k měření bezpečné vzdálenosti. Tato aplikace pomoci zlepšit bezpečnost provozu řidičů věžových jeřábů.

Další aplikací je automatické měření ryb v rybích sádkách, popsané v [38]. Morfologické rysy ryb, jako je délka těla, šířka těla, délka ocasní stopky, šířka ocasní stopky, průměr zornice a průměr oka, jsou velmi důležité ukazatele v chytré mari-kultuře. Přesné měření morfologických znaků má proto velký význam. Stávající metody měření však spoléhají především na ruční měření, které je provozně složité, má nízkou účinnost a vysokou subjektivitu. K vyřešení těchto problémů bylo navrženo schéma pro segmentaci obrazu ryb a měření indikátorů morfologických vlastností ryb na základě Mask R-CNN. Snímky rybího těla byly získávány podomácku vyrobeným zařízením pro získávání snímků. Poté jsou obrázky ryb předzpracovány a označeny a vloženy do Mask R-CNN pro trénink. Nakonec je natrénovaný model použit k segmentaci obrazu ryb, takže lze získat indikátory jejich morfologických vlastností (na

obrázku 14).



Obrázek 14: Použití Mask R-CNN pro bezpečnost v okolí jeřábů a pro měření morfologických vlastností ryb, převzato z [37] a [38]

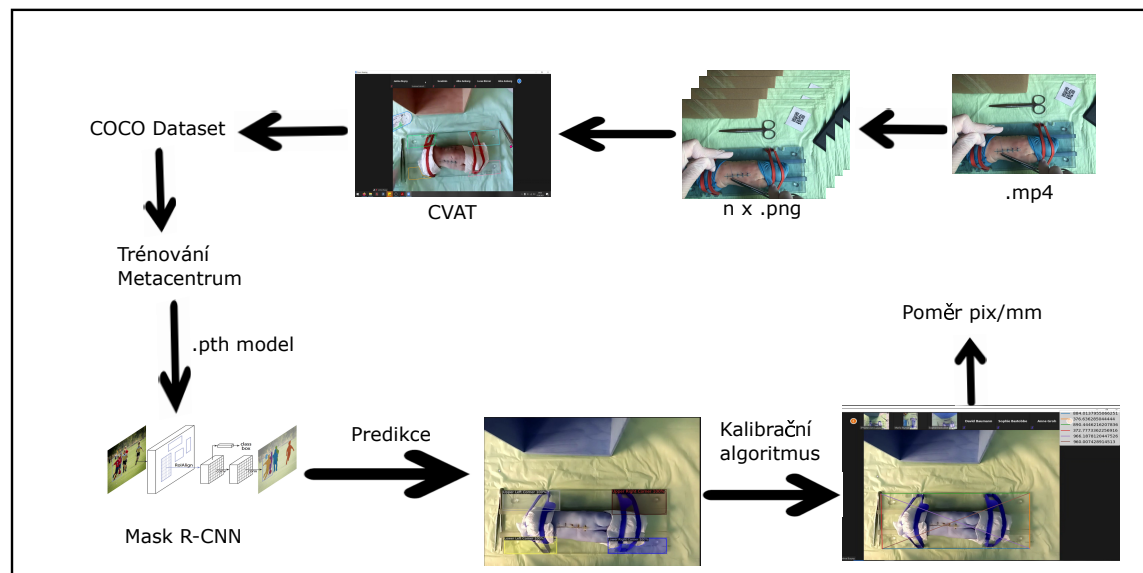
Podobné metody se také používají při automatické inventarizaci lesů, při které lze použít konvoluční neuronové sítě (CNN), popsané v [39]. Žádné předchozí studie však nevyvinuly model CNN, který by současně detekoval korunu a výšku stromu. Tato studie byla první svého druhu, která zkoumala trénování konvoluční neuronové sítě založené na oblasti masky (Mask R-CNN) pro automatickou a souběžnou detekci nespojitě koruny stromu a výšky jedle čínské.

## 2 Návrh systému pro kalibraci

### 2.1 Představení problému

Cílem této práce je navrhnout algoritmus, který zpřesní výpočet měření trajektorií při operačních výkonech. Tento algoritmus byl navržen přímo pro konkrétní aplikaci, a tou je automatizace hodnocení při praktické výuce na lékařských fakultách, konkrétně při tréninku šití ran. V době koronavirové pandemie byla prezenční výuka na většině škol nahrazena distanční výukou za pomoci informačních technologií. Posluchárny a laboratoře nahradily online videokonference. Záznamy těchto videokonferencí pořízené lektory ve Fakultní nemocnici Univerzity Friedricha Schillera v Jeně při distanční praktické výuce posloužily jako zdroj dat pro natrénování algoritmu.

Tato práce pojednává o možném přístupu ke kalibraci kamery pomocí neuronových sítí. Dosud používané řešení problému určení metrických vlastností zkoumané scény spočívalo v nalezení QR kódu, který byl vytištěn a musel být umístěn na scénu a toto řešení se ukázalo jako nespolehlivé. Přístup představený v této práci žádný přidaný QR kód nepoužívá. Metrické informace se zjišťují pomocí známé velikosti objektů a jejich relativní polohy v obraze.



Obrázek 15: Plán řešení

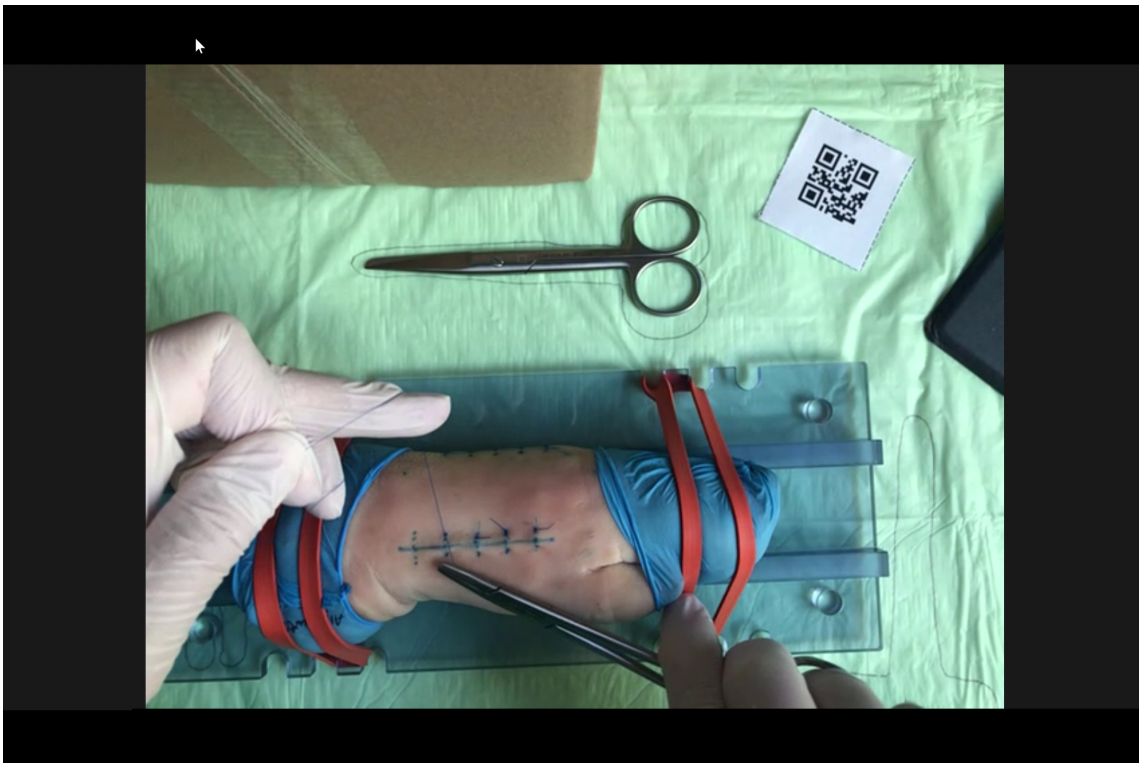


## 2.2 Příprava dat

Data, která jsou v této práci používána, pocházejí ze záznamů obrazovky instruktora, který přes aplikaci Zoom, která se používá pro online video konference, pozoruje práci studenta.

### 2.2.1 Typické video

Typické video, které se stává vstupem algoritmu, je záznamem obrazovky z počítače lektora. Běžně je na videu úplně celá obrazovka, včetně spodní lišty operačního systému a okna softwaru pro videokonference. Okno softwaru na videokonference (v tomto případě software Zoom) obsahuje přenos z kamery u studenta, na spodní straně se můžou ukázat ovládací prvky aplikace, pokud lektor přejíždí kurzorem přes aplikaci a na vrchní straně aplikace jsou malé náhledy několika dalších přenosů. Videokonference zpravidla navštěvuje více studentů, kteří zároveň provádějí stejnou činnost, v jednu chvíli ale lektor zaznamenává pouze jednoho z nich (viz obrázek 16).



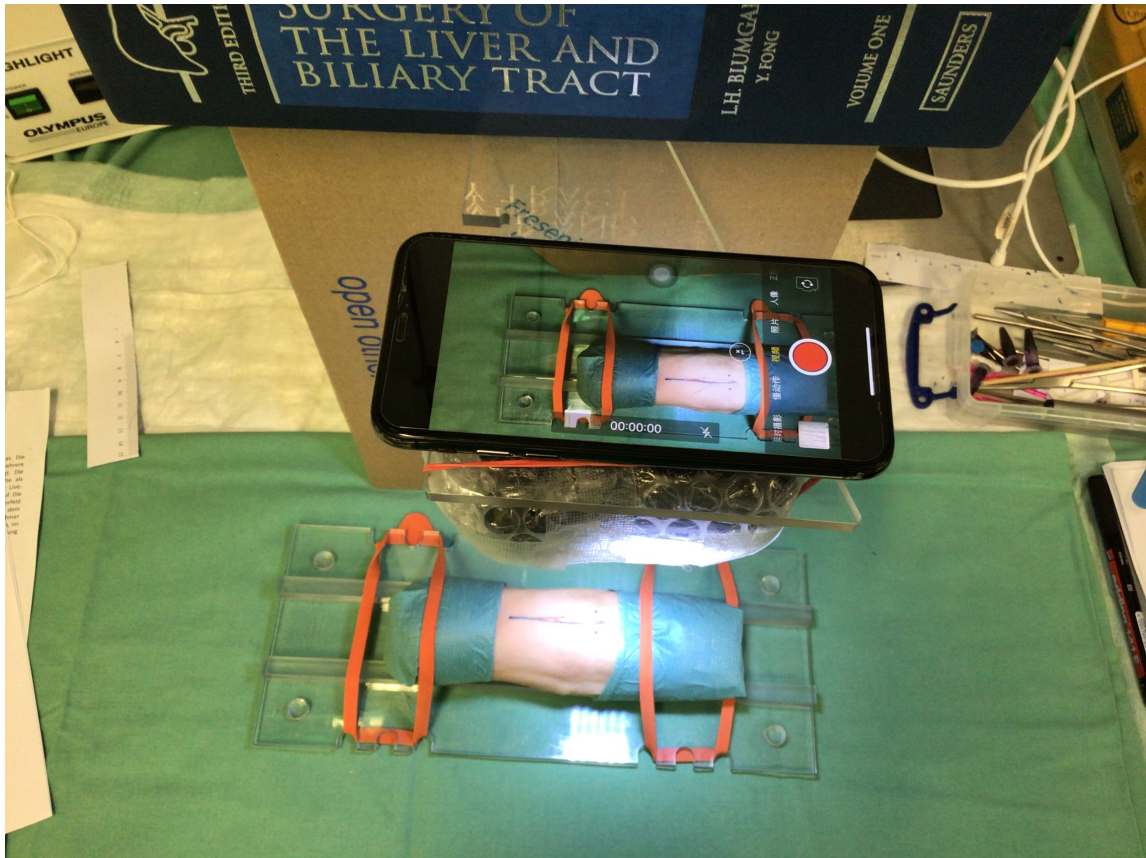
Obrázek 16: Pohled na obrazovku zaznamenávající videokonferenci

Scéna, kterou kamera studenta zaznamenává, se mezi jednotlivými studenty liší pouze v detailech. Každý z nich obdržel totožnou sadu pro praktickou výuku. Ve středu scény leží transparentní podložka, na které je pomocí gumiček připevněna prasečí noha. Ta má na sobě několik řezných ran, které má za úkol student sešít. Na šití používá několik nástrojů:

- Jעהlec - pomocný chirurgický nástroj, sloužící ke snadnějšímu šití - jehelcem se zachycuje jehla s návlekm šicího materiálu. Má nejčastěji podobu klíštěk s nůžkovitými nebo klešťovými rukojetmi a krátkými tupými čelistmi, opatřenými křížovým vroubkováním pro pevnější zachycení jehly.
- Sutura - vlákno se kterým se provádí šití tkáně - stehy.
- Chirurgická pinzeta
- Chirurgické nůžky

- Chirurgická šicí jehla

Technické parametry videí se liší, rozlišení se pohybuje od 1366x768 pixelů (HD) přes 1920x1080 pixelů (FullHD) až po rozlišení 2880x1800 a 60 snímků za vteřinu (také se můžeme setkat se zkratkou FPS - frames per second). Zde záleží na technických parametrech počítače, na kterém je spuštěný záznam obrazovky. Všichni studenti dostali stejné pokyny pro záznam videa. Na krabici, ve které jim přišlo veškeré vybavení k výuce a kterou postaví na stůl na výšku, připevní zařízení pro záznam obrazu. Kamera je položena na plexiskle, a to je zatíženo těžším předmětem, např. učebnicí chirurgie jako na obrázku 17. Kamera je tedy vždy pod pravým úhlem. Pod kamerou je připevněno světlo. Na videích použitých pro trénování neuronové sítě je také známá vzdálenost od záznamového zařízení studenta k desce stolu, je to 45 cm.



Obrázek 17: Scéna ze které jsou pořízeny snímky.

## 2.2.2 Anotace

Anotace jsou zásadní součástí učení s učitelem v prostředí počítačového vidění. Právě pomocí anotací může učitel předat znalosti a informace do algoritmů strojového učení. I tato práce obsahuje části algoritmů, ve kterých se tento přístup k učení uplatňuje.

Bylo potřeba najít ve vstupních datech objekty, nebo jejich části, u kterých jsou známy jejich rozměry a zároveň je bude snadné pro neuronovou síť vyhledat a označit. Už od začátku se jevila jako nejlepší objekt podložka, na které leží operovaný objekt. Jednou z nevýhod tohoto objektu je jeho transparentnost, která znesnadňuje detekci hran po obvodu podložky. V každém z rohů má podložka díry, které mají také jednotný průměr, proto byly uvažovány jako jeden z možných objektů pro kalibraci. V úvahu také přicházela schránka na suturu, které se na některých videích objevovala. Jde o malou oválnou plastovou schránku, jejíž rozměry byly také známy.

Nakonec byly vytvořeny třídy, do kterých se přiřadí jednotlivé části obrazu. Tyto části k třídám byly přiřazeny pomocí segmentace. Podložka nebyla segmentována celá, ale byli segmentovány pouze rohy. Vzniklo tedy 5 tříd:

1. Upper Left Corner - Levý horní roh
2. Upper Right Corner - Pravý horní roh
3. Lower Left Corner - Levý dolní roh
4. Lower Right Corner - Pravý dolní roh
5. Suture - Schránka na suturu

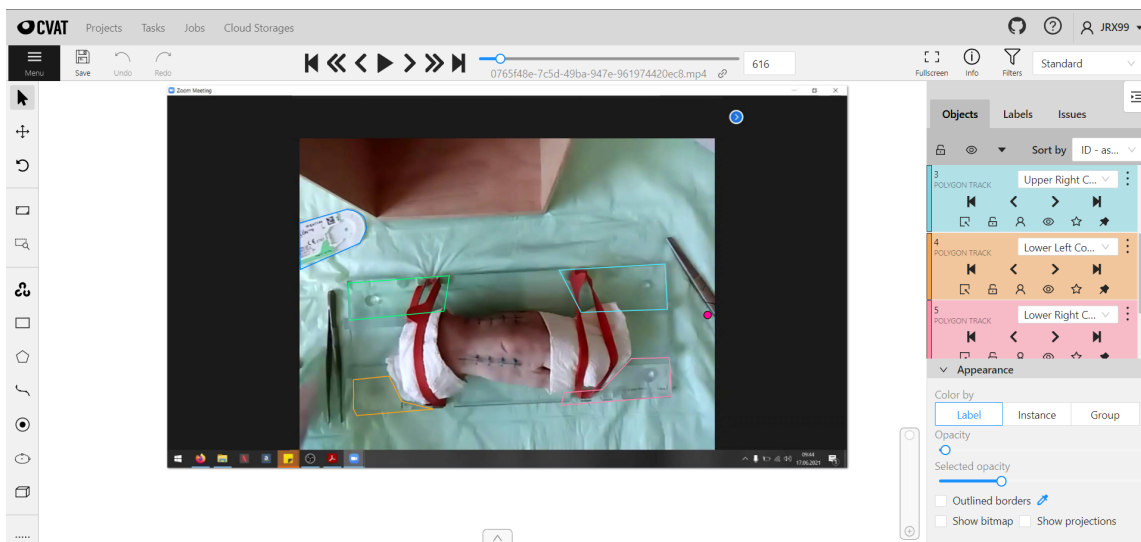
V průběhu práce se ukázalo, že pátá třída, Suture, se ve scéně vyskytuje minimálně, a pokud se ve scéně nachází, jde často jen o malou část, která by nestačila pro kalibraci. Z tohoto důvodu nebyla tato třída v dalších částech práce používána. Tuto třídu nakonec nahradila třída Hole, která označuje kruhové výřezy v rozích desky.

Jako trénovací množina bylo zvoleno 5 videí, které pak byly jednotlivě anotovány. Videá mají v závislosti na délce záznamu odpovídající počet snímků. Záznamový software na počítači lektora zaznamenává obraz s frekvencí 60 snímků za vteřinu. Například padesáti sekundové video má 3000 snímků. Videá použitá pro anotace mají 3000, 4415, 4460, 5298 a 7436 snímků, tedy dohromady 24609 snímků.

Jako limitující se ukázalo použití softwaru Zoom. Ten dokáže přenášet video maximální frekvencí 30 snímků za vteřinu. Obrazovka je ale zaznamenávána frekvencí která je dvakrát větší, proto jsou všechny snímky v zaznamenaném videu dvakrát. Rozlišení původního videa nahrávané studentem závisí na několika faktorech. Jedním

z nich je rozlišení kamery na přístroji, který student používá k záznamu, dále také záleží na nastavení přímo v aplikaci Zoom, kde se rozlišení může pohybovat od 640 x 360 pixelů do 1920 x 1080 pixelů. Při větším zatížení sítě je potřeba přizpůsobit nastavení, která ovlivňují rozlišení.

Pro anotování (proces vytváření anotací) bylo využito softwaru CVAT, což je open source webový nástroj pro anotaci obrázků a videa, který se používá pro označování dat pro algoritmy počítačového vidění. Tento software obsahuje veškeré nástroje které byly potřeba pro tuto úlohu.



Obrázek 18: Obrázek 2.3: Anotační software CVAT

Při procesu vytváření anotací bylo nejprve vybráno jedno z videí a na první snímek byly zaneseny informace pomocí anotačních nástrojů. Každý z viditelných rohů nebo schránka sutury (později kruhové výřezy) byly vyznačeny okolo obvodu (na obrázku 18). U anotací objektů, u kterých se předpokládá, že se nebudou pohybovat se může použít funkce Pin, která způsobí, že se na všechny další snímky anotace propíše. Pokud by se objekt přemístil, na posledním snímku kde je na původním místě se funkce ručně vypne.

V průběhu videa se může stát, že student svojí rukou nebo nástrojem zakryje některý z anotovaných objektů. V tomto případě je pro daný snímek a anotaci opět ručně zaznamenána vlastnost Occlusion, která může nabývat hodnot True nebo False. Pokud je hodnota True, znamená to, že je objekt překrytý a je to také dodatečná informace pro trénovací algoritmus, který by měl vzít tuto informaci v úvahu.

Aby byla trénovací data co nejlepší, bylo potřeba snímek po snímku procházet všechna videa a upravovat anotace, aby byli co nejpřesnější. Na prvních snímcích byla často zaznamenána přípravná fáze nahrávání na počítači lektora, kdy už byl spuštěný záznam, ale lektor stále vybíral, kterého ze studentů bude nahrávat. Studenti samotní také mohli v průběhu posunout záznamové zařízení, nebo podložku. Proto se scéna mohla v průběhu videa měnit a bylo potřeba důkladně procházet snímky, aby trénovací množina obsahovala co nejméně špatných dat.

### 2.2.3 Soubor COCO

Ideálním výstupem pro použití anotací v neuronových sítích je formát COCO. U každého z anotovaných videí je možno v CVATu vygenerovat soubor datasetu COCO, který je designovaný pro úlohy detekce a segmentace. Jde v podstatě o specificky strukturovaný soubor JSON. Soubor COCO má následující strukturu:

- info - informace o autorovi, datu vytvoření, verzi.
- licences - informace o licencích.
- category - informace o třídách, jejich identifikační číslo (ID), název, je také možné přiřadit nadřazenou třídu.
- images - informace o obrázcích, jejich ID, název, rozměry v pixelech
- annotations - informace o anotacích, jejich ID, obsahují souřadnice tzv. bounding boxu, což je oblast, která ohraničuje segmentovaný objekt, obsahuje také informace o překrytí objektu (atribut occluded).

Zde je ukázka zápisu jedné anotace ve formátu COCO. "id" je číslo anotace, "image\_id" je číslo snímku, "category\_id" je číslo třídy, následuje segmentace, výpočet plochy, souřadnice bounding boxu a další atributy:

---

```
{
  "id": 168,
  "image_id": 25,
  "category_id": 1,
  "segmentation": [
    [
      260.4,
      722.0,
      487.12,
```

```

        717.28,
        479.58,
        870.91,
        272.23,
        882.22
    ]
],
"area": 34099.0,
"bbox": [
    260.4,
    717.28,
    226.72,
    164.94
],
"iscrowd": 0,
"attributes": {
    "occluded": false
}
},

```

---

## 2.3 Trénování neuronové sítě

Hlavním programovacím jazykem pro tuto práci byl zvolen Python ve verzi 3.8.12. Použité balíčky v trénovacím skriptu:

- Matplotlib - knihovna pro vytváření grafů.
- Numpy - knihovna pro práci ve multidimenzionálními poli a maticemi.
- OpenCV - knihovna pro řešení problémů počítačového vidění.
- Detectron 2

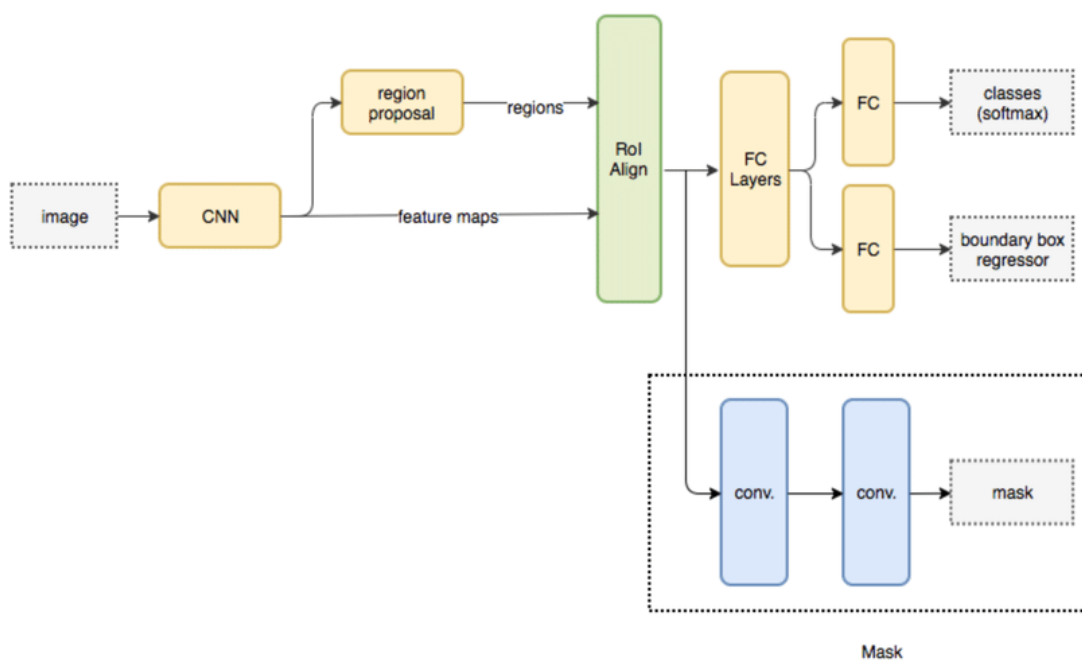
### 2.3.1 Mask R-CNN

Pro natrénování detekčního modelu byla použita platforma Detectron 2, open source projekt společnosti Facebook AI Research. Detectron 2 je implementován v PyTorch a specializuje se na detekci objektů v obraze. Detectron obsahuje předtrénované sítě v tzv. Model Zoo, takže není nutné vytvářet model úplně od začátku. Z Model Zoo byly použity dvě předtrénované sítě, Mask R-CNN (R50 FPN) a Mask R-CNN

(R101 FPN). Tyto dvě sítě se liší pouze množstvím skrytých vrstev. R50 FPN má 50 skrytých vrstev, R101 FPN má 101 skrytých vrstev. Jejich společnou vlastností je síť na které jsou obě založeny.

Mask R-CNN představená v [40], kombinuje řešení dvou problémů počítačového vidění do jednoho. Jde o segmentaci instancí (instance segmentation), která kombinuje detekci objektů, při které se snaží správně přiřadit objekt k třídě a ohraničit ho bounding boxem, a sémantickou segmentaci která má za cíl přiřadit každý pixel obrazu do některé ze zadaných tříd. Schéma sítě je na obrázku 19.

Chod sítě je rozdělen na dvě fáze, což je popsáno v [41]. První fáze se nazývá Region Proposal Network (RPN). Jejím úkolem je vytvořit bounding boxy objektů. V druhé fázi probíhají současně dvě úlohy. První z nich je ROIAlign k extrakci rysů bounding boxu a následné klasifikace bounding boxů do tříd. Druhou úlohou je vytvoření binární masky pro každý bounding box, která označuje pixely hodnotou True nebo False podle toho zda patří nebo nepatří do dané třídy.



Obrázek 19: Schéma sítě mask R-CNN převzato z [42]



Ztrátová funkce pro každou oblast zájmu je definovaná jako:

$$L = L_{cls} + L_{box} + L_{mask}$$

kde  $L_{cls}$  je chyba přiřazení do třídy,  $L_{box}$  je chyba bounding boxu a  $L_{mask}$  je chyba masky (segmentace).

$$L_{cls}(p_i, p_i^*) = -lb[p_i p_i^* + (1 - p_i)(1 - p_i^*)]$$

$L_{cls}$  je chyba přiřazení do třídy popsaná v [41], kde  $p_i$  reprezentuje predikovanou pravděpodobnost že je bod  $i$  cílem,  $p_i^*$  reprezentuje predikovanou hodnotu odpovídajícího popisku reálné plochy,  $lb$  je logaritmická ztrátová funkce .

$$L_{box}(t_i, t_i^*) = R(t_i - t_i^*)$$

$L_{box}$  je chyba bounding boxu [41], kde  $R$  je robustní ztrátová funkce,  $t_i$  představuje 4 parametrizované souřadnicové vektory predikovaného rámce a  $t_i^*$  představuje souřadnicový vektor odpovídající hranici reálné oblasti, .

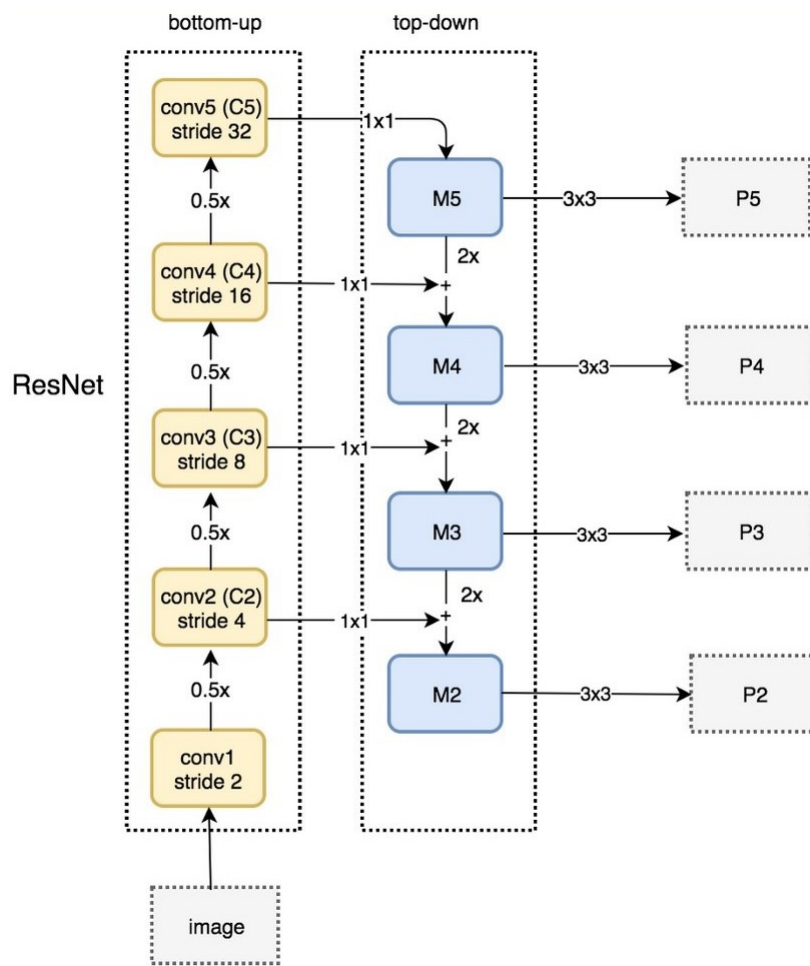
$$L_{mask} = -\frac{1}{x} \sum_i [x_i^* \log p(x_i) - (1 - x_i^*) \log(1 - p(x_i))]$$

$L_{mask}$  je chyba masky kde  $x$  představuje počet pixelů,  $x_i^*$  označuje označení kategorie, kde je pixel umístěn, a  $p(x)$  představuje pravděpodobnost  $x_i$  předpokládané kategorie.

$L_{mask}$  se vypočítá funkcí binární křížové entropie (Binary Cross Entropy), popsaná v [40], která porovnává každou z předpokládaných pravděpodobností se skutečným výstupem třídy, který může být buď 0 nebo 1. Poté vypočítá skóre, které penalizuje pravděpodobnosti na základě vzdálenosti od očekávané hodnoty. To znamená, jak blízko nebo daleko od skutečné hodnoty. Binární křížová entropie je záporný průměr logaritmu korigovaných předpokládaných pravděpodobností:

Páteřní síť Mask R-CNN je ResNet [43]. ResNet (Residual Neural Network) je architektura konvoluční neuronové sítě, která se využívá pro detekci a segmentaci v počítačovém vidění. Využívá funkce skip connection, nebo také shortcut, pro přeskočení některých vrstev. Hluboké neuronové sítě mají problém mizejícího gradientu (vanishing gradient problem) který se objevuje ve fázi zpětného šíření chyb, při kterém se chyba nepřenáší ve správné míře na první vrstvy. Díky přeskokování vrstev se tento problém podařilo síti ResNet vyřešit. Přeskakování efektivně zjednodušuje síť tím, že nejdříve přeskakuje některé vrstvy v počátečních fázích učení a při dalších

iteracích trénování se postupně obnovuje spojení zbylých vrstev, které původně síť přeskočila.

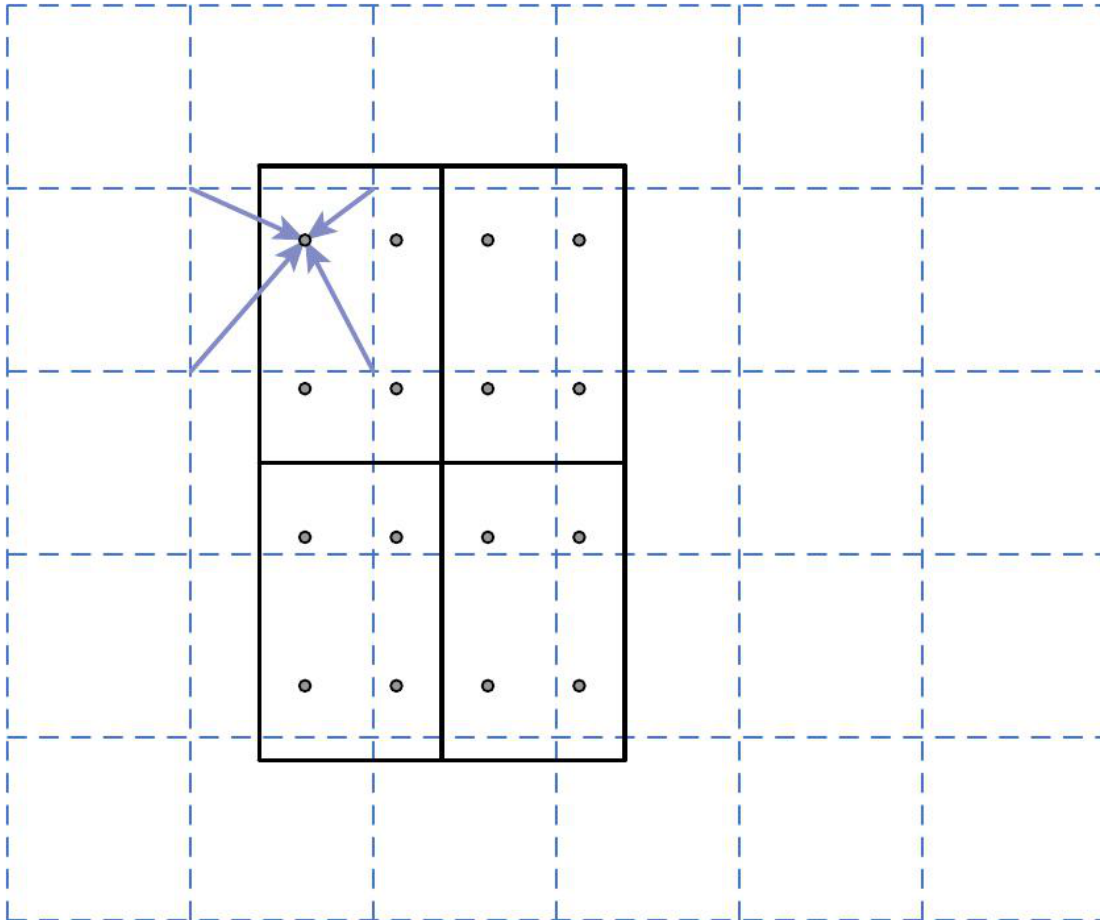


Obrázek 20: Schéma sítě ResNet s FPN převzato z [44]

V této aplikaci je využita vylepšená verze sítě ResNet, která využívá Feature Pyramid Network (FPN), popsaná v [44]. FPN se využívá pro extrakci příznaků u nižších vrstev konvoluční sítě pro zlepšení přesnosti a může sloučit rysy různých úrovní a vytvořit z ní silnou sémantickou informaci a silnou prostorovou informaci zároveň. FPN je obecná struktura, kterou lze použít v kombinaci s různými páteřními sítěmi. Na obrázku 20 je schéma sítě ResNet s FPN.

RoIAlign (na obrázku 21) je operace pro extrahování malé mapy funkcí z každé ROI (Region Of Interest) v úlohách založených na detekci a segmentaci, jako v [13].

Odstraňuje drsnou kvantizaci ROI Pool a správně zarovnává extrahované prvky se vstupem. RoIAlign používá bilineární interpolaci k výpočtu přesných hodnot vstupních funkcí na čtyřech pravidelně vzorkovaných místech v každé části ROI a výsledek je poté agregován (pomocí maxima nebo průměru).



Obrázek 21: Zarovnávání bounding boxu pomocí ROIAlign, převzato z [45]

### 2.3.2 Metacentrum

Trénování neuronových sítí s velkým objemem dat je velmi náročné na výpočetní výkon. Síť byla nejdříve trénována na lokálním laptopu, ten byl ale i přes obstojnou technickou vybavenost tímto úkolem na několik hodin naprosto zahlcen. Tento

způsob byl proto používán pouze v počátečních fázích pro vyřešení problémů, které způsobovali pád chodu programu. Po vyřešení problémů bylo využito výpočetních kapacit Metacentra, což je aktivita sdružení CESNET primárně věnovaná provozu a dalšímu rozvoji gridové infrastruktury v České republice. Zpracování objemných experimentálních výsledků, simulace či modelování složitých systémů a další typické úlohy moderního výzkumu vyžadují velký výpočetní výkon. Ten bývá dnes obvykle poskytován v podobě clusterů a gridů – systémů propojených a spolupracujících počítačů, mezi něž je úloha rozložena. Gridová infrastruktura proto tvoří jednu ze základních komponent e-infrastruktury CESNET.

Počítače Metacentra mají operační systém Linux. Pro zjednodušení práce byl vytvořen skript, který specifikuje požadované hardwarové specifikace, vytváří pracovní adresáře, aktivuje moduly Pythonu, a nakonec také spouští samotný skript pro trénování neuronové sítě. Jako minimální požadavky byly zvoleny:

- 2 procesory (CPU - Central Processing Unit)
- 2 grafické karty (GPU - Graphics Processing Unit)
- 10 GB operační paměti
- 30 GB pro data
- maximálně 6 hodin běhu skriptu

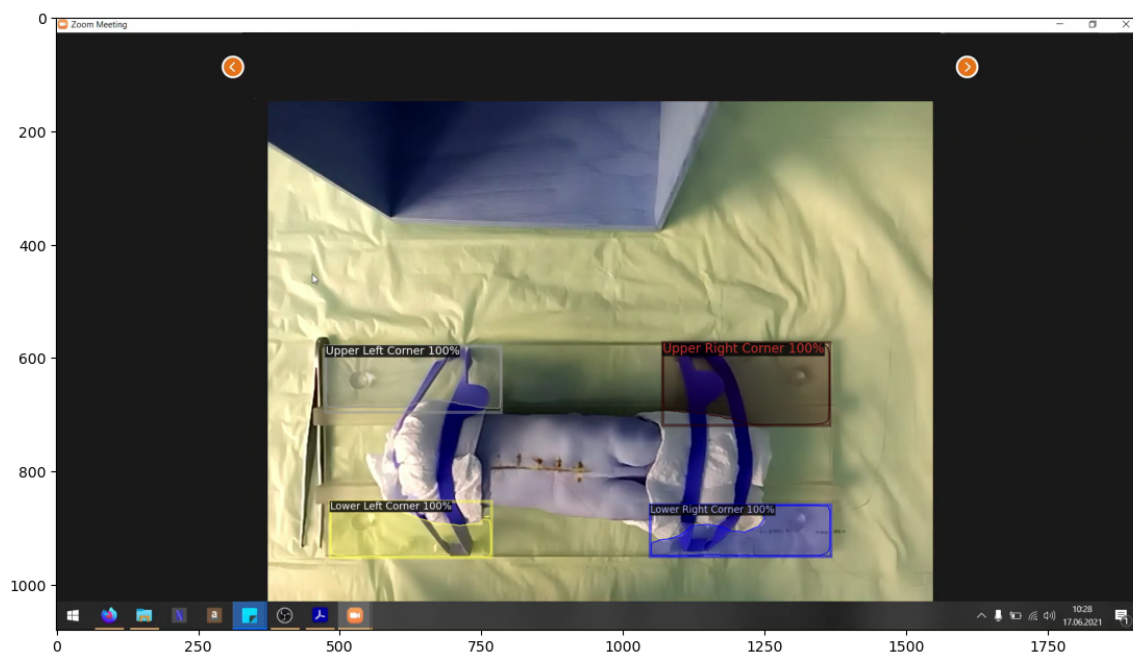
S tímto nastavením bylo možné optimální rychlostí natrénovat neuronovou síť a zároveň nevytvářet až příliš náročné úlohy, které by plánovač Metacentra zařazoval s nižší prioritou.

Pomocí výpočetních kapacit Metacentra byly natrénovány čtyři modely sítě Mask R-CNN. Pro obě architektury s různým počtem vrstev byly natrénovány dva modely, jeden s mírou učení (Learning rate) 0.0005 a druhý s mírou učení 0.001. Tabulka 1 uvádí, jak dlouho trval běh celého běhu skriptu, včetně kopírování dat na výpočetní uzel. Vzhledem k tomu, že Metacentrum přiděluje výpočetní uzly podle toho, které jsou zrovna k dispozici, trénování každého modelu probíhalo na jiném výpočetním uzlu. Doba trvání nahrání trénovacích dat na výpočetní uzel je přibližně 20 minut, zbytek času je trénování a vytváření výstupů. Tabulka také obsahuje název výpočetního uzlu, jeho konkrétní hardwarové vybavení je k dispozici na webových stránkách Metacentra.

Architektura	Learning rate	CPU čas [s]	Čas běhu [s]	V. uzel
R-CNN R50 FPN	0.0005	2604	2265	fau2
R-CNN R50 FPN	0.001	2214	3604	gita2
R-CNN R101 FPN	0.0005	5107	4358	adan31
R-CNN R101 FPN	0.001	5277	4508	adan4

Tabulka 1: Porovnání rychlosti běhu trénování

Po natrénování, které bylo omezeno na 5000 iterací, algoritmus vrací natrénovaný model ve formátu .pth. Natrénovaný model je nyní možné použít pro vytvoření prediktoru. Vstupem prediktoru může být jakýkoliv obrázek a v něm se prediktor snaží nalézt požadované objekty, segmentovat je a segmentace ohraničit obdélníkem (bounding boxem). Ke každé predikci ještě přiřadí procentuální hodnotu toho, jak si je prediktor jistý, že objekt označil správnou třídou. N obrázku 22 je příklad výstupu prediktoru.



Obrázek 22: Snímek který prošel prediktorem, který na něm provedl segmentaci instancí.

## 2.4 Navržený kalibrační algoritmus

Samotná kalibrace probíhala bez použití konvenčních kalibračních metod. Bylo využito pouze přesné detekce prediktoru, který označil body zájmu na zkoumaném objektu. Jako objekt se známými rozměry byla zvolena deska, na které byla připevněna prasečí noha.

Kalibrační algoritmus nejprve načítá informace o datasetu metodou `register_coco_instances()`, při které se ze souboru COCO s anotacemi načítají metadata o datasetu. Následuje konfigurace prediktoru, do kterého je nutné nahrát model, se který je zvolen pro úlohu, metodou `configure_network()`. Pro samotnou kalibraci se spouští metoda `calibrate()`, ta má za vstup obrázek, který je určen ke kalibraci ve formátu vícerozměrného pole.

Metoda `calibrate()` jako první spouští metodu `find_corners()`, která slouží pro nalezení požadovaných bodů zájmu. Metoda `find_corners()` má za vstup pole, ve kterém je uložený obrázek, a spouští prediktor, který použije natrénovaný model konvoluční neuronové sítě a pokusí se jím nalézt v ideálním případě 4 rohy desky a kruhové výřezy. V případech, kdy je na obrázku ruka studenta nebo nástroj který drží a který překrývá jeden z rohů, prediktor dokáže nalézt i několik dalších možných instancí jedné třídy, většinou ale s nižší mírou jistoty.

Tento problém je v algoritmu vyřešen tím že ukládá pouze instance s mírou jistoty větší než 80%. Pokud by ale i přesto prediktor našel více instancí jedné třídy, které mají koeficient jistoty alespoň 80%, algoritmus vybere ten s největší jistotou. Pokud by měli dvě instance stejnou jistotu, vybere tu, kterou prediktor našel jako první.

Vstupní obrázek je poté ještě oříznut bounding boxem instance, který prediktor také vytvoří. Obrázek je tím zmenšený na velikost bounding boxu. Tato operace se provede pro všechny další nalezené instance (toto se týká pouze instancí rohů, pro kruhové výřezy se tato část neprovádí) a každý z těchto rohů je uložen do seznamu, který je výstupem metody. Druhým výstupem metody jsou i odpovídající bounding boxy a masky segmentace.

### 2.4.1 Měření stran desky

Jednou z metod měření, která byla použita, je měření rozměrů samotné desky. Metoda `calibrate()` spouští poté, co nalezne bounding boxy, metodu `measure_table()`. Tato metoda má za vstupy pole, ve kterém je uložený obrázek, seznam bounding boxů, které našla předchozí metoda, a segmentaci instancí.

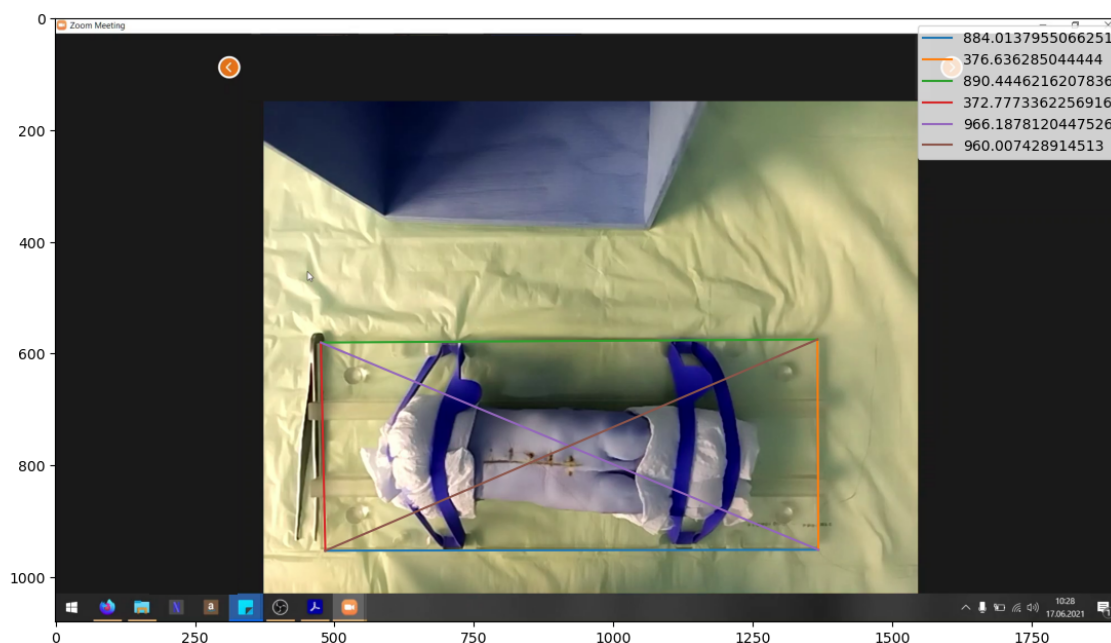
Bounding boxy mají vždy dva body, které ho definují. Jde o levý horní roh a pravý dolní roh. Oba tyto body mají určené souřadnice na ose  $x$  a  $y$ . Díky tomu že jsou známé tyto souřadnice, a také je známo ke které třídě tento bounding box patří,

můžeme určit body, které označují rohy podložky. Vzdálenost mezi body je zjištěna pomocí Pythagorovy věty.

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

Tímto způsobem je změřena vrchní, spodní, levá a pravá strana a také obě diagonály (viz obrázek 23). Pokud by nastala situace, že by některý z rohů nebyl vůbec nalezen, probíhá výpočet bez něj, ale nebudou zahrnuty výpočty strany, ve kterých tento roh figuruje.

Reálné hodnoty desky jsou 320 mm pro delší stranu desky, 135 mm pro kratší stranu desky a 350 mm pro diagonály. S těmito hodnotami byly naměřené hodnoty porovnány, výsledky jsou k nalezení v tabulce 6

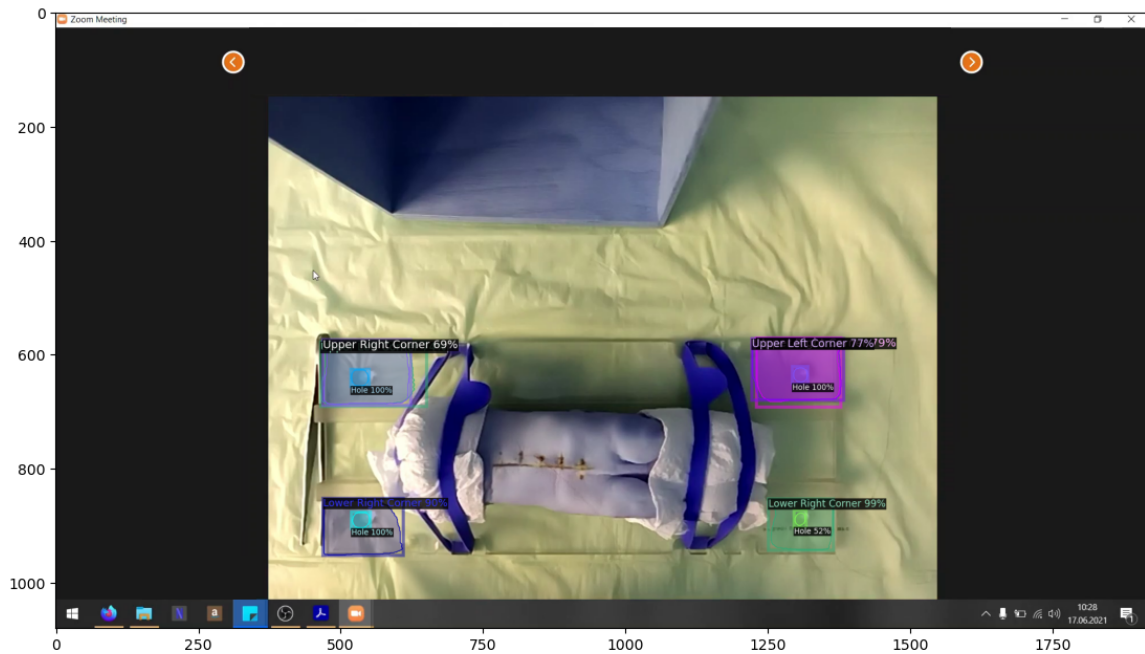


Obrázek 23: Úspěšné měření pomocí metody hledání rohů bounding boxů.

Tento způsob funguje poměrně dobře, ale pouze v případě, že je k dispozici dostatek nezakrytých rohů, ale především musí být deska položena co nejvíce souměrně s kamerou. Pokud bude totiž deska nakloněna na některou stranu, bounding boxy nebudou správně sedět na rohy.

## 2.4.2 Měření poloměru kruhových výřezů na desce

Další možností, jak k problému přistoupit, je změřit poloměr kruhových výřezů na desce. tyto výřezy se totiž dají spolehlivě identifikovat i při natočení desky na stole. Pro řešení tohoto problému bylo využito opět neuronových sítí, ale byla otestována i metoda klasického počítačového vidění, Houghova transformace.



Obrázek 24: Nalezení kruhových výřezů upraveným prediktorem, který byl přetrénovaný.

Pro úlohu vyhledání výřezů byl upraven již dříve použitý model neuronové sítě Faster R-CNN (R50 FPN). Bylo potřeba také doplnit anotace výřezů do původního datasetu. Na obrázku 24 je výstup z prediktora. V případě nalezení výřezu prediktorem se změří obsah nalezeného výřezu a pomocí vzorce pro výpočet obsahu kruhu se vypočítá poloměr výřezu:

$$r = \sqrt{\frac{S}{\pi}}$$

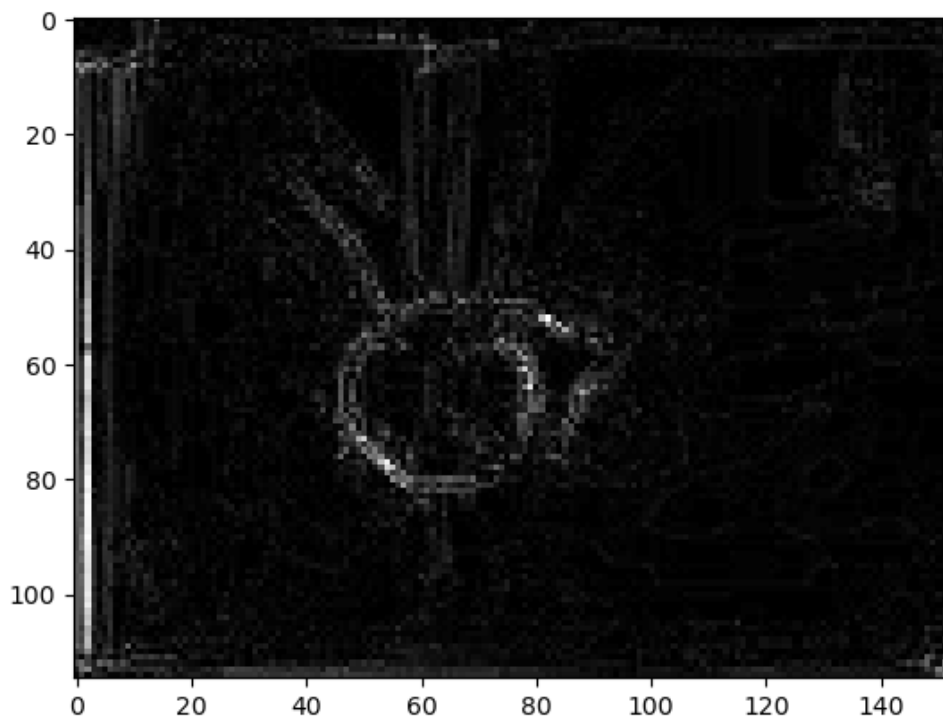
kde  $r$  je poloměr v pixelech a  $S$  obsah kruhového výřezu v pixelech. Obsah  $S$  je definován jako počet pixelů v segmentaci výřezu.



Alternativně lze také použít velikost jedné ze stran bounding boxu, který výřez obklopuje. Toto ale může být problematické, protože prediktor může bounding box vytvořit jako obdélník a výřez segmentovat jako elipsu.

Druhý způsob, který byl zvolen pro nalezení kruhových výřezů je Houghova transformace. Nejprve je potřeba vytvořit ořezy obrazu, ty získáme pomocí bounding boxů z metody `find_corners()`. Provádět Houghovu transformaci na menším obraze je důležité, protože je poté běh o mnoho rychlejší než kdyby se použila na celý obraz. V bounding boxu také předpokládáme výskyt právě jednoho kruhového výřezu a to usnadňuje výběr správné kružnice nalezené Houghovou transformací. Dále je potřeba tento na tento oříznutý obrázek aplikovat hranový detektor (na obrázku 25), v tomto případě do Laplaceův operátor, který byl v algoritmu implementován z knihovny Skimage. Laplaceův operátor detekuje hrany pomocí druhé derivace obrazu. Ta představuje prudké změny jasu, které jsou jedním z příznaků hran. Definice Laplaceova operátoru, který je popsán v [46]:

$$Laplace(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$



Obrázek 25: Obrázek oříznutý podle bounding boxu po průchodu Laplaceovým operátorem

Houghova transformace, popsaná v [47], je technika, kterou lze použít k izolaci prvků určitého tvaru v rámci obrazu. Protože vyžaduje, aby požadované vlastnosti byly specifikovány v nějaké parametrické formě, klasická Houghova transformace se nejčastěji používá pro detekci pravidelných křivek, jako jsou čáry, kružnice, elipsy atd. Zobecněnou Houghovu transformaci lze použít v aplikacích, kde není možný jednoduchý popis prvků. Houghův algoritmus je poměrně výpočetně náročný, ale navzdory omezením si zachovává mnoho aplikačních oblastí, protože většina vyráběných dílů (a mnoho anatomických částí zkoumaných v lékařských snímcích) obsahuje hranice prvků, které lze popsat pravidelnými křivkami. Hlavní výhodou techniky Houghovy transformace je to, že je tolerantní k mezerám v popisech hranic prvků a je relativně neovlivňována obrazovým šumem. Parametrická rovnice Hou-

ghovy transformace pro hledání kruhů je:

$$(x - a)^2 + (y - b)^2 = r^2$$

kde  $a$  a  $b$  jsou souřadnice středu kružnice a  $r$  je poloměr. Algoritmus postupně prochází všechny body, které by mohli být středem kružnice o zadaném poloměru, a počítá hodnoty jasů po obvodu kružnice. Jejich součet se zapíše na místo pixelu středu kružnice. Místo, kde je nejvyšší součet jasů (součet jasů hran, které předtím našel hranový detektor) je poté prohlášen středem kružnice. Algoritmus také může najít několik možných kružnic s nejvyšším jasem. Pro kruhovou Houghovu transformaci bylo využito metody `hough_circle()` z knihovny `Skimage`. Aby bylo dosaženo co nejrychlejšího chodu programu, byl omezen počet možných poloměrů na 20 - 40% vertikálního rozlišení. Příklad je na obrázku 26.

V obou případech je nalezený poloměr poté porovnán s reálnou hodnotou poloměru kruhového výřezu, která je 5.5 mm. Takto získáme výsledný poměr pixel/milimetr.



Obrázek 26: Houghova transformace najde několik kružnic s nejvyšším součtem jasů, v tomto případě 3.

## 2.5 Výsledky experimentů

### 2.5.1 Přesnost a rychlost trénování neuronové sítě

Pomocí výpočetních kapacit Metacentra byly natrénovány čtyři modely sítě Mask R-CNN. Pro obě architektury s různým počtem vrstev byly natrénovány dva modely, jeden s mírou učení 0.0005 a druhý s mírou učení 0.001. Pro zobrazení trénovacích statistik byl použit software TensorBoard. TensorBoard je sada vizualizačních nástrojů pro experimentování se strojovým učením. TensorBoard umožňuje sledování a vizualizaci metrik, jako je ztráta a přesnost nebo vizualizace grafu modelu. TensorBoard.

Do tabulky 2 byly zaznamenané hodnoty přesností obou částí modelu. Faster

Architektura	LR	Faster R-CNN accuracy	Maks R-CNN accuracy
R-CNN R50	0.0005	0.9897	0.9747
R-CNN R50	0.001	0.9894	0.9787
R-CNN R101	0.0005	0.9842	0.9623
R-CNN R101	0.001	0.9861	0.971

Tabulka 2: Porovnání přesností jednotlivých modelů, LR - learning rate

R-CNN zajišťuje detekci a klasifikaci. Mask R-CNN zajišťuje segmentaci instancí. Trénování mělo ve všech případech velmi podobný průběh i finální přesnost. Nejlepších výsledků dosáhl model s 50 skrytými vrstvami a koeficientem učení 0.001. Proto byl tento model zvolen pro predikce.

### 2.5.2 Porovnání kalibračních metod

Pro porovnávání bylo náhodně vybráno 31 snímků z datasetu, který je novější než dataset, který byl použitý pro trénování. Tyto snímky byly ručně změřeny a byl proveden výpočet převádějící naměřené hodnoty v pixelech na poměr pixelů na milimetr. Tyto hodnoty byly označeny jako ground truth, což znamená že je možné je považovat za pravdivé a přesné. Od těchto hodnot se bude odvíjet naměřená přesnost ostatních metod. Ground truth hodnoty jsou v tabulce 8.

Kromě metod navržených v této práci bylo použito k vyhodnocení i metody aktuálně používané v aplikaci pro měření trajektorií. Tou je měření pomocí čtení QR kódu, který je na každém z použitých snímků. Byl použit mírně upravený původní kód, který je stavěný na čtení celého videa, ale princip se nijak nemění. Algoritmus se pokouší v obrazu najít QR kód, pokud ho najde, přečte z něj informace. Informace obsažené v QR kódu obsahují i jeho velikost v milimetrech. Algoritmus také dokáže změřit velikost strany QR kódu v pixelech. Podílem těchto dvou hodnot se v původní metodě kalibrace získává kalibrační poměr. Algoritmus postupně procházel všech 31 testovacích obrázků a aplikoval na ně postupně predikci rohů, poté se spustila metoda na měření stran desky a kruhových výřezů. Následovala metoda Houghovy transformace a na konec se spustil původní algoritmus pro čtení QR kódu. Navíc byla také měřena doba běhu všech metod pro každý obrázek. Výsledky byly zaznamenány do tabulek 6 a 7. Pro získání přesnosti bylo potřeba výsledky porovnat s ground truth hodnotami. Výsledná chyba se získá jako rozdíl naměřených hodnot a ground truth hodnoty. Výsledné chyby jsou zaznamenány v tabulce 5. Při pohledu na tabulku 3, která obsahuje průměrné hodnoty chyby, můžeme vidět, že se jako nejpřesnější metoda ukazuje metoda nalezení kruhových výřezů a následný výpočet průměru. Druhá

nejpřesnější metoda je podle průměrné chyby Houghova transformace. Průměrnou chybu větší než 1 má metoda měření stran desky. Jako nejnepřesnější se zde ukazuje dosud používaná metoda měření pomocí čtení QR kódu.

Metoda	Průměrná chyba
QR	-1.716
Radius NN	-0.047
Measurements	1.031
Hough Transform	-0.846

Tabulka 3: Průměrné chyby jednotlivých metod

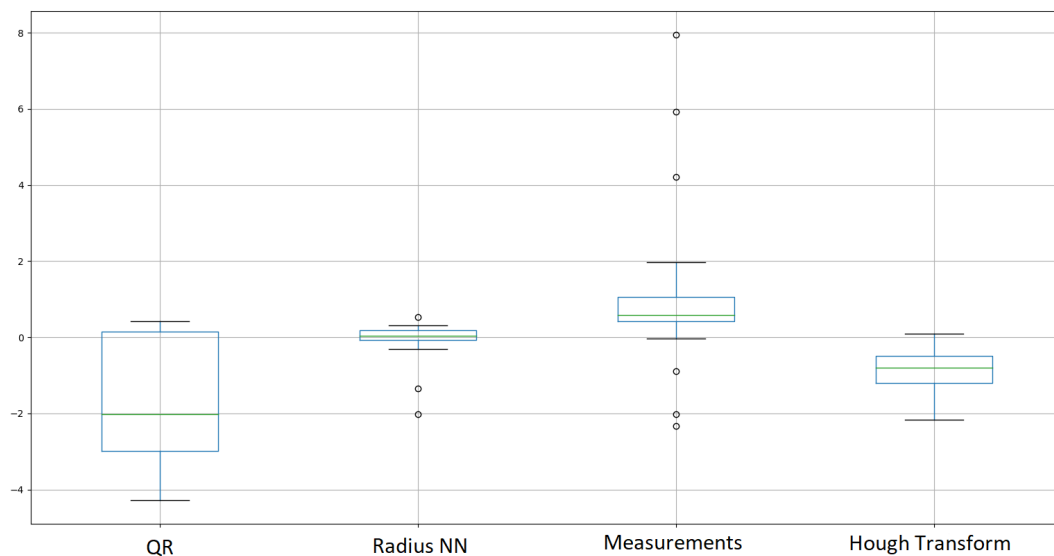
Co se časové náročnosti týče, průměrná doba trvání jednotlivých metod je v tabulce 4. Metoda měření desky `measure_table()`, která obsahuje měření stran i měření kruhových výřezů je sama o sobě nejrychlejší, jenže pro svůj běh potřebuje informace o poloze rohů z metody `find_corners()`. Stejně informace využívá i metoda pro Houghovu transformaci `hough_transform()`. Časová náročnost Houghovy transformace roste z rozlišením výřezu, protože musí testovat více průměrů na více pixelech. Nejrychlejší metoda je tedy měření QR kódem.

Metoda	Čas běhu [s]
<code>find_corners()</code>	1.616299
<code>measure_table()</code>	0.019005
<code>hough_transform()</code>	5.437360
QR code read	0.074268

Tabulka 4: Průměrné časy běhu použitých metod v sekundách

## 2.6 Diskuze výsledků

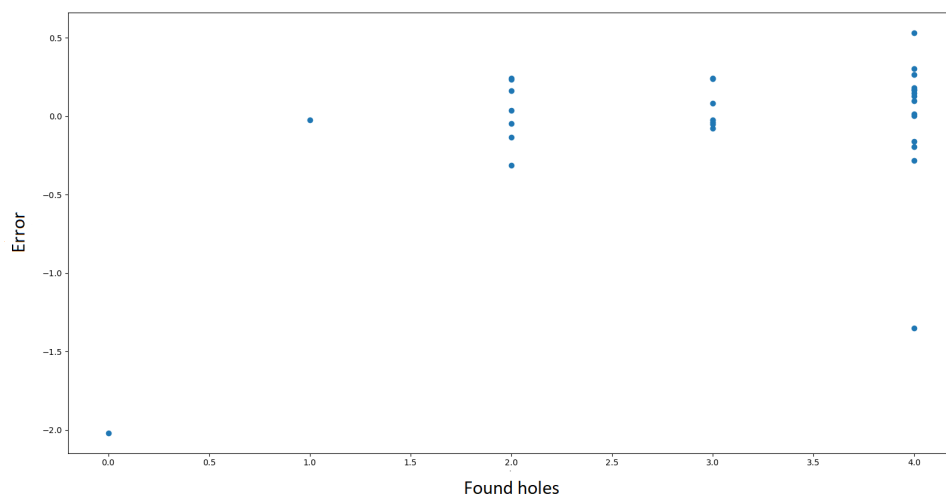
Přesnější analýzu výsledků umožňuje pohled na distribuci chyby ( obrázek 27). Na tomto obrázku je metoda pro grafické znázornění skupin číselných dat prostřednictvím jejich kvartilů. Rámeček sahá od kvartilových hodnot dat Q1 do Q3 s čarou na mediánu (Q2). Linie vyčnívají z okrajů rámečku a ukazují rozsah dat. Ve výchozím nastavení nepřesahují více než  $1,5 * IQR$  ( $IQR = Q3 - Q1$ ) od okrajů rámečku a končí v nejvzdálenějším datovém bodu v tomto intervalu. Odlehlé hodnoty jsou vyneseny jako samostatné tečky.



Obrázek 27: Distribuce výsledných chyb měření

Na obrázku 27 se nabízí jiný pohled na rozložení chyb v kontextu všech měření. Metoda měření pomocí QR kódu je poměrně přesná, ale to pouze za předpokladu, že hledaný QR kód vůbec najde. V 52,7% případů při měření testovací množiny dosud používaná metoda nenalezla QR kód, i přes to že je objektivně nečitelný pouze na jednom z nich.

Měření pomocí výřezů nalezených neuronovou sítí i zde obhájí velmi dobré výsledky, v případě když prediktor nemůže nalézt ani jeden z výřezů, což se stalo v celém trénovacím datasetu pouze jednou, nemůže metoda dobře fungovat.

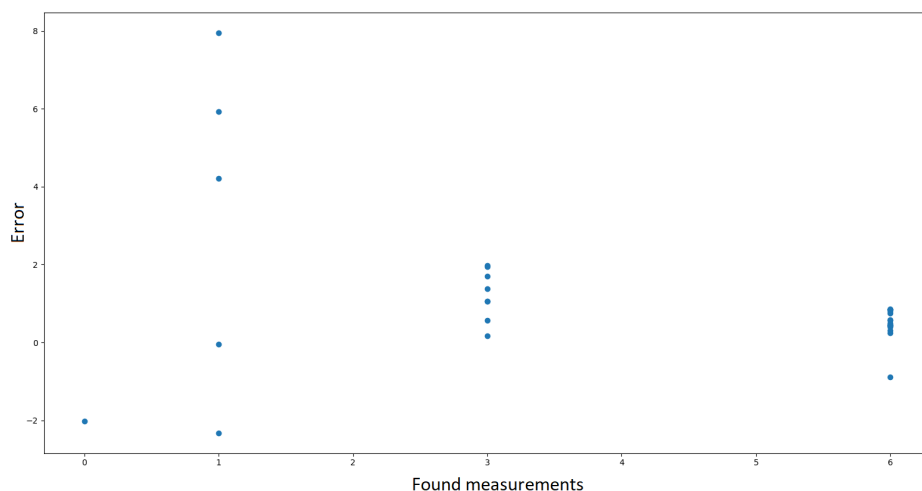


Obrázek 28: Poměr naměřené chyby a nalezených výřezů

Na obrázku 28 je vidět, že počet nalezených děr není zárukou zvýšené přesnosti této metody.

Naopak metoda měření stran ukazuje výrazné zlepšení se stoupajícím počtem nalezených rohů, a tím pádem i změřených stran, což je vidět na obrázku 29. Při změření většího počtu stran se chyba kompenzuje průměrem hodnot.

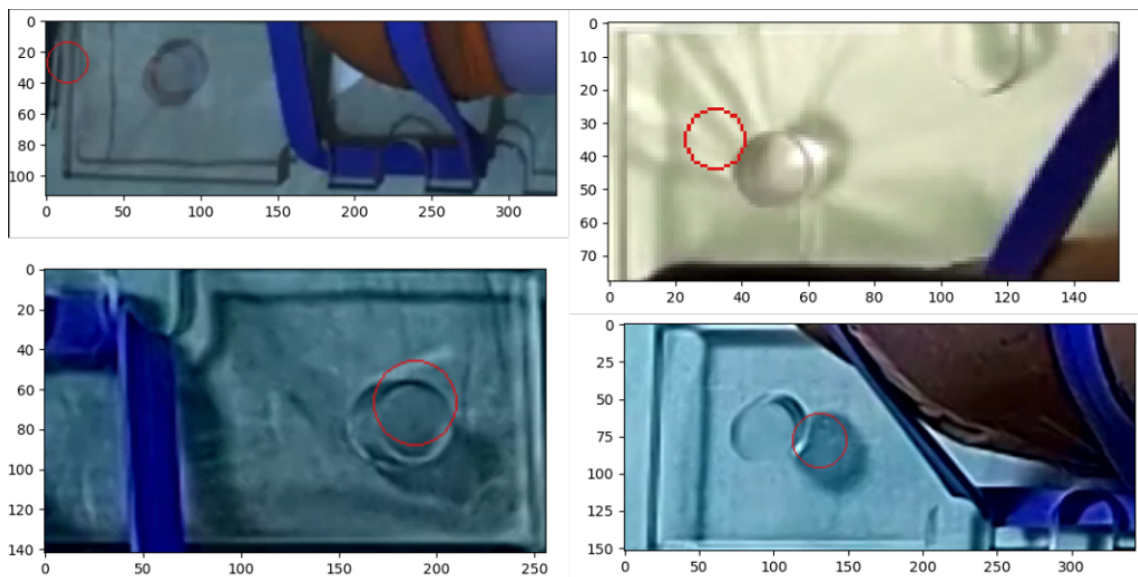




Obrázek 29: Poměr naměřené chyby a změřených stran

Tato metoda je nepřesná hlavně z důvodu měření podle rohů bounding boxů. Ty nebývají nalezené správně, a tím pádem nemůže být měření v těchto případech přesné. Tato chyba se nejvíce projevuje v případě, že je deska natočena. Příklady povedeného i nepovedeného měření jsou na obrázcích 31 a 32.

Metoda měření pomocí Houghovy transformace nalezne nějakou kružnici téměř vždy, ovšem to nestačí na to aby byla metoda přesná. V algoritmu je omezená velikost průměru kruhu od 20% do 40% vertikálního rozlišení, a to hlavně z důvodu zrychlení běhu programu. Tato velikost byla také experimentálně zjištěna jako neoptimálnější pro každý z výřezů. Bohužel kvůli kombinaci nízkého rozlišení videa a tím pádem špatně fungujícího hranového detektoru nalezne výřezy často úplně jinde, než ve skutečnosti jsou, jako je vidět na obrázku 30. Tento problém by se nejspíše dal vyřešit zneprůhledněním podložní desky a tím pádem lepším gradientem jasů pro hranový detektor.



Obrázek 30: Příklady Houghovy transformace. Červenou barvou jsou transformací nalezené kruhy. Metoda je poměrně nepřesná, obě měření na horních obrázcích jsou špatně a na pravém dolním obrázku metoda označila jako kruh stín výřezu.

Jako nejlepší z navržených metod kalibrace se ukázala metoda měření kruhových výřezů nalezených pomocí neuronové sítě. Závěrem vyhodnocení je nové nalezení metody, která má srovnatelnou přesnost určení správného poměru pixelů na milimetr, s o mnoho lepší spolehlivostí za kterou platí pomalejším během ve srovnání s dosud používanou metodou kalibrace pomocí měření QR kódu. Kód kalibračního algoritmu je uložen v repositáři na GitHub [48].

### 3 Závěr

V lékařství se čím dál častěji používají zařízení pro videozáznam prováděných procedur, a to i pro potřeby vzdělávání. V době koronavirové pandemie byla prezenční výuka na většině škol nahrazena distanční výukou za pomoci informačních technologií. Lékařské obory jsou ale z velké části postaveny na praktické výuce o kterou takto přišli. Fakultní nemocnice v Jeně se proto snažila tuto ztracenou výuku nahradit online videokonferencemi a dále rozvíjet možnosti distanční výuky. Ve spolupráci s Fakultou aplikovaných věd ZČU a Biomedicínského centra LF UK v Plzni se pokusili o automatizaci hodnocení studentů při tréninku šití ran.

Poskytnutý dataset se skládal ze záznamů videokonferencí, při kterých byli studenti online připojeni a prováděli požadované úkony. Přesnost těchto úkonů by měl automaticky hodnotit algoritmus. Pro přesnost výsledků tohoto algoritmu je ale potřeba kalibrace, získání informací o reálných rozměrech scény, kterou by měl zpracovávat.

Cílem této práce bylo navrhnout lepší kalibrační mechanismus obrazu, než dosud využívaný, který kalibroval obraz čtením QR kódu a následně ho měřil. Tento způsob byl ale velmi nespolehlivý. Způsob kalibrace navržený v této práci se opírá o znalost reálné velikosti konkrétního objektu na záznamech. Tímto objektem se stala podložní deska, na které je umístěn objekt lékařské procedury.

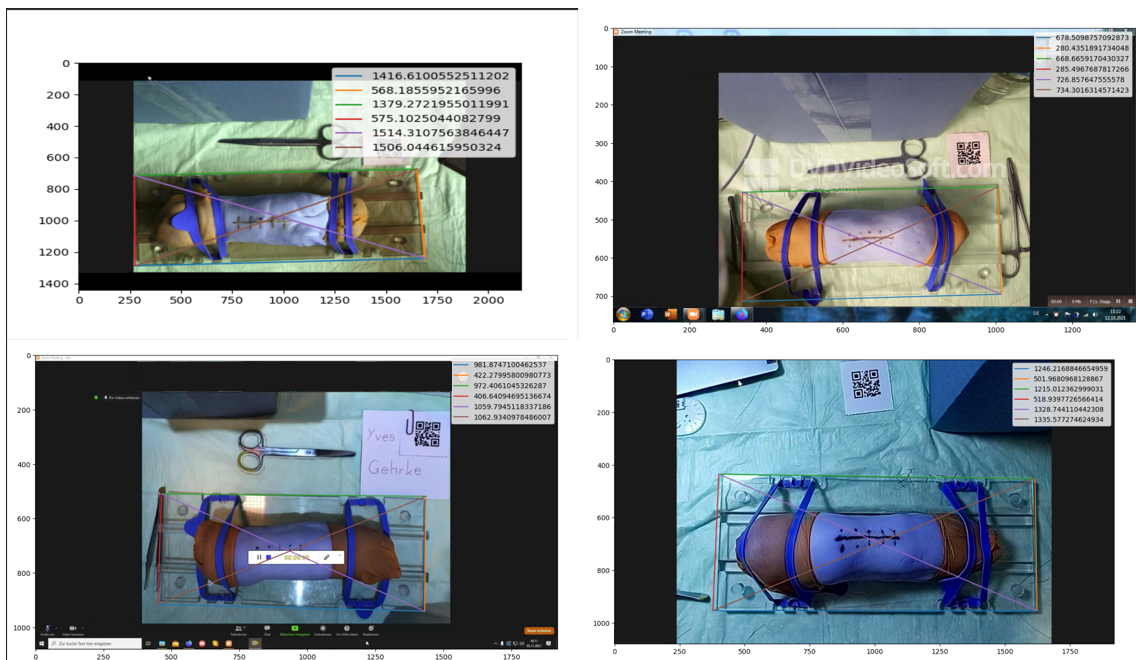
Prostřednictvím anotování 24609 snímků bylo umožněno trénování neuronové sítě. Do snímků byly zaneseny informace o poloze rohů podložní desky a kruhových výřezů. Jako model neuronové sítě byl zvolen Mask R-CNN schopný segmentace instancí. Tento model byl natrénovaný v několika variantách počtu skrytých vrstev a koeficientu učení. Trénování probíhalo na výpočetních uzlech Metacentra. Na základě výsledků trénování byl zvolen model s 50 skrytými vrstvami a koeficientem učení 0,001.

Po natrénování modelu neuronové sítě byly navrženy tři metody kalibrace. Jednou z nich je Houghova transformace pro hledání kruhových výřezů, která je jednou z klasických metod počítačového vidění. Další dvě metody používají informace získané od prediktoru natrénované neuronové sítě. První metoda měří strany a diagonály podložní desky a druhá metoda získává obsah pixelů v segmentaci kruhových výřezů, ze které pak získává informace o poloměru. Všechny tyto metody mají jako výstup poměr pixel/milimetr.

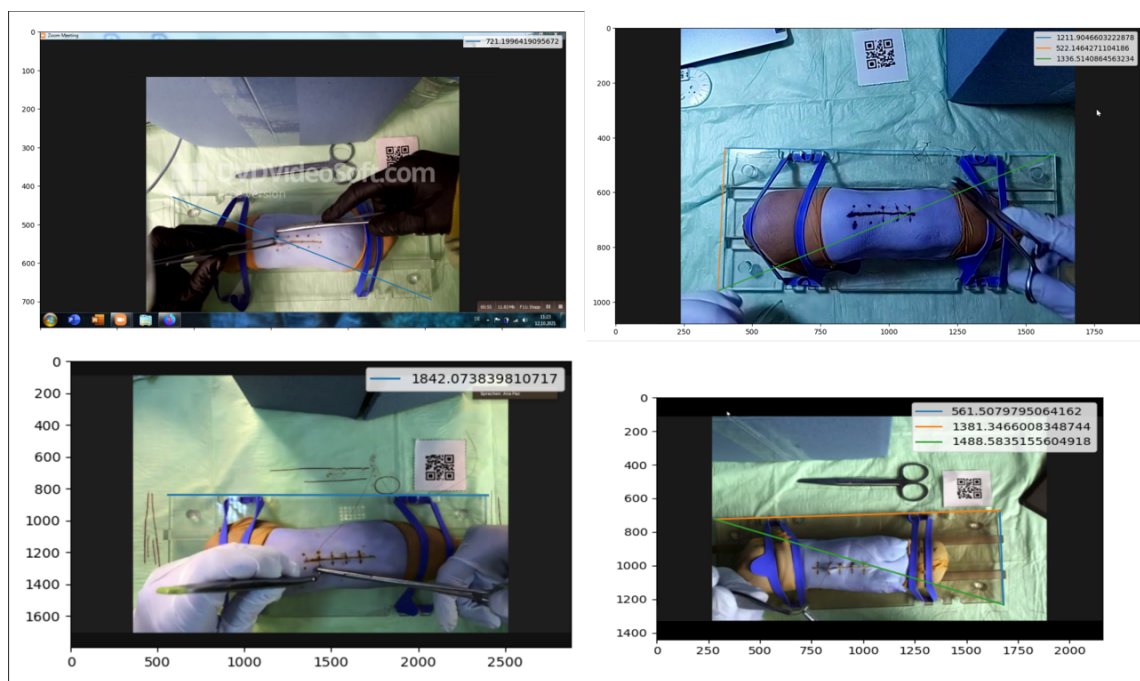
Nové metody byly porovnány se stávajícím řešením a bylo zjištěno, že metoda predikce a segmentace kruhových výřezů je velice přesná a o mnoho spolehlivější než metoda měření pomocí QR kódu.

## 4 Přílohy

### 4.1 Obrázky



Obrázek 31: Příklady obstojně změřených stran



Obrázek 32: Příklad měření s nenalezenými rohy

## 4.2 Tabulky

	QR	Radius NN	FH	Measurements	FM	Hough
0	-2.960000	0.147662	4	0.484681	6	-0.596364
1	0.425062	0.181780	4	0.864820	6	-0.929091
2	-2.060000	0.037747	2	0.313219	6	-0.241818
3	-0.009725	0.166685	4	0.579919	6	-0.467273
4	-2.830000	0.304362	4	0.453925	6	-0.830000
5	-0.078400	-0.036978	3	1.704478	3	-0.031818
6	0.236042	0.163640	2	7.954991	1	-2.053636
7	-2.100000	-0.135160	2	-0.039430	1	-0.463636
8	0.131285	0.097351	4	0.431549	6	-0.476364
9	0.368030	0.240832	3	1.373335	3	-1.312727
10	-4.250000	-0.045592	2	5.934487	1	-1.522727
11	-4.270000	0.241567	3	1.940551	3	-0.815455
12	0.236389	0.014003	4	0.850409	6	-1.326364
13	0.206389	0.265087	4	0.840835	6	0.098182
14	0.226389	0.002596	4	0.840336	6	-0.245455
15	0.010275	-0.023955	3	1.052769	3	-1.174545
16	0.183423	-0.194888	4	0.757331	6	-0.507273
17	0.017051	-0.074952	3	1.053719	3	-1.204545
18	0.158219	0.128513	4	0.407066	6	-0.486364
19	-4.120000	-0.311123	2	0.162252	3	-1.210909
20	-1.188715	-1.350095	4	-0.884925	6	-2.160000
21	-2.830000	-0.160943	4	0.426530	6	-1.011818
22	-2.030000	-0.025094	1	0.573702	3	-0.393636
23	-4.280000	-0.280743	4	0.560452	6	-0.643636
24	-4.250000	-0.050744	3	1.974059	3	-1.340909
25	-2.020000	-2.020000	0	-2.020000	0	-0.565455
26	-2.040000	0.081825	3	0.251665	6	-0.585455
27	-4.070000	0.241552	2	-2.325757	1	-0.979091
28	-4.040000	0.169077	4	0.799681	6	-1.312727
29	-2.990000	0.233524	2	4.205281	1	-0.808182
30	-2.990000	0.533215	4	0.440574	6	-0.626364

Tabulka 5: Tabulka chyb, FH - počet nalezených kruhových výřezů, FM - počet změřených stran

	QR	Radius NN	Measurements	Hough Transform
0	0.000000	3.107662	3.444681	2.363636
1	4.445062	4.201780	4.884820	3.090909
2	0.000000	2.097747	2.373219	1.818182
3	3.730275	3.906685	4.319919	3.272727
4	0.000000	3.134362	3.283925	2.000000
5	3.771600	3.813022	5.554478	3.818182
6	5.926042	5.853640	13.644991	3.636364
7	0.000000	1.964840	2.060570	1.636364
8	2.971285	2.937351	3.271549	2.363636
9	4.408030	4.280832	5.413335	2.727273
10	0.000000	4.204408	10.184487	2.727273
11	0.000000	4.511567	6.210551	3.454545
12	5.926389	5.704003	6.540409	4.363636
13	5.926389	5.985087	6.560835	5.818182
14	5.926389	5.702596	6.540336	5.454545
15	3.730275	3.696045	4.772769	2.545455
16	5.963423	5.585112	6.537331	5.272727
17	3.767051	3.675048	4.803719	2.545455
18	3.008219	2.978513	3.257066	2.363636
19	0.000000	3.808877	4.282252	2.909091
20	2.971285	2.809905	3.275075	2.000000
21	0.000000	2.669057	3.256530	1.818182
22	0.000000	2.004906	2.603702	1.636364
23	0.000000	3.999257	4.840452	3.636364
24	0.000000	4.199256	6.224059	2.909091
25	0.000000	0.000000	0.000000	1.454545
26	0.000000	2.121825	2.291665	1.454545
27	0.000000	4.311552	1.744243	3.090909
28	0.000000	4.209077	4.839681	2.727273
29	0.000000	3.223524	7.195281	2.181818
30	0.000000	3.523215	3.430574	2.363636

Tabulka 6: Naměřené hodnoty, poměr pixely na milimetr

	<code>find_corners()</code>	<code>measure_table()</code>	<code>hough_transform()</code>	QR code read
0	3.978746	0.023590	4.210107	0.056540
1	1.592907	0.022389	9.175574	0.088694
2	1.358875	0.013362	1.339325	0.036460
3	1.462412	0.021512	7.819669	0.092431
4	1.469479	0.021411	2.877660	0.053285
5	1.455585	0.017042	5.512745	0.086218
6	1.637434	0.015931	14.495611	0.118104
7	1.342880	0.008655	1.003879	0.035409
8	1.477722	0.020015	2.223881	0.056482
9	1.559178	0.019229	5.090645	0.088476
10	1.556566	0.013557	4.860975	0.081383
11	1.517965	0.021171	5.425231	0.079068
12	1.882674	0.029116	13.837760	0.123778
13	1.785412	0.030689	14.076791	0.125168
14	1.775742	0.030199	14.446300	0.128100
15	1.439855	0.016464	4.522768	0.066921
16	1.757660	0.028865	14.701767	0.129091
17	1.484619	0.016167	4.501644	0.062884
18	1.495495	0.021973	1.575205	0.057583
19	1.556507	0.013522	5.920710	0.060177
20	1.503707	0.020797	1.662431	0.056706
21	1.481910	0.021683	1.560688	0.054340
22	1.343433	0.007953	0.639920	0.034799
23	1.555882	0.021222	5.492022	0.086374
24	1.557417	0.018969	5.488962	0.085274
25	1.362700	0.002509	0.714477	0.036594
26	1.425297	0.014706	0.862635	0.039229
27	1.595087	0.014705	5.560287	0.089374
28	1.693157	0.028831	4.880721	0.077146
29	1.469845	0.011074	1.862976	0.060218
30	1.529130	0.021848	2.214788	0.056000

Tabulka 7: Doby trvání běhu jednotlivých metod v sekundách



Číslo snímku	Ground truth [pix/mm]
0	2.96
1	4.02
2	2.06
3	3.74
4	2.83
5	3.85
6	5.69
7	2.10
8	2.84
9	4.04
10	4.25
11	4.27
12	5.69
13	5.72
14	5.70
15	3.72
16	5.78
17	3.75
18	2.85
19	4.12
20	4.16
21	2.83
22	2.03
23	4.28
24	4.25
25	2.02
26	2.04
27	4.07
28	4.04
29	2.99
30	2.99

Tabulka 8: Ručně naměřené ground truth hodnoty jednotlivých snímků

## Seznam obrázků

1	Snímky z CT skenu, na který byly použity metody počítačového vidění pro nalezení nádorů jater, převzato z [6] . . . . .	2
2	Záběry při operaci klasickým způsobem za použití kamery na hlavovém nástavci, převzato z [11] . . . . .	4
3	Bayerův filtr, převzato z [17] . . . . .	7
4	Segmentace ve snímku z CT břišní dutiny, červeně játra, zeleně žaludek, žlutě slezina, převzato z [6] . . . . .	8
5	Detekce hran, převzato z [21] . . . . .	9
6	Aplikace počítačového vidění pro detekci rakoviny kůže, převzato z [24]	11
7	Radiální zkreslení - bez zkreslení, pozitivní radiální zkreslení, negativní radiální zkreslení, převzato z [28] . . . . .	12
8	Model perceptronu, převzatý z [32] . . . . .	14
9	Binární skoková funkce [32] . . . . .	15
10	Sigmoidální funkce [32] . . . . .	16
11	Funkce ReLU [32] . . . . .	17
12	Struktura neuronové sítě [32] . . . . .	18
13	Model sítě s konvoluční a pooling vrstvou, převzato z [36] . . . . .	21
14	Použití Mask R-CNN pro bezpečnost v okolí jeřábů a pro měření morfologických vlastností ryb, převzato z [37] a [38] . . . . .	22
15	Plán řešení . . . . .	23
16	Pohled na obrazovku zaznamenávající videokonferenci . . . . .	25
17	Scéna ze které jsou pořízeny snímky. . . . .	26
18	Obrázek 2.3: Anotáčnický software CVAT . . . . .	28
19	Schéma sítě mask R-CNN převzato z [42] . . . . .	31
20	Schéma sítě ResNet s FPN převzato z [44] . . . . .	33
21	Zarovnávání bounding boxu pomocí ROIAlign, převzato z [45] . . . . .	34
22	Snímek který prošel prediktorem, který na něm provedl segmentaci instancí. . . . .	36
23	Úspěšné měření pomocí metody hledání rohů bounding boxů. . . . .	38
24	Nalezení kruhových výřezů upraveným prediktorem, který byl přetrénovaný. . . . .	39
25	Obrázek oříznutý podle bounding boxu po průchodu Laplaceovým operátorem . . . . .	41
26	Houghova transformace najde několik kružnic s nejvyšším součtem jasů, v tomto případě 3. . . . .	43
27	Distribuce výsledných chyb měření . . . . .	46

28	Poměr naměřené chyby a nalezených výřezů . . . . .	47
29	Poměr naměřené chyby a změřených stran . . . . .	48
30	Příklady Houghovy transformace. Červenou barvou jsou transformací nalezené kruhy. Metoda je poměrně nepřesná, obě měření na horních obrázcích jsou špatně a na pravém dolním obrázku metoda označila jako kruh stín výřezu. . . . .	49
31	Příklady obstojně změřených stran . . . . .	51
32	Příklad měření s nenalezenými rohy . . . . .	52

## Seznam tabulek

1	Porovnání rychlosti běhu trénování . . . . .	36
2	Porovnání přesností jednotlivých modelů, LR - learning rate . . . . .	44
3	Průměrné chyby jednotlivých metod . . . . .	45
4	Průměrné časy běhu použitých metod v sekundách . . . . .	45
5	Tabulka chyb, FH - počet nalezených kruhových výřezů, FM - počet změřených stran . . . . .	53
6	Naměřené hodnoty, poměr pixely na milimetr . . . . .	54
7	Doby trvání běhu jednotlivých metod v sekundách . . . . .	55
8	Ručně naměřené ground truth hodnoty jednotlivých snímků . . . . .	56

## Reference

- [1] S. Silverstein, “What is medical informatics, and why is it an important specialty? [online],” 2000.
- [2] E. SHORTLIFFE, “The structure of medical informatics,” *Med Inform*, vol. 9, 1984.
- [3] A. Shachak and S. Reis, “The impact of electronic medical records on patient-doctor communication during consultation: A narrative literature review,” *Journal of Evaluation in Clinical Practice*, vol. 15, 2009.
- [4] A. H. Jobe, “Amplitude integrated eeg-the future?,” *Journal of Pediatrics*, vol. 154, 2009.
- [5] S. Vulliemoz, R. Thornton, R. Rodionov, D. W. Carmichael, M. Guye, S. Lhatoo, A. W. McEvoy, L. Spinelli, C. M. Michel, J. S. Duncan, and L. Lemieux, “The

- spatio-temporal mapping of epileptic networks: Combination of eeg-fmri and eeg source imaging,” *NeuroImage*, vol. 46, 2009.
- [6] M. Moghbel, S. Mashohor, R. Mahmud, and M. I. B. Saripan, “Automatic liver tumor segmentation on computed tomography for patient treatment planning and monitoring,” *EXCLI Journal*, vol. 15, 2016.
- [7] E. Mittra and A. Quon, “Positron emission tomography/computed tomography: The current technology and applications,” 2009.
- [8] P. Pergami, “Pdas in clinical practice: Having a database in your hand but keeping the decision in your brain,” 2003.
- [9] A. P. Leshner and S. R. Shah, “Telemedicine in the perioperative experience,” *Seminars in Pediatric Surgery*, vol. 27, 2018.
- [10] A. J. Larner, “Teleneurology: An overview of current status,” 2011.
- [11] A. Nair, S. Kamal, T. Dave, K. Mishra, H. Reddy, D. Rocca, R. D. Rocca, A. Andron, and V. Jain, “Surgeon point-of-view recording: Using a high-definition head-mounted video camera in the operating room,” *Indian Journal of Ophthalmology*, vol. 63, 2015.
- [12] J. A. Hoschtitzky, D. B. Trivedi, and M. J. Elliott, “Saved by the video: Added value of recording surgical procedures on video,” *Annals of Thoracic Surgery*, vol. 87, 2009.
- [13] J. Zheng, J. Wang, and Y. Li, “I am your eyes—the reflection of being a camera-holder in laparoscopic gastrointestinal surgery,” *Annals of Laparoscopic and Endoscopic Surgery*, vol. 2, 2017.
- [14] V. Velanovich, “Laparoscopic vs open surgery,” *Surgical Endoscopy*, vol. 14, 2000.
- [15] Wikipedia, “Rgb color model,” 2016.
- [16] R. Hain, C. J. Kähler, and C. Tropea, “Comparison of ccd, cmos and intensified cameras,” *Experiments in Fluids*, vol. 42, 2007.
- [17] B. Barbosa, A. J. R. Neves, S. C. Soares, and I. D. Dimas, “Analysis of emotions from body postures based on digital imaging,” *SIGNAL 2018: The Third International Conference on Advances in Signal, Image and Video Processing*, vol. 10, 2018.

- [18] L. G. Shapiro, “Computer vision: the last 50 years,” 2020.
- [19] X. Liu, Z. Deng, and Y. Yang, “Recent progress in semantic image segmentation,” *Artificial Intelligence Review*, vol. 52, 2019.
- [20] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, 1986.
- [21] S. Dhar, S. Alam, M. Santra, P. Saha, and S. Thakur, “A novel method for edge detection in a gray image based on human psychovisual phenomenon and bat algorithm,” 2017.
- [22] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive into Deep Learning Release 0.17.5*, vol. 17. 2021.
- [23] T. Pun, G. Gerig, and O. Ratib, “Image analysis and computer vision in medicine,” *Computerized Medical Imaging and Graphics*, vol. 18, 1994.
- [24] M. Lewis, “An artificial intelligence tool that can help detect melanoma,” 2021.
- [25] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, 2000.
- [26] J. Heikkila and O. Silven, “Four-step camera calibration procedure with implicit image correction,” 1997.
- [27] J. S. Kim and K. S. Hong, “A recursive camera resectioning technique for off-line video-based augmented reality,” *Pattern Recognition Letters*, vol. 28, 2007.
- [28] S. Kaustubh, “Understanding lens distortion,” 2020.
- [29] J.-W. Lin, “Artificial neural network related to biological neuron network: a review,” *Advanced Studies in Medical Sciences*, vol. 5, 2017.
- [30] F. Amato, A. López, E. M. Peña-Méndez, P. Vañhara, A. Hampl, and J. Havel, “Artificial neural networks in medical diagnosis,” 2013.
- [31] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, 1958.
- [32] P. Baheti, “12 types of neural networks activation functions: How to choose?,” 5 2022.

- [33] M. Usama, J. Qadir, A. Raza, H. Arif, K. L. A. Yau, Y. Elkhatib, A. Hussain, and A. Al-Fuqaha, "Unsupervised machine learning for networking: Techniques, applications and research challenges," *IEEE Access*, vol. 7, 2019.
- [34] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," vol. 2018-January, 2018.
- [35] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, "Learning pooling for convolutional neural network," *Neurocomputing*, vol. 224, 2017.
- [36] C. C. Chatterjee, "Basics of the classic cnn," 2019.
- [37] Z. Yang, Y. Yuan, M. Zhang, X. Zhao, Y. Zhang, and B. Tian, "Safety distance identification for crane drivers based on mask r-cnn," *Sensors (Switzerland)*, vol. 19, 2019.
- [38] C. Yu, X. Fan, Z. Hu, X. Xia, Y. Zhao, R. Li, and Y. Bai, "Segmentation and measurement scheme for fish morphological features based on mask r-cnn," *Information Processing in Agriculture*, vol. 7, 2020.
- [39] Z. Hao, L. Lin, C. J. Post, E. A. Mikhailova, M. Li, Y. Chen, K. Yu, and J. Liu, "Automated tree-crown and height detection in a young forest plantation using mask region-based convolutional neural network (mask r-cnn)," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 178, 2021.
- [40] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," 2017.
- [41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, 2017.
- [42] R. Bharath, "An overview of human pose estimation with deep learning," 2019.
- [43] C. Zhang, P. Benz, D. M. Argaw, S. Lee, J. Kim, F. Rameau, J. C. Bazin, and I. S. Kweon, "Resnet or densenet? introducing dense shortcuts to resnet," 2021.
- [44] X. Li, Z. Xu, X. Shen, Y. Zhou, B. Xiao, and T. Q. Li, "Detection of cervical cancer cells in whole slide images using deformable and global context aware faster rcnn-fpn," *Current Oncology*, vol. 28, 2021.
- [45] G. Karara, R. Hajji, and F. Poux, "3d point cloud semantic augmentation: Instance segmentation of 360° panoramas by deep learning techniques," *Remote Sensing*, vol. 13, 2021.

- [46] X. Qi and X. Liu, “Evolution of the first eigenvalue of the laplace operator and the p-laplace operator under a forced mean curvature flow,” *Open Mathematics*, vol. 18, 2020.
- [47] R. O. Duda and P. E. Hart, “Use of the hough transformation to detect lines and curves in pictures,” *Communications of the ACM*, vol. 15, 1972.
- [48] <https://github.com/JRX99/camera-calibration>