



FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI

KATEDRA INFORMATIKY  
A VÝPOČETNÍ TECHNIKY



## Bakalářská práce

Analýza defektů dřeva z CT  
snímků

Daniel Hauser







FAKULTA APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ UNIVERZITY  
V PLZNI

KATEDRA INFORMATIKY  
A VÝPOČETNÍ TECHNIKY

## **Bakalářská práce**

Analýza defektů dřeva z CT  
snímků

Daniel Hauser

**Vedoucí práce**

Ing. Ladislav Lenc, Ph.D.

© Daniel Hauser, 2024.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

**Citace v seznamu literatury:**

HAUSER, Daniel. *Analýza defektů dřeva z CT snímků*. Plzeň, 2024. Bachelářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce Ing. Ladislav Lenc, Ph.D.

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd  
Akademický rok: 2023/2024

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Daniel HAUSER**  
Osobní číslo: **A20B0099P**  
Studijní program: **B0613A140015 Informatika a výpočetní technika**  
Specializace: **Informatika**  
Téma práce: **Analýza defektů dřeva z CT snímků**  
Zadávající katedra: **Katedra informatiky a výpočetní techniky**

## Zásady pro vypracování

1. Seznamte se s dodanými CT skeny dubové kulatiny.
2. Seznamte se s knihovnamy a nástroji pro zobrazení a zpracování dat v datovém standardu DICOM.
3. Prostudujte dostupné metody používané pro detekci vad dřeva.
4. Vyberte alespoň dvě vhodné metody, implementujte je a otestujte na dodaných datech.
5. Provedte srovnání metod, zhodnoťte dosažené výsledky a navrhněte možná rozšíření.

Rozsah bakalářské práce: **doporuč. 30 s. původního textu**  
Rozsah grafických prací: **dle potřeby**  
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Vedoucí bakalářské práce: **Ing. Ladislav Lenc, Ph.D.**  
Nové technologie pro informační společnost

Datum zadání bakalářské práce: **2. října 2023**  
Termín odevzdání bakalářské práce: **2. května 2024**

L.S.

---

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

---

**Doc. Ing. Přemysl Brada, MSc., Ph.D.**  
vedoucí katedry

V Plzni dne 25. října 2023

# Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Plzni dne 2. května 2024

.....

Daniel Hauser

## Abstrakt

Tato bakalářská práce se zaměřuje na analýzu defektů dřeva z CT snímků pomocí metod zpracování obrazu s úmyslem vyvinout efektivní způsob pro automatickou detekci. Tento přístup by mohl zlepšit proces zpracování dřeva a maximalizovat výtěžnost suroviny.

V první části práce je popsána implementace vlastní segmentační metody pro detekci trhlin v dřevě. Další část práce je věnována zpracování a anotací dat, včetně konverze formátů mezi různými anotačními nástroji a formáty používanými neuronovými sítěmi. Dále je prezentováno trénování dvou hlubokých neuronových sítí, YOLO a Mask R-CNN, pro identifikaci dalších defektů na CT snímcích dřeva. Během experimentů byla provedena analýza úspěšnosti těchto modelů, která ukázala mírně nižší přesnost detekcí. Tato zjištění poskytují cenné poznatky pro další optimalizaci tréninkových procesů a zlepšení výkonu modelů. Kromě toho byla provedena konverze výsledných metrik pro srovnání mezi různými experimenty a modely. Celkově práce přináší poznatky o úspěšnosti a omezeních současných metod a technik v oblasti detekce defektů pomocí analýzy CT snímků.

## Abstract

This Bachelor thesis focuses on the analysis of wood defects from CT images using image processing methods with the intention of developing an efficient method for automatic detection. This approach could improve the woodworking process and maximize the raw material yield.

The first part of the paper describes the implementation of a custom segmentation method for cracks detection in wood. The next part of the thesis is devoted to data processing and annotation, including format conversion between different annotation tools and formats used by neural networks. Furthermore, the training of two deep neural networks, YOLO and Mask R-CNN, to identify additional defects in CT images of wood is presented. During the experiments, the success rate of these models was analyzed and showed slightly lower detection accuracy. These findings provide valuable insights to further optimize the training processes and improve the performance of the models. In addition, a conversion of the resulting metrics was performed for comparison between different experiments and models. Overall, the work provides insights into the success and limitations of current methods and



techniques in defect detection using CT image analysis.

## **Klíčová slova**

CT snímky • segmentace • zpracování obrazu • defekty dřeva • anotace • YOLO • Detectron2 • Mask R-CNN



# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Metody detekce defektů dřeva</b>	<b>5</b>
2.1	Vady dřeva . . . . .	5
2.1.1	Trhliny . . . . .	5
2.1.2	Suky . . . . .	6
2.1.3	Cizí tělesa . . . . .	6
2.1.4	Dřevokazný hmyz . . . . .	6
2.2	Metody pro detekci vad dřeva . . . . .	6
2.2.1	Neinvazivní metody . . . . .	7
2.2.2	Invazivní metody . . . . .	8
2.3	Počítačové metody analýzy dřeva . . . . .	9
2.3.1	Metoda Houghovy transformace . . . . .	9
2.3.2	Metoda detekce suků . . . . .	10
<b>3</b>	<b>Zpracování CT snímků</b>	<b>11</b>
3.1	Principy CT a formát DICOM . . . . .	11
3.2	Normalizace obrázků . . . . .	12
<b>4</b>	<b>Metoda pro segmentaci trhlín</b>	<b>13</b>
4.1	Příprava CT snímků . . . . .	13
4.2	Binarizační metody prahování . . . . .	14
4.3	Morfologické transformace . . . . .	15
4.4	Detekce kontur . . . . .	15
<b>5</b>	<b>Dataset</b>	<b>17</b>
5.1	Anotace CT snímků . . . . .	17
5.2	Formát anotací . . . . .	18
5.3	Struktura datasetů . . . . .	19
5.3.1	Trénovací dataset . . . . .	19
5.3.2	Testovací dataset . . . . .	20

5.3.3	Augmentovaný dataset . . . . .	22
<b>6</b>	<b>Neuronové sítě</b>	<b>23</b>
6.1	Použité neuronové sítě . . . . .	23
6.1.1	Ultralytics - YOLO . . . . .	23
6.1.2	Detectron2 - Mask R-CNN . . . . .	24
6.1.3	Metriky . . . . .	25
6.2	Augmentace Dat . . . . .	28
<b>7</b>	<b>Experimenty a testování</b>	<b>31</b>
7.1	Vyhodnocení segmentační metody pro trhliny . . . . .	31
7.2	Výsledky neuronové sítě YOLO . . . . .	32
7.2.1	YOLO bez augmentace . . . . .	32
7.2.2	YOLO s vlastní augmentací . . . . .	33
7.2.3	YOLO s augmentací . . . . .	34
7.2.4	YOLO s kombinovanou augmentací . . . . .	36
7.2.5	Shrnutí YOLO modelů . . . . .	37
7.3	Výsledky neuronové sítě Mask R-CNN . . . . .	38
7.3.1	COCO metrika . . . . .	39
7.3.2	Mask R-CNN bez augmentace . . . . .	39
7.3.3	Mask R-CNN s vlastní augmentací . . . . .	40
7.3.4	Mask R-CNN s augmentací . . . . .	40
7.4	Převod YOLO metrik na COCO . . . . .	41
7.5	Srovnání přesnosti neuronových sítí . . . . .	42
7.6	Srovnání výsledků vlastní metody a neuronové sítě . . . . .	43
<b>8</b>	<b>Závěr</b>	<b>45</b>
	<b>Seznam obrázků</b>	<b>47</b>
	<b>Seznam tabulek</b>	<b>49</b>
	<b>Seznam zkratk</b>	<b>51</b>
	<b>Bibliografie</b>	<b>53</b>

Motivace pro tuto práci vychází z aktuální potřeby v oblasti dřevařského průmyslu, kde stávající metody měření a vyhodnocování nedosahují optimálních výsledků. S ohledem na rostoucí důraz na udržitelnost a efektivní využívání přírodních zdrojů, představuje optimalizace procesů vyhodnocování defektů v dřevě prostřednictvím pokročilých technologií klíčový krok směrem k udržitelnější a efektivnější dřevařské výrobě.

Dřevo je základním stavebním materiálem, který má široké uplatnění ve stavebnictví, nábytkářství, výrobě papíru a mnoha dalších odvětvích. Přesná analýza kvality dřeva je klíčovým faktorem pro zajištění bezpečnosti a spolehlivosti využití tohoto materiálu. Jedním z přístupů k tomuto problému je využití zobrazovacích technik, jako je výpočetní tomografie (CT), která umožňuje neinvazivní zkoumání vnitřní struktury dřeva. Nicméně, různorodé defekty obsažené v dřevě představují výzvu při přesné analýze CT snímků.

Cílem této bakalářské práce je provést analýzu a navrhnout metodu pro segmentaci trhlin ve dřevě pomocí tradičních metod pro zpracování obrazu a srovnání s přístupem používajícím neuronové sítě.

První část práce se zaměřuje na segmentaci trhlin dřeva, kde aplikuji různé metody segmentace obrazu na CT snímky dřeva. Cílem těchto metod je oddělit různé struktury a defekty dřeva od pozadí, což usnadní identifikaci potenciálních problémů. Segmentace tím současně přispěje k tvorbě anotací a přípravě dat pro neuronové sítě, což představuje významnou část práce.

Dalším krokem je využití již získaných dat k tvorbě datasetů k natrénování a validaci neuronových sítí.

V rámci této práce budu zkoumat a porovnávat účinnost těchto metod segmentace obrazu a neuronové sítě na různých typech defektů a různých dřevěných vzorcích. Výsledky této analýzy budou sloužit k vyhodnocení přesnosti a efektivity jednotlivých metod.

Tímto způsobem bude tato bakalářská práce přispívat k rozvoji technik pro analýzu defektů na CT snímcích dřeva a může mít potenciální aplikace ve dřevařském průmyslu, což může přispět k větší výtěžnosti suroviny.



# Metody detekce defektů dřeva

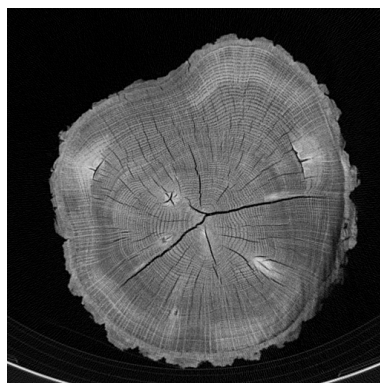
## 2

Bakalářská práce je zaměřena na identifikaci vad v dřevu pomocí CT. Dřevo je známé svou náchylností k různým vadám, v této práci se zaměřuji na detekci konkrétních typů vad, jako jsou trhliny, suky a různé defekty způsobené dřevokazným hmyzem. Tyto vady představují hlavní oblast zájmu při analýze CT dat. V následujících sekcích budou tyto vady podrobně rozebrány a bude poskytnut popis základních metod detekce vad dřeva, včetně počítačových metod.

## 2.1 Vady dřeva

### 2.1.1 Trhliny

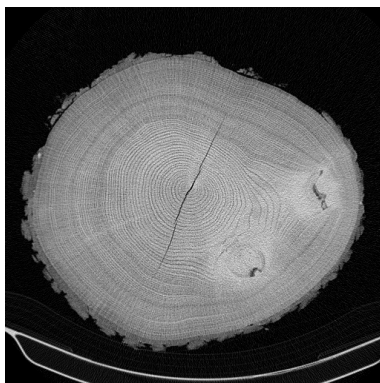
Trhliny jsou násilně od sebe oddělené vrstvy dřevních vláken. Jde o porušení celistvosti dřeva a tudíž o závažnou vadu, která výrazně snižuje pevnost, ohýbatelnost a další vlastnosti dřeva a tím i možnosti jeho využití [REJ14].



Obrázek 2.1: Příklad trhlín ve struktuře dřeva na CT snímku

### 2.1.2 Suky

Suky vznikají zarůstáním větví do kmene stromu. Představují jednu z nejčastějších vad dřevin. Větve jsou z hlediska životaschopnosti stromu nezbytným funkčním orgánem, avšak z pohledu výtěžnosti a následného použití dřeva má jejich růst negativní dopad. Suk totiž vrůstáním přerušuje normální průběh letokruhů. Tím je snížena pevnost a vzniká riziko rozvoje výsušných trhlin při změnách objemu[17].



Obrázek 2.2: Příklad suků a trhlin dřeva na CT snímku

### 2.1.3 Cizí tělesa

Cizími tělesy rozumíme přítomnost těles různého původu ve dřevě. Mohou to být kovové částice, plasty, kameny, hřebíky a další materiály, které mohou ovlivnit pevnost a kvalitu dřeva. Detekce cizích těles je důležitým aspektem, neboť umožňuje identifikovat potenciální rizika, která by mohla vést k degradaci dřeva a mohou mít negativní vliv na jeho využití.

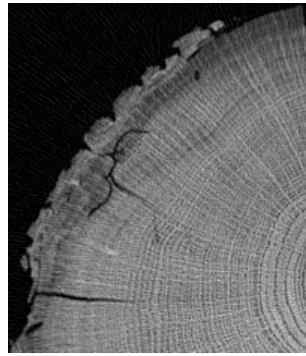
### 2.1.4 Dřevokazný hmyz

Defekty způsobené dřevokazným hmyzem jsou výsledkem činnosti škůdců, jako jsou tesařici, kůrovci a další, kteří napadají dřevo a mohou způsobit vážné poškození jeho struktury. Tyto defekty zahrnují otvory, tunely, díry a jiné anomálie, které jsou viditelné na CT snímcích viz. obrázek 2.3.

## 2.2 Metody pro detekci vad dřeva

Základní klasifikace metod pro detekci defektů v dřevinách vychází z rozdělení mezi invazivní a neinvazivní přístupy. Invazivní metody zahrnují fyzické manipulace s dřevinou, například prostřednictvím vrtání či řezání, což umožňuje získání přímého přístupu k vnitřní struktuře dřeviny. Naopak neinvazivní metody využívají





Obrázek 2.3: Detail na hmyzí stopy na dřevě

škálu technologií, včetně ultrazvuku, rentgenového záření, magnetické rezonance a dalších, s cílem získat informace o vnitřní struktuře dřeviny bez nutnosti fyzického poškození dřeva.

## 2.2.1 Neinvazivní metody

Nedestruktivní metody pro zhodnocení stavu dřevěných konstrukcí patří mezi nejčastěji používané metody diagnostiky. Tyto metody nevyžadují fyzický zásah do stromu, což je zvláště výhodné, protože minimalizují riziko poškození nebo narušení jeho struktury. Tím umožňují získávat důležité informace o zdraví a stavu stromů bez ohrožení jejich životaschopnosti.

### 2.2.1.1 Ultrazvuk

Metoda ultrazvuku využívá vysokofrekvenční zvukové vlny, které jsou vysílány do dřeviny. Na základě odezvy se určuje stav dřeviny. Ultrazvukové vlny se šíří různou rychlostí v závislosti na materiálu, kterým procházejí. Pokud ultrazvuková vlna narazí na defekt v dřevině, jako je například dutina nebo trhlina, bude se šířit rychleji než v okolním zdravém dřevu. Na základě tohoto rozdílu v rychlosti šíření ultrazvukových vln lze určit, zda je dřevina zdravá nebo obsahuje defekty [97]. Tato metoda je velmi užitečná pro detekci defektů dřevin, jako jsou například dutiny, trhliny, hniloba a další.

### 2.2.1.2 Gamma záření

Metoda Gamma záření je neinvazivní technika pro odhalování defektů v materiálech, včetně dřeva. Využívá vysokoenergetické záření z radioaktivních zdrojů, jako jsou iridium-192 nebo kobalt-60, které proniká dřevem a odhaluje přítomnost vad, jako jsou dutiny, trhliny nebo zlomy.

Při použití této metody je radioaktivní zdroj umístěn v těžkém oloveném kontejneru, který má dálkově ovládanou clonu. Clona reguluje tok záření skrz stíněnou hadici na testovaný objekt. Množství použitého záření závisí na vlastnostech testovaného materiálu.

Metoda Gamma záření se aplikuje v průmyslu pro nedestruktivní testování velkých objektů, včetně dřeva. Může sloužit k posouzení defektů před prodejem nebo zpracováním dřeva. Také se využívá k testování stromů, identifikaci vad, které mohou představovat riziko pro lidi nebo majetek.

Vliv na výsledky mohou mít faktory, jako je hustota dřeva nebo tloušťka kůry, a proto se metoda většinou kombinuje s dalšími pro zvýšení přesnosti detekce defektů ve stromech [97].

### 2.2.2 Invazivní metody

#### 2.2.2.1 Odporové zaražení trnu

Do dřeva je vrtáno přístrojem, který umožňuje kontinuální vrtání do předem určené hloubky v závislosti na přístroji. Před samotným měřením se doporučuje provést důkladnou inspekci na místě a v závislosti na zjištěných podmínkách optimalizovat počet a rozmístění sondovaných prvků.

Během vrtání do zkoumaného materiálu detekuje přístroj množství energie nutné k udržení konstantní vrtané rychlosti. Propojení s přenosným počítačem je nezbytné pro zaznamenání měření. Výstupem je křivka, která reflektuje změny energie během vrtání a odpovídá různým hustotám dřeva. Zlomy křivky v horních a dolních vrcholech naznačují místa s defekty.

Na grafu jsou vzdálenosti od povrchu zobrazeny na vodorovné ose x, zatímco na svislé ose y je hustotní profil naměřený pomocí Resistographu. Tyto hodnoty jsou vyhlazeny váženým aritmetickým průměrem. Numerické hodnoty délek vrtané dráhy poskytují přesné informace o lokalizaci a velikosti defektů. K vyhodnocení hustotních profilů se používá poměr délky křivky a vrtané dráhy nebo plochy pod křivkou, což poskytuje přesné informace o defektech v materiálu [Ngu15].

#### 2.2.2.2 Vrtání a Řezání

Invazivní postup, při kterém se dřevo fyzicky řeže nebo vrtá, aby byla odhalena jeho vnitřní struktura. Tato metoda umožňuje přímý vizuální přístup k vrstvám dřeva a umožňuje identifikaci vad, jako jsou například hniloba, zlomy nebo nepravidelnosti.

Je důležité vzít v úvahu normy a standardy kvality, které určují vhodnost materiálu pro dané účely. Tyto normy definují kritéria pro hodnocení jakosti řeziva, zahrnující vizuální kontrolu, měření rozměrů a identifikaci výrobních vad. Dodr-

žování technologických postupů je klíčové pro zajištění přesnosti rozměrů, kvality povrchu a minimalizaci výrobních vad.

Výsledné řezivo je tříděno podle stanovených kritérií a kvalitativně označeno, přičemž materiál nesplňující požadavky je upraven nebo klasifikován jako průmyslový odpad.

## 2.3 Počítačové metody analýzy dřeva

V této sekci se zaměřím na využití počítačové analýzy a zobrazovacích technik, zejména CT snímků, při identifikaci různých charakteristik dřeva. Jelikož má každá charakteristika specifické vlastnosti, budu představovat metody, které jsou navrženy pro detekci konkrétních aspektů dřeva.

### 2.3.1 Metoda Houghovy transformace

Metoda Houghovy transformace (HT) pro kružnice se dá využít k lokalizaci dřene v průřezu dřeva. Dřeň je střed stromu, který při zpracování dřeva může způsobovat praskliny nebo zvlnění dřevěných desek. K identifikaci dřene pomocí Houghovy transformace začneme nahráním dat do paměti GPU k jejich paralelnímu zpracování. Každý pixel je analyzován a jsou vypočítány informace o gradientech, následně se aplikuje Sobelův filtr pro detekci hran. Poté se použije Cannyho detektor hran k získání binárního obrazu s hranami. Většina detekovaných hran jsou letokruhy kmene. Postupně generujeme kolmice ve směru růstu a hledáme průsečíky. Nakonec je dřeň identifikována jako průsečík s nejvyšší vahou. Tato metoda je přesná ale výpočetně náročná [Gaz+20].

Pro minimalizaci výpočetní náročnosti byla navržena metoda MIN, která má za cíl nahradit časté používání metody HT. Tato metoda identifikuje na obrázku oblasti s nejnižší hustotou na CT snímcích a následně porovnává nalezenou dřeň s předchozím snímkem. Pokud rozdíl mezi nimi nepřesahuje předem stanovenou mez, je tato lokalizace dřene zachována. V opačném případě se k detekci dřene na obrázku použije metoda HT. Tento přístup je zaměřen na optimalizaci výpočetního výkonu a detekci dřene přednostně provádí rychlou metodou (Metoda MIN).

V případě, že ve snímku existuje vysoké množství šumu a letokruhy kmene nejsou viditelné, může dojít k selhání obou metod HT i MIN. V takové situaci přichází na řadu alternativní metoda nazvaná Boundary Approximation (BA). Metoda BA postupně prochází každý pixel v obraze a pokud alespoň 3 z okolních 3x3 pixelů jsou prázdné (tj. představují pozadí), je pixel přidán na seznam hran, což představuje vnější hranici kmene. Souřadnice dřene jsou poté vypočteny jako bodový průměr souřadnic těchto pixelů [Gaz+20].

### 2.3.2 **Metoda detekce suků**

V rámci této studie [Lon+12] byl představen algoritmus navržený pro automatizovanou detekci a extrakci charakteristik suků z CT snímků dřeva. Navrhnutý algoritmus se skládá z několika klíčových kroků, zahrnujících segmentaci, prahování, 3D transformaci vzdálenosti, 3D spojování komponent. Tyto kroky umožňují efektivní identifikaci suků a získání relevantních informací z obrazových dat.

V první části algoritmu se provádí detekce dřene. Podobně jako jsem popisoval v sekci 2.3.1, je použita Houghova transformace k nalezení kolmých přímk na letokruhy stromu. Místo s největším průnikem těchto přímk je pak označeno jako dřev kmene. Dřev je důležitá k určení orientace růstu suků.

V dalším kroku algoritmu dochází k segmentaci suků: Byla použita prahovací metoda. Hodnota prahu byla vybrána na základě histogramu odstínů šedi, který byl vyhlazen pomocí lokální polynomické regrese. Tímto způsobem jsou suky odděleny od kmene.

Následuje analýza 3D spojených komponent, což je technika určená pro identifikaci spojených oblastí v obraze. Tento postup umožňuje identifikovat a oddělit jednotlivé suky od okolního materiálu na základě jejich vzájemného propojení a struktury, a získat tak jejich 3D reprezentaci. Pro zachování efektivity extrakce spojených komponent a současně úspory paměti je použita strategie zpracování 3D obrazu po vrstvách. Pouze aktuální a předchozí vrstvy jsou udržovány v paměti.

Po detekci suků jsou 3D komponenty použity k další analýze a měření jejich charakteristik, jako je sklon, průměr, délka a objem. Dále je zjišťováno, zda se suky protínají s dřeví stromu.

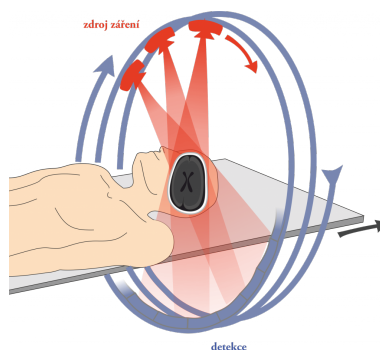
Tímto způsobem tato metoda dokázala získat 3D reprezentaci suků ve dřevě [Lon+12].

# Zpracování CT snímků

# 3

## 3.1 Principy CT a formát DICOM

CT (Výpočetní tomografie) je zobrazovací metoda, která umožňuje, za použití rentgenového záření, zobrazit celé tělo v sérii řezů. Výsledný obraz vzniká matematickou rekonstrukcí z řady rentgenových projekcí získaných postupně z různých úhlů. Získáme tak sadu tenkých vrstev, reprezentujících skenovaný objekt.



Obrázek 3.1: Princip CT snímku

CT snímky poskytují podrobnější informace, než běžné rentgenové snímky. U běžných rentgenových snímků je paprsek mířen na desku se snímaným objektem. V CT se rentgenový paprsek pohybuje v kruhu kolem snímaného objektu. Díky tomu máme mnoho různých pohledů na danou strukturu, což nám umožňuje získat mnohem větší množství detailů.

Pro zpracování CT snímků dřeva do použitelné podoby budu používat knihovnu DICOM, která umožňuje práci se stejnojmenným formátem. DICOM je zkratka pro Digital Imaging and Communications in Medicine a je hlavním nástrojem pro ukládání a přenos souborů lékařských zobrazovacích dat [Mc08]. Nicméně, jeho význam zahrnuje mnohem více než pouhý formát souborů či přenosový protokol.

DICOM funguje jako komplexní standard, který definuje jak formáty, tak i normy pro kvalitu a přenosové protokoly ve zdravotnickém prostředí. Tento stan-

dard poskytuje strukturovaný přístup k ukládání, sdílení a přenosu medicínských dat, což je zvláště klíčové v klinickém prostředí.

Struktura DICOMu zahrnuje přesné specifikace pro ukládání a organizaci dat. Každý DICOM soubor obsahuje metadata, která popisují nejen samotný obraz, ale také informace o pacientovi, zařízení, použitém protokolu a další relevantní údaje. Tato struktura umožňuje konzistentní interpretaci a výměnu dat mezi různými diagnostickými zařízeními a softwarovými platformami.

V souborové struktuře DICOMu se nachází hlavička sestávající z preambule a prefixu, kde preambule je 128 bajtová, pokud je implementovaná, může nabývat jakýchkoliv hodnot jinak nabývá hodnot 0x00H. Prefix je 4 bajtový identifikátor DICM, jehož účelem je usnadnit kompatibilitu s běžnými formáty souborů pro obrazová data.

Každý soubor DICOM obsahuje jednu datovou sadu. Datová sada představuje sadu informací o jednom konkrétním objektu nebo instanci, nazývanou Service-Object Pair (SOP). Tato datová sada obsahuje veškeré informace o daném záznamu, včetně lékařských obrazů, a je strukturována pomocí datových elementů.

## 3.2 Normalizace obrázků

Při zpracování obrázků z DICOM formátu byla provedena normalizace s cílem zjednodušit jejich zpracování a přípravu pro použití v neuronových sítích. DICOM soubory jsou často uloženy s hodnotami šedé škály v rozsahu od 0 do 4095, což může být nepraktické pro některé algoritmy zpracování obrázků. Proto jsem se rozhodnul normalizovat intenzity pixelů na rozsah od 0 do 255.

Tato normalizace způsobila ztrátu některých informací o obrázku a snížení jeho kvality. Například mohou být ztraceny detaily v extrémních hodnotách intenzity, což může ovlivnit schopnost algoritmů identifikovat jemné struktury nebo anomálie.

Přestože dochází k určité ztrátě informací, normalizace obrázků má několik výhod. Normalizované obrázky mají menší velikost, což usnadňuje manipulaci s nimi a ukládání. Celkově lze říci, že normalizace obrázků na rozsah od 0 do 255 byla nezbytná pro účinné využití v neuronových sítích, i když s sebou nesla určité kompromisy v kvalitě a detailnosti obrázků.

# Metoda pro segmentaci trhlin

## 4

Segmentace obrazu je klíčovým prvkem v oblasti zpracování obrazu, který má za cíl rozdělit obrázek na jednotlivé oblasti nebo objekty. Tento proces hraje důležitou roli v různých aplikacích, včetně analýzy lékařských snímků, rozpoznávání objektů v počítačovém vidění a mnoha dalších. V rámci této bakalářské práce byla provedena segmentace obrazu za použití knihovny OpenCV, což je open-source knihovna pro zpracování obrazu.

OpenCV poskytuje širokou škálu nástrojů a funkcí pro zpracování a analýzu obrazu, včetně filtrů, binarizačních metod a metod pro práci s konturami. Tato knihovna byla klíčovým nástrojem pro implementaci segmentačního procesu v rámci této práce, což umožnilo přesnou identifikaci objektů a vad v obrazových datech.

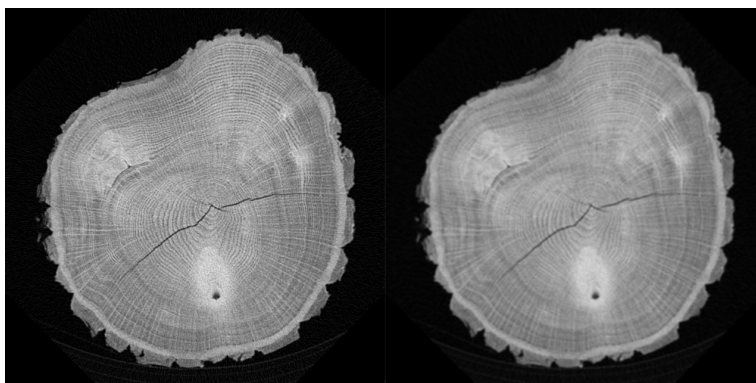
V následujících částech této práce budou detaily segmentačního procesu a použití knihovny OpenCV pro dosažení stanovených cílů důkladněji rozebrány.

## 4.1 Příprava CT snímků

Při práci s CT je nezbytné si uvědomit, že snímky, které zpracovávám, nejsou vždy dokonale čisté. Každý CT snímek obsahuje určité množství šumu, který může výrazně ovlivnit kvalitu a přesnost analýzy. Šum může být způsoben různými faktory, jako je radiační expozice, elektronický přenos signálu nebo třeba složitost snímkané struktury či jiné okolnosti při zpracování obrazu.

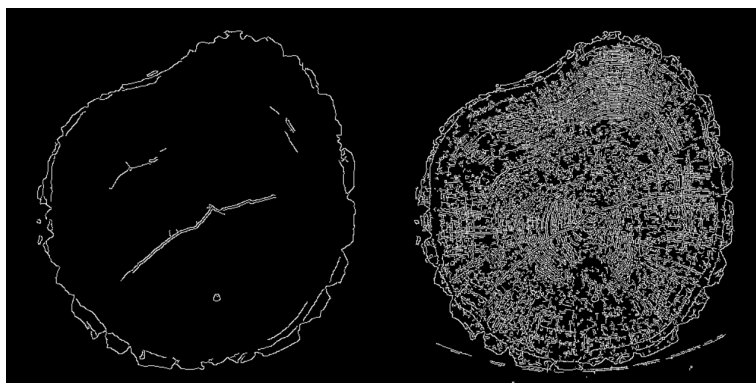
Odstranění šumu z CT snímků je klíčovým krokem, který mi umožňuje získat jasnější a kvalitnější obrazy. Při zpracování CT snímků dřeva se setkávám s komplexní strukturou dřeva, včetně letokruhů a dalších detailů, které mohou být rušivé při analýze. Proto je nezbytné provést redukci šumu v těchto snímcích, což umožňuje dosáhnout lepší kvality obrazu.

Jednou z běžných metod odstranění šumu je použití Gaussova filtru viz. obrázek 4.1, který mírně rozostří obraz a zjemní detaily. Tímto způsobem se sníží potřebný kontrast pro aplikaci metod, jako je tresholding nebo Cannyho detekce hran, což



Obrázek 4.1: Porovnání původního obrázku a obrázku po aplikaci Gaussova filtru

umožňuje lépe identifikovat a extrahovat různé defekty v dřevě, včetně suků, děr a dalších defektů.



Obrázek 4.2: Porovnání detekce hran s a bez použití Gaussova filtrování

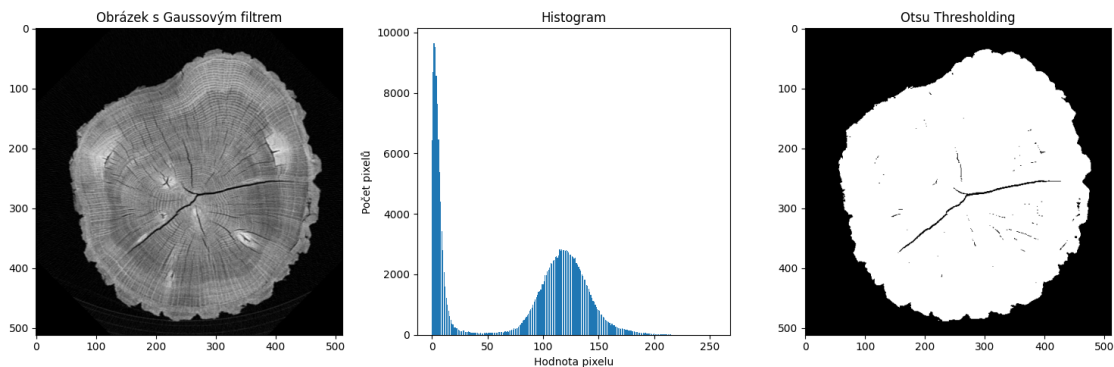
Z obrázku 4.2 je poznat, že pokud nepoužijeme žádnou metodu pro odstranění šumu tak je obraz nepřehledný. Při použití Gaussova filtru jsou nejdůležitější charakteristiky snímku dřeva přehledně vidět.

## 4.2 Binarizační metody prahování

Prahování je jednou ze základních metod segmentace obrazu. Tato technika spočívá v tom, že jsou vybrány pouze ty pixely obrazu, jejichž intenzita přesahuje určitý práh. Tímto způsobem je možné rozlišit objekty od pozadí na základě rozdílné intenzity, kterou mají vůči zvolenému prahu. V práci jsem využíval Otsuovu metodu prahování, což je globální přístup k volbě optimálního prahu. Otsuova metoda automaticky určuje vhodný práh pro prahování na základě intenzitního rozložení v obraze. To umožňuje dosáhnout ideálního prahu, který je přizpůsoben různým



typům defektů s různými intenzitami, a to bez nutnosti ručního určování prahu [GW08].



Obrázek 4.3: Gaussův filtr, histogram a Otsuovo prahování

Na obrázku 4.3 je zobrazen postup zpracování snímku. V levé části vidíme vstupní obrázek, který byl upraven Gaussovým filtrem pro odstranění šumu. V prostřední části je zobrazen histogram intenzit pixelů, který poskytuje informace o distribuci intenzit v obrázku. V pravé části je výsledek Otsuova prahování.

## 4.3 Morfologické transformace

Morfologické operace jsou jednoduché operace založené na tvaru obrazu, a většinou se využívají při zpracování binárních obrázků. Pro provedení těchto operací jsou potřebné dva vstupy: vstupní obraz a jádro. Dvě základní operace jsou eroze a dilace, ale existují také další varianty, jako je otevření, uzavření, Gradient a další. V mém konkrétním případě jsem využil erozi k rozšíření úzkých trhlin s cílem zlepšit detekci pomocí kontur. Současně jsem použil operaci uzavření k odstranění šumu v trhlínách.

## 4.4 Detekce kontur

Kontury, v kontextu zpracování obrazu, mohou být jednoduše popsány jako křivky spojující všechny sousedící body na hranici objektu, které mají stejnou barvu nebo intenzitu. Tento koncept je klíčovým nástrojem pro analýzu tvarů a detekci objektů na obrazech. Kontury umožňují extrahovat důležité informace o tvaru a struktuře objektů na obrazech, což je důležité pro řadu aplikací, včetně detekce trhlin a deformací v materiálech.

V této práci jsem využil metody pro detekci kontur z knihovny OpenCV k ohraničení trhlin a jiných defektů na obrázcích dřeva. Tento postup byl zásadní pro

vytvoření tréninkového datasetu pro následnou detekci trhlin a deformací na CT snímcích dřeva. Po aplikaci operací, které jsem rozebíral v sekci 4.2 a 4.3, jsem na obraze našel kontury pomocí funkce `cv2.findContours`.

Pro zdokonalení detekce kontur jsem implementoval filtrování podle velikosti (obsahu plochy). Tímto způsobem jsem zajistil, že nejsou detekovány zbytečné kontury o malých rozměrech, které nejsou relevantní pro analýzu. Zároveň jsem umožnil uživateli filtrovat i větší kontury, pokud má zájem pouze o malé defekty. Pro zjednodušení dalšího zpracování jsem následně prováděl aproximaci polygonů, které obklopují detekované defekty. Tímto krokem jsem snížil počet vrcholů polygonů, což umožnilo snadnější zpracování pro anotace.

Tyto kroky přispěly k vytvoření datasetu, který byl následně využit k tréninku detekčního modelu, schopného automaticky identifikovat trhliny, suky a hmyz.

V rámci této práce jsem vytvořil vlastní program pro generování anotací a tvorbu datasetů pro účely detekce a segmentace obrazu popsany v kapitole 4. Tento program mi umožnil automatizovat proces vytváření anotací a datasetů pro další zpracování.

Jedním z klíčových nástrojů, který jsem používal, byl anotační tool Supervisely<sup>1</sup>. Po vytvoření anotací v mém vlastním programu jsem je manuálně klasifikoval v rámci tohoto nástroje a nahrál je do formátu, který je vhodný pro použití s algoritmy detekce objektů, jako je YOLO nebo Mask R-CNN.

Mé programy pro segmentaci obrazu byly schopny generovat jednoduchý formát pro YOLO pro segmentaci a také formát pro Supervisely. Díky těmto nástrojům jsem mohl efektivně převádět anotace do požadovaných formátů a připravit je pro trénink mého modelu.

Dalším užitečným nástrojem byl COCO-merger<sup>2</sup> a spliter<sup>3</sup>, který jsem použil pro tvorbu validačních datasetů z hlavního souboru dat. Tento nástroj mi umožnil rozdělit dataset na tréninkovou a validační sadu s ohledem na optimální rozložení dat pro trénink a validaci modelu.

Při potřebě rozšíření datasetu jsem využil funkci mergeru pro sloučení mého stávajícího datasetu s nově vytvořenými augmentovanými daty. Tento postup mi umožnil vytvořit robustnější dataset pro trénink mého modelu s větší variabilitou v datech.

## 5.1 Anotace CT snímků

Anotace v datasetu jsou prováděny pomocí nástroje Supervisely, jak jsem již zmínil na začátku kapitoly 5. Pro anotace používám dataset, který byl vygenerován pomocí mé segmentační metody, jak bylo popsáno v kapitole 4. Po nahrání datasetu do nástroje Supervisely ručně přiřazuji detekované polygony do tříd, vynechané defekty označím polygonem a klasifikuji. Tímto postupem jsou vytvořeny anotace pro

<sup>1</sup><https://supervisely.com/>

<sup>2</sup><https://pypi.org/project/COCO-merger/>

<sup>3</sup><https://github.com/akarazniewicz/cocosplit>

každý obrázek z datasetu, přičemž všechny anotace jsou reprezentovány polygony, které ohraničují defekty.

## 5.2 Formát anotací

K tvorbě anotací jsem používal online platformu Supervisely, což je framework pro počítačové vidění a práci s anotacemi dat. K exportu anotací z mých programů do Supervisely jsem používal jejich stejnojmenný formát Supervisely.

Po ručním vytvoření anotací jsem převedl formát na YOLO, který používá jedna z neuronových sítí. Supervisely využívaly JSON soubory pro strukturu anotací, zatímco YOLO používá jednoduchou strukturu, kde každý záznam pro ukládání polygonů má následující formát: **class\_id x1 y1 x2 y2 ... xn yn** kde hodnoty x a y jsou souřadnice polygonu, který je vykreslen nad defektem, takto jsou zaznamenány všechny defekty a uloženy jako soubor .txt se stejným názvem jako popisovaný obrázek.

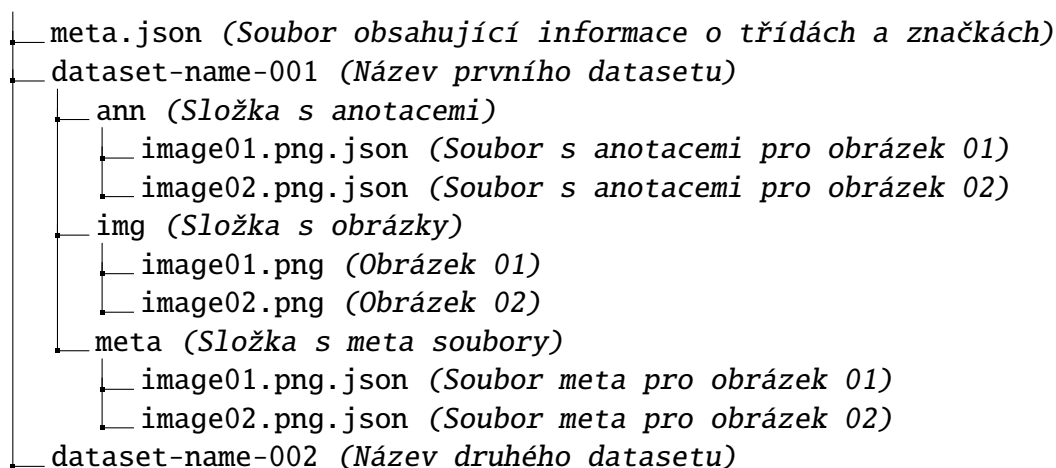
Dalším formátem, který jsem používal je formát COCO nebo-li Common Objects in Context, který veškeré informace o datasetu včetně anotací uchovává v jedné JSON struktuře. Struktura JSON souboru obsahuje 5 objektů:

- **info:** Obsahuje obecné informace o datasetu.
- **licenses:** Obsahuje informace o licencích, pokud jsou potřeba.
- **categories:** Obsahuje výčet všech tříd, které jsou v datasetu použity. Každá třída má své ID, název a případně i nadřazenou kategorii.
- **images:** Obsahuje informace o jednotlivých obrázcích v datasetu. Každý obrázek má přidělené ID a obsahuje informace o cestě, velikosti a dalších atributů.
- **annotations:** Obsahuje anotace objektů na jednotlivých obrázcích. Každá anotace odkazuje na ID obrázku a ID třídy.

V COCO formátu jsou třídy indexovány od jedné, přičemž nula je rezervována pro identifikaci pozadí. Anotace jsou popsány polygonem, který může být reprezentován buď kompresí pomocí RLE (Run-Length Encoding) nebo souřadnicemi. Kromě toho obsahuje informace o rozloze oblasti, bounding boxu.

Supervisely struktura může obsahovat více datasetů. Soubor meta.json obsahuje informace o všech třídách a značkách, které mohou být použity k anotacím v datasetech. Každá třída uvedená v souboru meta.json obsahuje jedinečné ID, tvar, barvu a případně značku. Značky neboli tagy můžou obsahovat dodatečné informace o třídách nebo scénách ve kterých se nachází.

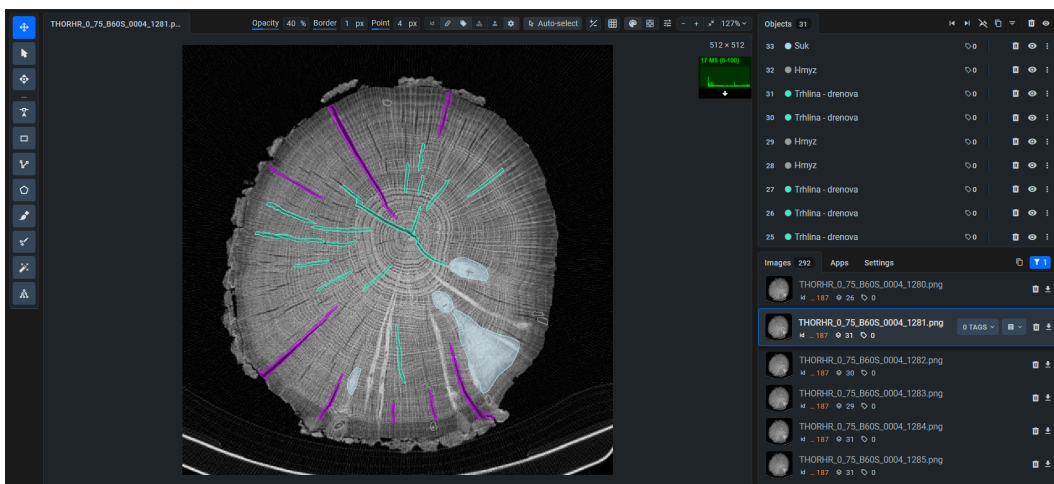
project-name (*Název projektu*)



Tento formát umožňuje širokou škálu podporovaných tvarů, jako jsou body, obdélníky, polygony, bitmapy, lomené čáry, kuboidy, 3D masky a další.

## 5.3 Struktura datasetů

Všechny datasety, které jsem použil, byly vytvořeny kombinací výsledů mých skriptů a následně ručně anotovány v nástroji Supervisely viz. obrázek 5.1. Anotace byly provedeny pomocí polygonů, což přineslo přesnost a detailnost, ale také vyžadovalo časově náročnou práci. Použití polygonů umožnilo využití segmentačních technik v neuronových sítích.



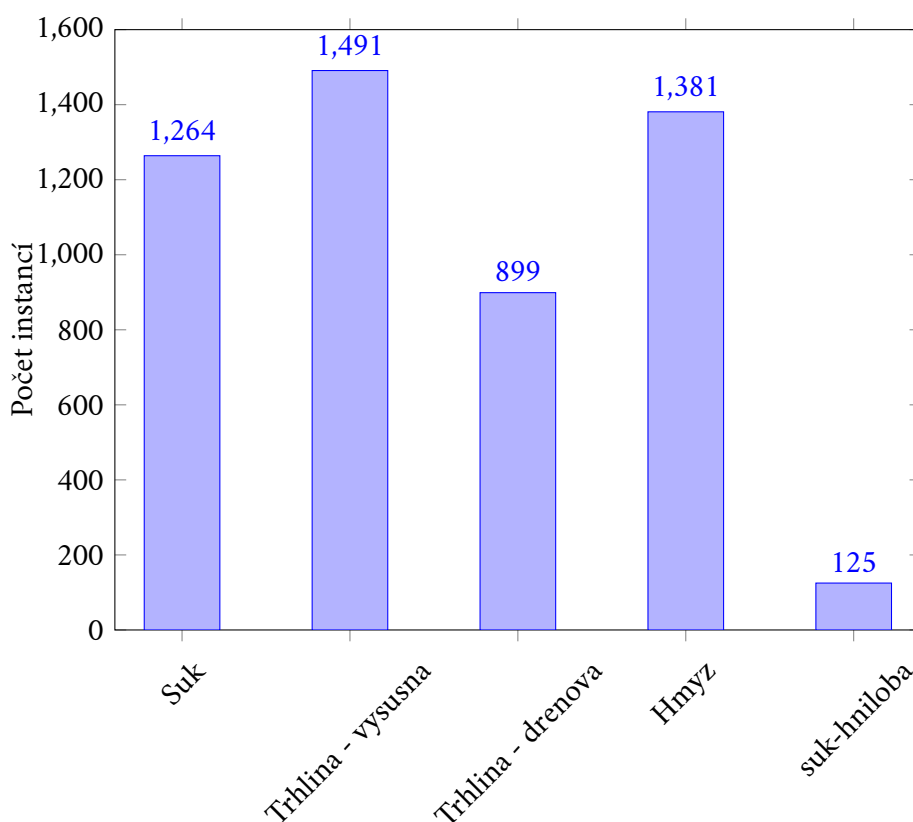
Obrázek 5.1: Ukázka tvorby anotací z prostředí Supervisely

### 5.3.1 Trénovací dataset

Základní trénovací dataset obsahuje 614 obrázků pocházejících ze 4 různých kmenů. Původní struktura datasetu zahrnovala 15 tříd, avšak kvůli nedostatku výskytu ně-

kterých tříd v datech a obtížnosti rozlišení některých defektů pro správnou anotaci byl počet tříd redukován na 5. Tento krok byl nezbytný pro zajištění kvalitních anotací a přesného tréninku detekčního modelu.

Kromě toho byl trénovací dataset následně rozdělen do validační části a části pro trénink. Validace dat je klíčovým prvkem při vývoji detekčního modelu, neboť umožňuje monitorovat vývoj jeho výkonu a zabránit přetrénování. Rozdělení bylo provedeno tak, aby validační část obsahovala přibližně 10% celkového počtu obrázků. Následující graf 5.2 ukazuje rozložení defektů napříč datasetem.

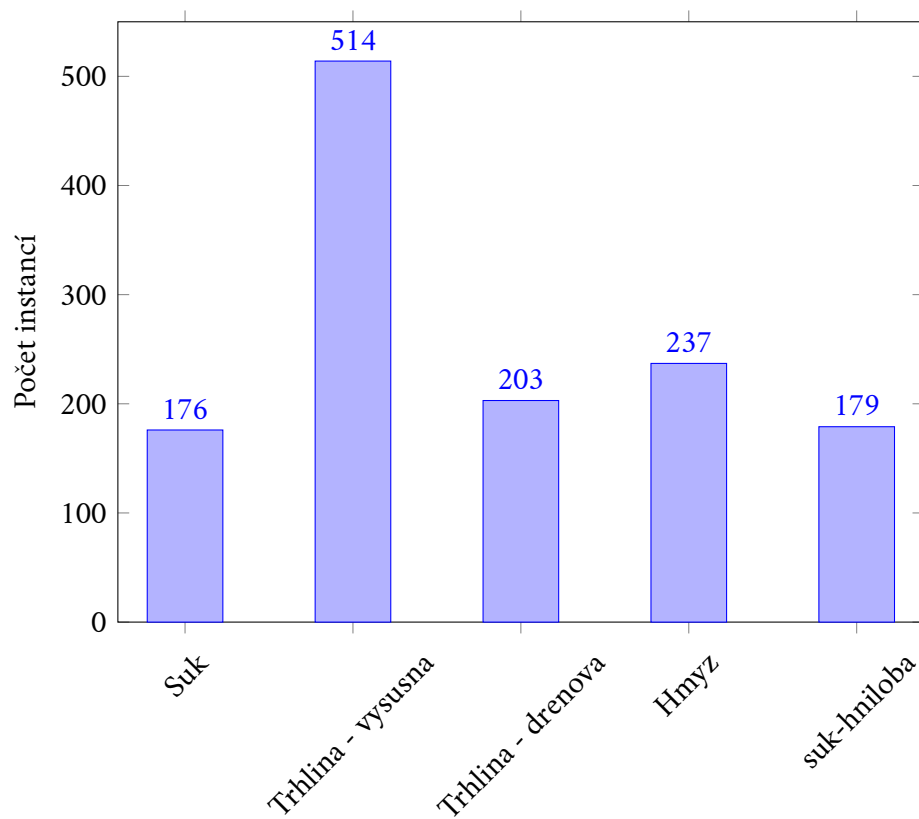


Obrázek 5.2: Četnost výskytu tříd v trénovacím datasetu

### 5.3.2 Testovací dataset

Testovací dataset pro validaci výkonu neuronových sítí byl sestaven ze 104 obrázků, které nebyly součástí trénovacího datasetu a jsou z různých kmenů. Při výběru obrázků jsem se zaměřil na co největší četnost defektů, což vedlo k zařazení obrázků, které obsahují kmeny s vysokým stupněm popraskanosti, čímž byla zvýšena četnost trhlin. Nepravidelnosti v anotacích defektů suk-hniloby a suku jsou důsledkem přítomnosti dřeva s velkými suky, které mohou obsahovat několik menších loži-

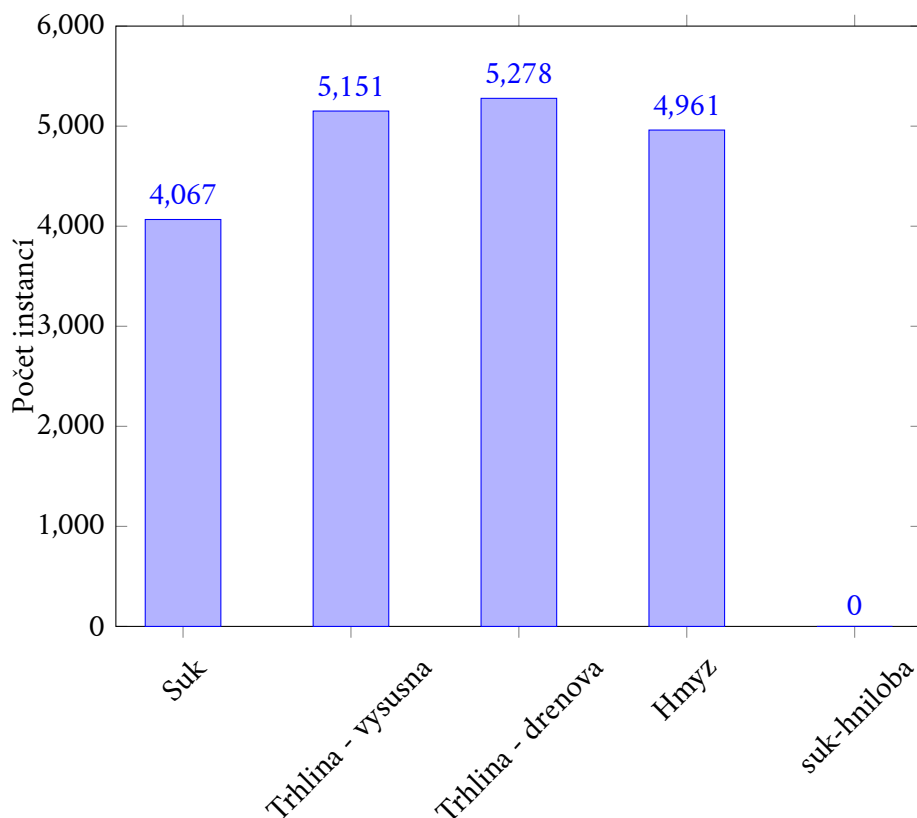
sek hniloby, jež jsou anotovány samostatně. Graf 5.3 znázorňuje rozložení instancí jednotlivých tříd v datasetu.



Obrázek 5.3: Četnost výskytu tříd v testovacím datasetu

### 5.3.3 Augmentovaný dataset

Můj augmentovaný dataset o velikosti 3000 obrázků je sestaven z výřezů defektů z jiných obrázků, které jsou následně náhodně vybírány a vkládány tak, aby nedocházelo k jejich kolizi. Třída suk-hniloba byla zcela vynechána z datasetu, jelikož jeho přítomnost závisí na konkrétním suku a nemohu ho vkládat zcela náhodně, jako je tomu u ostatních defektů viz. graf 5.4.



Obrázek 5.4: Četnost výskytu tříd v augmentovaném datasetu



# Neuronové sítě

## 6

Neuronové sítě představují klíčový nástroj v oblasti zpracování obrazu a jejich využití v analýze obrazu nabízí mnoho výhod pro specifický úkol této práce, detekci a segmentaci defektů dřeva. Díky své schopnosti automaticky extrahovat vzory a charakteristiky z obrazových dat, jsou schopny přizpůsobit se různým formám defektů, což je klíčové pro úspěšnou lokalizaci vad.

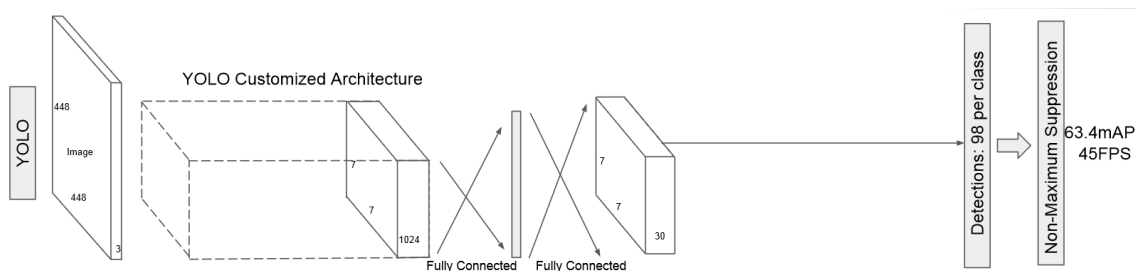
Jednou z výhod neuronových sítí je schopnost generalizace, která umožňuje aplikovat jejich znalosti na nová, dosud neviděná data. Tím se zajišťuje použitelnost modelu na různé typy defektů a nové snímky dřeva. Z pohledu práce jsou neuronové sítě schopny významně snížit potřebu manuální anotace a klasifikace defektů. Tím pádem umožňují rychlý a efektivní proces analýzy, což je klíčové v situacích, kdy pracujeme s velkým množstvím obrazových dat. Obecně lze říci, že v rámci analýzy defektů dřeva přináší neuronové sítě vysokou efektivitu, přesnost a schopnost zpracovávat složitá obrazová data.

## 6.1 Použité neuronové sítě

### 6.1.1 Ultralytics - YOLO

YOLO (You Only Look Once) je algoritmus detekce objektů v reálném čase vyvinutý Josephem Redmonem a Ali Farhadim v roce 2015 [Liu+16]. Tento jednorůchodový detektor objektů používá konvoluční neuronové sítě (CNN) k predikci bounding boxů a pravděpodobností tříd objektů na vstupních obrazech. YOLO rozděljuje vstupní obrázek do mřížky buněk a pro každou buňku předpovídá pravděpodobnost přítomnosti objektu a souřadnice bounding boxu objektu. Tento přístup umožňuje zpracovat celý obrázek najednou, což znamená, že dokáže detekovat objekty mnohem rychleji než metody používající regionální CNN. Navíc YOLO dosahuje vyšší efektivity, protože nepotřebuje více stupňů nebo navržených regionů, aby detekoval objekty.

Jednou z hlavních výhod YOLO je jeho schopnost provádět detekci objektů v reálném čase s vysokou rychlostí a přesností. Na rozdíl od metod jako R-CNN, které



Obrázek 6.1: Struktura jednorůchodového modelu YOLO

vyžadují několik fází a návrhu regionů, YOLO zpracovává celý obrázek najednou, což výrazně zrychluje detekci objektů. Navíc, YOLO se může chlubit širokým spektrem aplikací, včetně bezpečnostních systémů a automatického řízení vozidel.

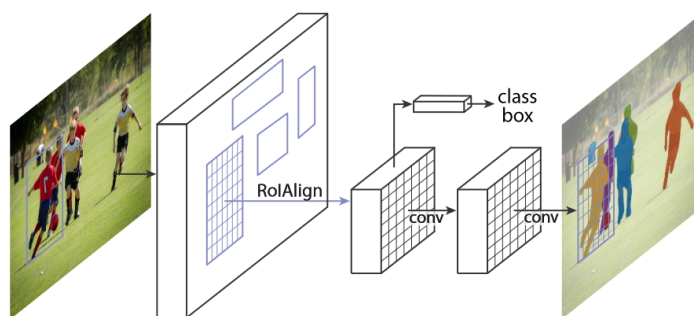
Nicméně, YOLO není bez nevýhod. Jednou z nich je menší přesnost ve srovnání s metodami, které používají více stupňů, jako je R-CNN. Tato nižší přesnost může být zvláště patrná při detekci menších objektů nebo v obtížných podmínkách. Další nevýhodou je sklon YOLO k nepřesnému ohraničení objektů, což může vést k chybám v lokalizaci [Liu+16].

Pro potřeby této práce jsem se rozhodl využít YOLOv8, což je nejnovější verze této architektury. YOLOv8 přináší několik vylepšení a nových funkcionalit, a především, je schopný poskytovat možnosti segmentace obrazu. Tato schopnost umožňuje precizní oddělení objektů na obraze a poskytuje tak důležité informace o kontextu a struktuře scény. Celkově tedy YOLOv8 nabízí široké spektrum možností, zejména v oblasti segmentace obrazu, což je pro mé účely klíčové.

## 6.1.2 Detectron2 - Mask R-CNN

Detekční model Mask R-CNN od Detectron2, vyvinutý jako rozšíření Faster R-CNN, reprezentuje významný pokrok v oblasti segmentace objektů v obrazech. Jednou z klíčových inovací této architektury je využití regionální konvoluční sítě (R-CNN), která umožňuje identifikaci potenciálních regionů zájmu obsahujících objekty. Oproti jednofázovému přístupu, jaký používá YOLO, R-CNN provádí detekci objektů dvoufázovým procesem. Nejprve se pomocí algoritmu selektivního vyhledávání identifikují různé oblasti v obraze, které by mohly obsahovat objekty. Tyto oblasti jsou pak podrobně analyzovány konvoluční neuronovou sítí, aby se určilo, zda obsahují objekty, a pokud ano, jakého typu, paralelně s touto operací se zjišťují i binární masky detekovaných objektů viz. obrázek 6.2. Tento přístup umožňuje Mask R-CNN dosáhnout větší přesnosti a detailnosti v detekci objektů, což je zvláště důležité ve složitých scénách s menšími nebo překrývajícími se objekty [He+18].

Ve srovnání s jinými metodami, jako je YOLO, je Mask R-CNN pomalejší, ale



Obrázek 6.2: Mask R-CNN framework pro segmentaci objektů.

jeho síla tkví v kvalitě segmentace. Tato síť vytváří detailní masky pro jednotlivé instance objektů v obraze, což umožňuje precizní a detailní segmentaci. To je zvláště významné v aplikacích, kde je klíčová vysoká přesnost a detailnost segmentace.

## 6.1.3 Metriky

Metrika ve strojovém učení slouží jako nástroj pro objektivní vyhodnocení výkonu algoritmů. Při posuzování úspěšnosti modelů je klíčové zohlednit nejenom jednu, ale i několik metrik, neboť různé metriky odhalují různé aspekty výkonu modelu. V této sekci popíšu několik klíčových metrik pro vyhodnocování algoritmů nebo modelů strojového učení. Každá z těchto metrik poskytuje určitý pohled na výkon modelu a může být vhodná v různých kontextech a aplikacích.

### 6.1.3.1 Intersection over Union

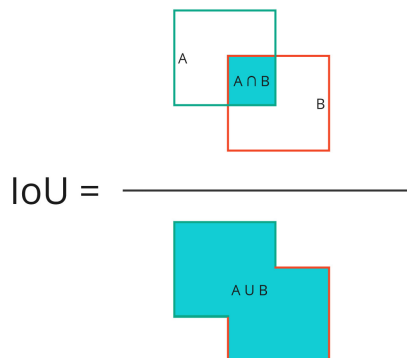
IoU (Jaccardův index), známý také jako Intersection over Union, je metrika používaná zejména v oblasti počítačového vidění a strojového učení k vyhodnocování přesnosti detekce objektů. Tato metrika je klíčová při tréninku a testování algoritmů detekce objektů, neboť umožňuje kvantifikaci míry shody mezi predikovanými a skutečnými oblastmi objektů [Rez+19].

Princip IoU spočívá v porovnání dvou oblastí: predikované oblasti a skutečné (anotované) oblasti. Predikovaná oblast je oblast, kterou algoritmus identifikoval jako obsahující objekt, zatímco skutečná oblast je oblast označená pomocí anotací. IoU je poté vypočteno jako poměr plochy překryvu mezi predikovanou a skutečnou oblastí k ploše sjednocení těchto dvou oblastí.

Matematicky je toto vyjádřeno vzorcem:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{|I|}{|U|}$$

Kde  $A$  a  $B$  představují predikovanou a skutečnou oblast,  $|A \cap B|$  je velikost průniku těchto oblastí,  $|A \cup B|$  je velikost jejich sjednocení a  $|I|$  a  $|U|$  jsou velikosti průniku a sjednocení.



Obrázek 6.3: Vizualizace překryvu u IoU

IoU se pohybuje mezi 0 a 1, kde 0 znamená žádný překryv a 1 znamená úplné překrytí mezi predikovanou a skutečnou oblastí. Vyšší hodnota IoU naznačuje lepší shodu mezi predikovanou a skutečnou oblastí a tím i vyšší přesnost detekce. Nižší hodnota IoU naopak znamená menší shodu a nižší přesnost detekce. Ukázka výpočtu překrytí dvou objektů viz. obrázek 6.3.

IoU je jednou z klíčových metrik používaných pro vyhodnocování výkonu algoritmů detekce objektů a je běžně využívána při vývoji a testování nových metod a algoritmů v oblasti počítačového vidění a strojového učení. V oblasti segmentace, kde se pracuje s maskami predikovaných objektů, princip metriky IoU zůstává stejný, ale místo porovnávání bounding boxů se porovnávají masky (segmenty) predikovaných objektů.

### 6.1.3.2 Accuracy

Přesnost (*Accuracy*) je jednou z klíčových metrik používaných k hodnocení výkonnosti klasifikačních modelů. Tato metrika měří celkovou míru správných predikcí modelu a je definována jako podíl počtu správně predikovaných pozitivních ( $TP$  - *True Positives*) a negativních ( $TN$  - *True Negatives*) událostí k celkovému počtu událostí v datasetu. Vzorec pro výpočet přesnosti je:

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP}$$

Kde:

- $TN$  (*True Negatives*) je počet skutečně negativních případů, které byly správně klasifikovány jako negativní,

- *TP (True Positives)* je počet skutečně pozitivních případů, které byly správně klasifikovány jako pozitivní,
- *FN (False Negatives)* je počet skutečně pozitivních případů, které byly nesprávně klasifikovány jako negativní,
- *FP (False Positives)* je počet skutečně negativních případů, které byly nesprávně klasifikovány jako pozitivní.

Tato metrika poskytuje ucelený pohled na schopnost modelu správně klasifikovat instance obou tříd a je často používána jako základní ukazatel úspěšnosti modelu. Na tuto metriku navazují další důležité metriky jako *precision* a *recall*, které poskytují hlubší náhled do výkonu modelu [Mis20].

### 6.1.3.3 Precision

Precision je metrika, která určuje kvalitu pozitivních predikcí tím, že měří jejich správnost. Je to počet správně predikovaných pozitivních výsledků dělený součtem správně predikovaných pozitivních a nesprávně predikovaných pozitivních výsledků. Vzorec pro výpočet přesnosti je:

$$Precision = \frac{TP}{TP + FP}$$

Přesnost lze chápat jako metriku kvality; vyšší přesnost implikuje, že algoritmus poskytuje relevantnější výsledky. Je zcela zaměřen na správnost pozitivních predikcí a nevěnuje pozornost správné detekci negativních predikcí.

### 6.1.3.4 Recall

Recall, také označovaný jako úplnost modelu, měří schopnost modelu správně detekovat pozitivní predikce. Jedná se o procento správně predikovaných pozitivních predikcí ze všech skutečně pozitivních predikcí. Pro výpočet recallu klasifikačního modelu platí vzorec

$$Recall = \frac{TP}{TP + FN}$$

Vysoké skóre recallu naznačuje, že klasifikátor správně předpovídá většinu relevantních výsledků. Nicméně, metrika recallu nebere v úvahu potenciální dopady falešných pozitiv, tj. predikcí, které jsou mylně identifikovány jako pozitivní [Mis20].

### 6.1.3.5 mAP a AP

Další z metrik, používaná k vyhodnocení výkonu detekčních algoritmů je průměrná přesnost (mAP) [Tan22]. Je to průměrná hodnota přesností (AP) pro všechny třídy

objektů v testovaném datasetu. Vzorec pro výpočet mAP je:

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_i$$

Kde  $n$  je počet tříd objektů a  $AP_i$  je průměrná přesnost (Average Precision) pro každou třídu  $i$ .

Průměrná přesnost (AP) je míra vyjadřující kvalitu detekce objektů v konkrétní třídě. AP měří plochu pod křivkou přesnosti (precision-recall curve), která ukazuje změnu přesnosti detekce s rostoucím počtem detekovaných objektů. Vzorec pro výpočet AP je:

$$AP = \int_0^1 p(r) dr$$

Kde  $p(r)$  je funkce precision-recall a  $r$  je recall, což je podíl skutečně detekovaných objektů na všech skutečných objektech v datasetu viz. sekce 6.1.3.4.

### 6.1.3.6 F1 Score

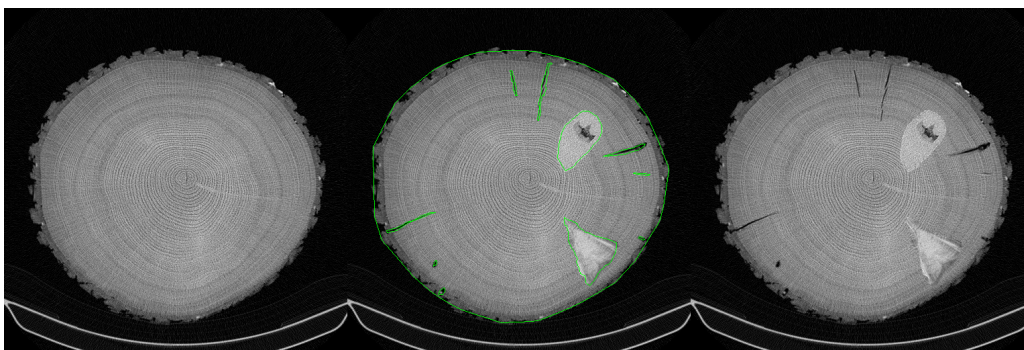
F1 Score je metrika, která kombinuje přesnost (Precision) a úplnost (Recall) do jednoho čísla a je často používána k hodnocení výkonu klasifikačních modelů. F1 Score je definováno jako harmonický průměr přesnosti a úplnosti a je vypočítáno pomocí následujícího vzorce:

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Kde Precision a Recall jsou metriky, které jsou definovány v sekcích 6.1.3.3 a 6.1.3.4. Vyšší hodnota F1 Score ukazuje na lepší vyvážení mezi přesností a úplností klasifikace. F1 Score je často používáno v oblastech, kde je důležité minimalizovat jak falešně pozitivní, tak falešně negativní klasifikace.

## 6.2 Augmentace Dat

Augmentace dat je technika rozšiřování dat, která má za úkol zlepšovat výkon modelů neuronových sítí. Technik pro tvorbu augmentací u strojového vidění může být mnoho jako je například rotace, rozostření, saturace nebo převrácení obrázku. Některé neuronové sítě mají automaticky zapnutou augmentaci dat, která se provádí v průběhu trénování, většinou se jedná o kombinaci dříve zmíněných technik. V tomto kontextu jsem vytvořil vlastní metodu augmentace specifickou pro potřeby této práce, místo automatické augmentace. Konkrétně se jedná o generování obrázků s náhodně rozloženými defekty, které byli vyřezány z jiných obrázků, tento druh augmentace se nazývá CutMix.



Obrázek 6.4: Proces augmentace od počátečního pozadí (vlevo) přes plánované polygony s umístěnými defekty (uprostřed) až po konečný augmentovaný obrázek (vpravo)

Nejprve vezmu anotovaný dataset a podle anotací na obrázku vyřezu všechny defekty, které uložím. Takto získám přesný výřez defektu i jeho ohraničující polygon. Mám dvě vstupní složky: jedna obsahuje vyřezané defekty, zatímco druhá obsahuje několik kmenů bez zjevných defektů, sloužící jako pozadí pro vkládání defektů. V programu následně vyberu náhodné pozadí ze složky a postupně vkládám defekty na pozadí. Defekty otáčím okolo středu obrázku náhodně mezi  $0^\circ$  a  $360^\circ$ . Tím zajistím potřebné množství různých obrázků. Samozřejmě řeším kolize defektů, žádný defekt se nemůže překrývat s jiným. To zajistím detekcí kolizí ohraničujících polygonů okolo defektů. Aby se předešlo výskytu defektů mimo kmen, vytvořil jsem konvexní obal okolo kůry kmene viz. obrázek 6.4. Tento polygon funguje jako prostor, kde lze umisťovat další defekty a zároveň slouží jako bariéra, která zabráni jejich výskytu mimo tuto specifikovanou oblast. Díky informacím o geometrické povaze každého z defektů, generuji ke každému augmentovanému obrázku, již klasifikované anotace ve formátu YOLO.





# Experimenty a testování

## 7

Pro provedení experimentů jsem připravil testovací dataset obsahující 104 obrázků z 4 různých oskenovaných kmenů. Každý obrázek v datasetu byl anotován s ohledem na objekty v obraze. Následně jsem použil tuto anotaci k vyhodnocení kvality detekovaných objektů generovaných různými metodami.

Každou z metod jsem nechal detekovat objekty a jejich segmentaci na stejných obrázcích z testovacího datasetu. Poté jsem pomocí již předem zmíněných metrik, vyhodnotil kvalitu těchto detekcí ve srovnání s anotacemi.

V této sekci popisuji postup experimentů, včetně použitých metod a vyhodnocených metrik, a analyzuji výsledky, kterých jsem dosáhl.

## 7.1 Vyhodnocení segmentační metody pro trhliny

K vyhodnocení mé vlastní segmentační metody, jsem využíval techniky zpracování obrazu k úpravě obrázků a následné segmentaci identifikovaných defektů dřeva. Tato metoda je navržena tak, aby lokalizovala defekty v dřevu a následně je exportovala jako polygony. Je třeba zdůraznit, že tato metoda není schopna klasifikovat jednotlivé objekty do určitých tříd, což v praxi znamená, že nedokáže přesně rozlišit specifické typy defektů, jako jsou trhliny nebo hmyz.

Přestože moje metoda nedokáže provádět klasifikaci objektů do tříd, umožňuje mi provést kvalitativní porovnání detekovaných defektů s anotovanými defekty pomocí metriky IoU (Intersect over union). Tato metrika měří míru překrytí dvou oblastí a je základním nástrojem pro porovnání přesnosti segmentace. Pro každou detekci jsem tedy analyzoval predikované instance s nastaveným prahem IoU na hodnotu 0.25 a 0.5, což znamená minimální stupeň překrytí objektů, který musí být dosažen pro to, aby byl detekovaný defekt považován za korektní. Výsledky této analýzy jsou shrnuty v tabulce 7.1 při výpočtu jsem v testovacím datasetu ignoroval třídy "Suky" a "suk-hniloba".

IoU	TP	FP	FN	Precision	Recall	F1 score
0.25	490	519	460	48.56	51.57	50.02
0.5	241	768	709	23.88	25.30	24.60

Tabulka 7.1: Výsledky segmentace trhlin s použitím různých prahů IoU (%)

## 7.2 Výsledky neuronové sítě YOLO

K tréninku jsem zvolil již předtrénovaný model od YOLO: YOLOv8s-seg, který byl pro mé účely dostatečně výkonný a dotrénoval jsem k němu data zmíněná v sekci 5.3.1. Trénink jsem prováděl na grafické kartě GeForce GTX 1060 s 6GB paměti a podporou CUDA verze 11.8.

Tréninkový proces byl proveden s použitím techniky early stopping. Trénink začal s 300 epochami a byla stanovena podmínka trpělivosti (patience) na 50 epoch. To znamená, že pokud se model nepodařilo zlepšit po 50 po sobě jdoucích epochách, tedy pokud hodnota loss funkce přestala klesat, byl trénink ukončen. Před tréninkem jsem dále provedl rozdělení datasetu, přičemž 70 obrázků bylo vyhrazeno pro validační sadu, zatímco zbylé obrázky byly použity pro trénink.

Po dokončení tréninku jsem provedl vyhodnocení natrénovaného modelu na stejném testovacím datasetu, který jsem použil při hodnocení předchozích segmentačních metod. Výsledky tohoto vyhodnocení posloužily k analýze účinnosti natrénovaného modelu a zhodnocení vlivu augmentace dat na jeho výkon.

### 7.2.1 YOLO bez augmentace

Model byl vyhodnocen pomocí několika metrik, které mi poskytují důležité informace o jeho schopnosti správně identifikovat objekty a segmentovat oblasti zájmu viz. sekce 6.1.3.

Třída	Precision	Recall	mAP50	mAP50-95
Všechny	47.4	28.7	30.7	13.4
suk-hniloba	89.8	3.86	28.7	8.02
Suk	46.4	35.6	32.6	14.5
Trhlina - vysusna	53.6	65.7	61.3	33.8
Trhlina - drenova	10.6	13.2	5.96	2.61
Hmyz	36.7	25.1	24.9	8.22

Tabulka 7.2: Výsledky modelu YOLO pro bounding boxy (%)

Tabulka 7.2 obsahuje metriky pro detekci bounding boxů. Pro všechny třídy jsem pozoroval relativně nízkou precision (průměrně 47.4%) a recall (průměrně 28.7%),

což poukazuje, že mám obtíže správně identifikovat všechny instance objektů v obraze.

Pokud se zaměřím na jednotlivé třídy, zjišťuji, že třída "suk-hniloba" má vysokou precision (89.8%) s nízkým recall (3.86%). To naznačuje, že správně identifikuji většinu suk-hniloby, ale mnoho skutečných instancí této třídy zůstává nezachyceno. Na druhou stranu, třída Trhlina - výsušná má vyváženou kombinaci precision (53.6%) a recall (65.7%), takže efektivně identifikuji většinu instancí této třídy. Naopak třída Trhlina - dřevná vykazuje nízkou precision (10.6%) a recall (13.2%), z testování vím že můj model často zaměňuje tuto třídu s trhlinou - výsušnou.

Třída	Precision	Recall	mAP50	mAP50-95
Všechny	32.1	31.2	28.1	10.1
suk-hniloba	22.9	1.31	14.1	4.44
Suk	47.9	43.6	37.5	12.1
Trhlina - vysusna	50.8	68.5	60.6	22.4
Trhlina - drenova	9.79	16.8	5.73	1.64
Hmyz	29.1	25.6	22.6	9.66

Tabulka 7.3: Výsledky modelu YOLO pro segmentaci (%)

Tabulka 7.3 obsahuje vyhodnocení na základě metrik pro segmentaci. Pro všechny třídy byla pozorována relativně nízká precision (průměrně 32.1%) a recall (průměrně 31.2%), což ukazuje, že mám obtíže s přesnou segmentací objektů v obraze. S ohledem na složitost a velikost jednotlivých defektů a s tím, že YOLO je primárně určen k detekci objektů, se s horšími výsledky u segmentace počítalo. Zbytek tabulky je ekvivalentní k tabulce s bounding boxy, pouze s nižší precision. To implikuje, že přesná segmentace objektů, zejména drobných a složitých defektů dřeva, je pro YOLO obtížná.

## 7.2.2 YOLO s vlastní augmentací

Augmentace dat je často využívaná technika pro zlepšení výkonu modelů detekce objektů a segmentace. Následují tabulky, které obsahují výsledky pro model natrénovaný pomocí již zmíněného datasetu v sekci 5.3.1 a k tomu 3000 augmentovaných obrázků vygenerovaných mým programem viz. sekce 6.2.

Z tabulky 7.4, která obsahuje metriky pro bounding boxy lze vidět, že augmentace dat zvýšila přesnost a úplnost detekce. Celkově dosahuje tento model lepších výsledků než model bez augmentace tedy (mAP50) je kolem 38.5% a s recallem (R) 35.7%, což znamená že identifikuje přibližně 35.7% skutečných instancí objektů ve scéně. Taky jde vidět výrazné zlepšení u tříd, kterých se augmentace přímo týkala (suk, trhlina - výsušná, trhlina - dřevná, hmyz). Nicméně nedostatek trénovacích dat pro třídu suk-hniloba, způsobuje, že model nedosahuje žádných detekcí této

Třída	Precision	Recall	mAP50	mAP50-95
Všechny	41.4	35.7	38.5	21.2
suk-hniloba	0	0	0	0
Suk	62.6	30.1	48.3	20.9
Trhlina - vysusna	68.3	97.2	88.3	58
Trhlina - drenova	31.4	25.6	26.6	15.6
Hmyz	44.6	25.8	29.2	11.7

Tabulka 7.4: Výsledky modelu YOLO pro bounding boxy s augmentací (%)

třídy. Kvůli tomu, že suk-hniloba přímo závisí na Suku tak jsem tuto třídu nezahrnul do augmentace což způsobilo, že byl dataset nevyvážený.

Třída	Precision	Recall	mAP50	mAP50-95
Všechny	42	37	39.8	16.6
suk-hniloba	0	0	0	0
Suk	72.3	35.3	56	21.7
Trhlina - vysusna	68.5	99.2	89.9	44.8
Trhlina - drenova	25.9	23.1	24.2	7.62
Hmyz	43.6	27.2	29	9

Tabulka 7.5: Výsledky modelu YOLO pro segmentaci s augmentací (%)

Z tabulky 7.5 plyne, že augmentace dat zlepšila celkový výkon modelu. Zejména u tříd Suk a Trhlina - vysusna dosahuje model výrazně lepších výsledků s precision přes 70% a recall nad 35%. To naznačuje, že augmentace dat efektivně pomáhá modelu identifikovat objekty ve scéně s vysokou přesností a úplností. Nicméně, i přes augmentaci, třídy s nízkým počtem instancí, jako suk-hniloba a Hmyz, stále dosahují nižší přesnosti a úplnosti, což může být důsledkem nedostatečného množství trénovacích dat.

### 7.2.3 YOLO s augmentací

V této sekci prezentuji výsledky experimentu s použitím YOLO vestavěné augmentace. K augmentaci dat jsem použil kombinaci následujících parametrů:

- *Shear = 180* Parametr určuje zkosení obrázku podél osy x nebo y maximálně o zadanou hodnotu.
- *Scale = 0.5* Parametr určuje zmenšení nebo zvětšení obrázku o zadanou hodnotu.
- *Flipud = 0.5* Pravděpodobnost vertikálního převrácení obrázku.

- *Fliplr* = 0.5 Pravděpodobnost horizontálního převrácení obrázku.
- *Mixup* = 0.1 Pravděpodobnost spojení dvou obrázků dohromady.

Kromě toho jsem využil augmentační techniky z knihovny Albumentations, která je integrována přímo do YOLO frameworku, jako je Blur, medianBlur a To-Gray. Celkově trénink trval přibližně 2 hodiny pro 614 obrázků a byl ukončen po 100 epochách.

Třída	Precision	Recall	mAP50	mAP50-95
Všechny	54.2	44.6	49.9	23.7
suk-hniloba	41.2	3.17	22.8	9.4
Suk	72.1	85.2	84.4	36.4
Trhlina - vysusna	71.2	75.2	81	50
Trhlina - drenova	31.3	30.9	21.9	9.63
Hmyz	55.1	28.6	39.3	12.9

Tabulka 7.6: Výsledky modelu YOLO pro boxy s YOLO augmentací (%)

Z tabulky 7.6. Lze vidět, že v porovnání s ostatními YOLO modely viz. sekce 7.2.2 a 7.2.1 dosahuje tento model nejlepších výsledků u všech tříd. Stále přetrvává problém s detekcí třídy "suk-hniloba", tedy kvůli nízkému Recallu vím, že se hodně instancí této třídy vůbec nedetekuje.

Třída	Precision	Recall	mAP50	mAP50-95
Všechny	56.7	44	50.7	18.3
suk-hniloba	64.7	4.98	35.2	8.9
Suk	74.3	87.7	88.1	31.5
Trhlina - vysusna	72.9	76.9	80.7	36.1
Trhlina - drenova	27.6	27.3	18.7	4.21
Hmyz	44.1	22.9	30.7	10.6

Tabulka 7.7: Výsledky modelu YOLO pro segmentaci s YOLO augmentací (%)

Stejný závěr vyplývá i z tabulky 7.7. Lze zde pozorovat zlepšení v segmentaci, zejména u tříd suků a trhlin - vysušných, což poukazuje na použitelnost modelu v těchto případech.

Z obrázku 7.1 je patrné, že i přesto, že je bounding box správně predikován pro větší trhliny napravo, segmentace není kompletní. YOLO má obecně potíže se segmentací trhlin. Naopak u suků, které zabírají větší plochu, není problém se segmentací, ale spíše s tím, že každý z těchto suků obsahuje určitou část hniloby, která je detekována pouze v některých případech. To přesně ilustruje problém, který je popsán v tabulkách s výsledky detekčního modelu.



- $Flipud = 0.5$  Pravděpodobnost, že se obrázek převrátí podél horizontální osy.
- $Fliplr = 0.5$  Pravděpodobnost, že se obrázek převrátí podél vertikální osy.
- $Mixup = 0.1$  Pravděpodobnost, že se aplikuje technika mixup, která kombinuje dva obrázky dohromady.

Model jsem trénoval po dobu 150 epoch, což trvalo přibližně 9 hodin.

Třída	Precision	Recall	mAP50	mAP50-95
Všechny	55.1	41.1	47.4	25.5
suk-hniloba	0	0	0	0
Suk	90.3	54.2	73.3	35.9
Trhlina - vysusna	84.7	85.2	88.5	57.5
Trhlina - drenova	36.1	37.8	30.5	17
Hmyz	64.3	28.2	44.6	17.4

Tabulka 7.8: Výsledky modelu YOLO pro boxy s kombinovanou augmentací (%)

Z tabulky 7.8 je patrný podobný trend jako v sekci 7.2.2 pro "suk-hniloba". Kvůli její nízké četnosti v trénovacích datech není model schopen ji spolehlivě detekovat. Nicméně oproti ostatním modelům má lepší schopnost detekce trhlín.

Třída	Precision	Recall	mAP50	mAP50-95
Všechny	53.6	38.8	44.4	17.3
suk-hniloba	0	0	0	0
Suk	86.4	50.3	69.1	29
Trhlina - vysusna	86.7	85.2	88.7	39.2
Trhlina - drenova	38.3	37.1	28.6	8.2
Hmyz	56.7	21.1	35.4	10.1

Tabulka 7.9: Výsledky modelu YOLO pro segmentaci s kombinovanou augmentací (%)

Stejné chování modelu je i pro segmentaci viz. sekce 7.9 jen s horší přesností pro jednotlivé defekty.

## 7.2.5 Shrnutí YOLO modelů

V této sekci prezentuji shrnutí výsledků detekčních modelů YOLO pro různé experimenty a varianty modelů. Níže uvedené tabulky 7.10 a 7.11 obsahují hodnoty precision, recallu a průměrné precision při IoU 0.5 až 0.95 pro všechny třídy objektů a pro všechny modely za účelem snadnější čitelnosti a porovnání.

Model YOLO	Precision	Recall	mAP50	mAP50-95
bez augmentace	47.4	28.7	30.7	13.4
vlastní augmentace	41.4	35.7	38.5	21.2
kombinována augmentace	55.1	41.1	47.4	25.5
augmentace	54.2	44.6	49.9	23.7

Tabulka 7.10: Shrnutí výsledků detekčních modelů YOLO pro detekci bounding boxů (%)

Model YOLO	Precision	Recall	mAP50	mAP50-95
bez augmentace	32.1	31.2	28.1	10.1
vlastní augmentace	42.0	37.0	39.8	16.6
kombinována augmentace	53.6	38.8	44.4	17.3
augmentace	56.7	44.0	50.7	18.3

Tabulka 7.11: Shrnutí výsledků detekčních modelů YOLO pro segmentaci (%)

S každým modelem lze pozorovat určité zlepšení ve výkonnosti detekce objektů. Nejvýraznější zlepšení lze však sledovat u posledního modelu ze sekce 7.2.3, který kombinuje různé augmentační techniky a nezaměřuje se čistě na jednu z nich. Avšak i přes zlepšení výkonu jsou stále zjevné oblasti, kde lze provést další optimalizaci, zejména je třeba zdokonalit schopnost modelů identifikovat všechny objekty ve scéně, což může být klíčové pro úspěšnou detekci ve složitějších prostředích. Vzhledem k tomu, že YOLO augmentace dosáhla lepších výsledků než ostatní modely, rozhodl jsem se kombinovat model s mou vlastní augmentací a YOLO augmentací. Nicméně, i když výsledky byly porovnatelné, samotná YOLO augmentace na normálním trénovacím datasetu dosahuje lepšího výsledku viz tabulka 7.11.

## 7.3 Výsledky neuronové sítě Mask R-CNN

V této sekci prezentuji výsledky experimentů s neuronovou sítí Mask R-CNN, implementovanou pomocí frameworku Detectron2. Pro dosažení těchto výsledků používám architekturu *mask\_rcnn\_R\_101\_FPN\_3x*, která je jednou z variant modelu Mask R-CNN a využívá ResNet-101 jako základní konvoluční síť s Feature Pyramid Network (FPN) pro detekci a segmentaci objektů.

Každý model je trénován a testován v několika variantách, které zahrnují výsledky bez použití augmentace, výsledky s vlastní augmentací a výsledky s kombinací augmentačních technik dostupných v Detectronu2. Při každém tréninku importuji příslušný konfigurační soubor pro danou architekturu a testuji různé kombinace parametrů a hyperparametrů. Trénink probíhal na grafické kartě GeForce GTX 1650 s 4GB paměti a podporou CUDA verze 11.8.



Všechny použité datasey byly převedeny do formátu COCO, což mi umožnilo využít standardní metriky jako COCO evaluace pro hodnocení výsledků detekce a segmentace. Tyto metriky byly automaticky vyhodnoceny pomocí frameworku Detectron2, který využívá knihovnu pycocotools pro evaluaci v rámci COCO datasetu.

### 7.3.1 COCO metrika

Pro vyhodnocení výkonu modelů na COCO datasetu se využívá celkem 12 metrik. Průměrná přesnost (AP) je klíčovou metrikou, která měří úspěšnost detekce při různých prahových hodnotách IoU (Intersection over Union). Průměrný recall (AR) je další důležitou metrikou, která hodnotí schopnost detektoru najít všechny objekty v obrazech. Tyto metriky jsou blíže vysvětleny v sekci 6.1.3. COCO je ale dále rozděluje podle velikosti objektů na malé, střední a velké.

V tabulkách budu používat průměrnou přesnost s IoU 50-95 dále jen AP. Pak metriky podle velikosti defektu APs (small při IoU 50-95) a APm (medium při IoU 50-95). Large využívat nebudu, jelikož tak velké defekty dataset neobsahuje. Pro upřesnění malé objekty jsou do  $32^2$  pixelů, střední od  $32^2$  do  $96^2$  a velký nad  $96^2$ . Recall v tabulkách budu uvádět jako AR, což znamená průměrný recall od IoU 50-95 při maximálním počtu detekcí 100 instancí.

### 7.3.2 Mask R-CNN bez augmentace

Během trénování pomocí Mask R-CNN jsem využil trénovací dataset, s parametry 15000 iterací s počáteční rychlostí učení 0.0025. Warm-up fáze, která trvala 2000 iterací, byla implementována k dynamickému přizpůsobení rychlosti učení. Po dokončení této fáze byla následně zachována nejlepší dosažená hodnota rychlosti učení po zbytek tréninku.

Třída	AP	AP50	APs	APm	AR	ARs	ARm
Všechny	6.7	15.2	11.7	1.5	23.9	25.4	2.4

Tabulka 7.12: Výsledky Mask R-CNN pro bounding boxy

Výsledky v obou tabulkách 7.12 a 7.13 by mohly být dále vylepšeny, 15000 iterací byla optimální doba pro získání těchto hodnot při vyšším počtu iterací, jako je například 50000 model vykazoval horších výsledků a téměř žádných detekcí. Nejlepší precision byla dosažena při detekci trhlin s úspěšností mezi 9 a 12%. Trénování tohoto modelu trvalo přibližně 2 hodiny.

Třída	AP	AP50	APs	APm	AR	ARs	ARm
Všechny	4.1	10.7	5.3	1.7	14.8	18.3	2.6

Tabulka 7.13: Výsledky Mask R-CNN pro segmentaci (%)

### 7.3.3 Mask R-CNN s vlastní augmentací

Model byl natrénován pomocí 3000 vygenerovaných augmentovaných obrázků 6.2, které byly převedeny do formátu COCO. Augmentovaný dataset byl kombinován s trénovacím datasetem obsahujícím 614 obrázků. Následně byl použit skript k oddělení 20% dat pro validační část. Model byl trénován na 35000 iteracích s rychlosti učení 0.00025. Warm-up fáze byla nastavena na 5000 iterací. Model se trénoval přibližně 9 hodin.

Třída	AP	AP50	APs	APm	AR	ARs	ARm
Všechny	15.4	28.1	19.6	20	45.7	46.5	55.6

Tabulka 7.14: Výsledky Mask R-CNN pro bounding boxy s vlastní augmentací (%)

Z výsledků uvedených v tabulce 7.14 je patrné, že model Mask R-CNN vykazuje vysoký počet detekcí, avšak z bližšího zkoumání plyne, že mnohé z těchto detekcí jsou falešně pozitivní. Tento jev lze přičíst nadměrné citlivosti modelu na drobné detaily, které interpretuje jako hmyz, ačkoliv se ve skutečnosti jedná o artefakty. Naopak, detekce větších defektů, jako jsou trhliny rozprostírající se přes celý objekt jsou často vynechané. Precision klasifikace tříd trhlín a suků se pohybovala okolo 20%, zatímco třída hmyzu dosahovala precision pouze 4%.

Třída	AP	AP50	APs	APm	AR	ARs	ARm
Všechny	7.7	21.0	8.0	23.2	25.6	26.2	53.3

Tabulka 7.15: Výsledky Mask R-CNN pro segmentaci s vlastní augmentací (%)

Obecně lze říci, že rozšíření datasetu o augmentované obrázky pomohlo modelu zlepšit výkon oproti stavu bez augmentace, jak je patrné v tabulce 7.15.

### 7.3.4 Mask R-CNN s augmentací

V této sekci jsem využil implementaci augmentace pomocí vestavěných nástrojů v Detectronu. Tato augmentace zahrnovala následující metody:

- *RandomSaturation*[0.9, 1.2] Náhodně upravuje saturaci obrázků v rozmezí od 0.9 do 1.2.

- *RandomContrast*[0.9, 1.2] Náhodně upravuje kontrast obrázků v rozmezí od 0.9 do 1.2.
- *RandomHorizontalFlip* = 0.5 S pravděpodobností 50 % provede horizontální zrcadlení obrázků.
- *RandomVerticalFlip* = 0.5 S pravděpodobností 50 % provede vertikální zrcadlení obrázků.
- *RandomRotation*[0,90] Náhodně rotuje obrázky o úhel v rozmezí od 0 do 90 stupňů.

Pro trénování modelu jsem zvolil 20 000 iterací s rychlosti učení 0.0025 a nechal jsem zbytek nastavení podle konfigurace modelu, na kterém jsem prováděl trénink. Celkově se model trénoval přibližně 8 hodin, což bylo zčásti způsobeno rozsáhlými augmentacemi dat. Následující tabulky shrnují výsledky tréninku modelu.

Třída	AP	AP50	APs	APm	AR	ARs	ARm
Všechny	13.7	37.7	11.6	26.5	23.2	17.4	53.9

Tabulka 7.16: Výsledky Mask R-CNN pro bounding boxy s Detectron augmentací (%)

Ve srovnání s ostatními modely uvedenými v sekci 7.3.2 a 7.2.2, tento model dosáhl nejvyšší precision detekce. Začal detekovat trhliny ve větším množství, avšak často zaměňoval trhliny výsušné za trhliny dřevové, což vedlo ke snížení přesnosti detekce dřevových trhlín na 5%, zatímco trhliny výsušné dosáhly precision 29%. Precision detekce suků byla 22%, ostatní třídy měly precision nižší než 10%. Tyto výsledky pro bounding boxy jsou shrnuty v tabulce 7.16.

Třída	AP	AP50	APs	APm	AR	ARs	ARm
Všechny	12.0	36.9	7.8	25.5	18.3	13.0	30.2

Tabulka 7.17: Výsledky Mask R-CNN pro segmentaci s Detectron augmentací (%)

Při segmentaci z tabulky 7.17, dosáhl nejlepší přesnosti Suk 31% a trhlina výsušná s přesností 23%, ostatní třídy měli přesnost pod 10%.

## 7.4 Převod YOLO metrik na COCO

V této sekci jsem provedl transformaci YOLO metrik na COCO metriky pro usnadnění porovnání YOLO a Mask R-CNN modelů. Při porovnání s Mask R-CNN, který

používá COCO metriky, je důležité použít stejný standard pro spravedlivé srovnání výsledků.

Pro dosažení tohoto cíle jsem nejprve vygeneroval predikce na testovacím datasetu pomocí YOLO a uložil je do JSON struktury. Následně jsem provedl úpravu těchto predikcí tak, aby odpovídaly formátu COCO datasetu. Konkrétně jsem převedl identifikátory obrázků z názvů .png souborů na jejich číselné ID, které odpovídají testovacímu datasetu COCO.

V dalším kroku jsem upravil identifikátory tříd o 1, protože COCO indexuje třídy od jedničky, zatímco YOLO od nuly. Poté jsem využil knihovnu *pycocotools* k načtení testovacího COCO datasetu, který odpovídá obsahu YOLO datasetu, a použil jsem upravené predikce k vytvoření COCO evaluace.

Rozdíl v metrikách mezi evaluátory COCO a YOLO může být způsoben několika faktory. Evaluátor COCO využívá externích evaluačních metrik a knihovny *pycocotools*, zatímco YOLO provádí evaluaci pomocí interních evaluačních metod. Kromě toho se různé evaluační skripty mohou lišit v nastavení parametrů, což může různě ovlivnit výsledky. I přesto, že oba evaluátory pracují s podobnými principy, tyto faktory mohou způsobit rozdílné výsledky při porovnání metrik.

## 7.5 Srovnání přesnosti neuronových sítí

V této sekci prezentuji výsledky všech modelů pro bounding boxy v tabulce 7.18 a pro segmentaci v tabulce 7.19.

<b>Model YOLO</b>	<b>AP</b>	<b>AP50</b>	<b>APs</b>	<b>APm</b>	<b>AR</b>	<b>ARs</b>	<b>ARm</b>
bez augmentace	8.3	17.6	12.4	2.5	13.9	16.4	4.6
vlastní augmentace	16.4	29.4	19.9	17.7	21.2	22.6	23.6
kombinovaná augmentace	18.7	36.5	16.4	40.7	24.2	18.9	50.4
augmentace	17.8	39.0	20.1	22.4	23.1	22.1	28.0
<b>Model Mask R-CNN</b>							
bez augmentace	6.7	15.2	11.7	1.5	23.9	25.4	2.4
vlastní augmentace	15.4	28.1	19.6	20	45.7	46.5	55.6
augmentace	13.7	37.7	11.6	26.5	23.2	17.4	53.9

Tabulka 7.18: Výsledky všech modelů ve formátu COCO metrik pro detekci bounding boxů (%)

Z tabulek 7.18 a 7.19 lze vyčíst, že mezi nejlepší modely patří YOLO s kombinovanou augmentací a YOLO s vlastní augmentací. Model YOLO s kombinovanou augmentací dosáhl nejlepších výsledků s průměrným skóre z oblastí (APm) pro bounding boxy a model YOLO s vlastní augmentací vykazuje nejlepší výsledky pro segmentaci defektů.

<b>Model YOLO</b>	<b>AP</b>	<b>AP50</b>	<b>APs</b>	<b>APm</b>	<b>AR</b>	<b>ARs</b>	<b>ARm</b>
bez augmentace	4.7	17.1	7.2	2.4	8.5	10.2	4.0
vlastní augmentace	10.6	28.0	14.3	14.6	15.2	14.3	21.1
kombinována augmentace	11.8	34.1	9.8	25.9	16	11.4	38.2
augmentace	11.1	35.3	11.0	21.0	14.5	12.4	25.2
<b>Model Mask R-CNN</b>							
bez augmentace	4.1	10.7	5.3	1.7	14.8	18.3	2.6
vlastní augmentace	7.7	21.0	8.0	23.2	25.6	26.2	53.3
augmentace	12.0	36.9	7.8	25.5	18.3	13.0	30.2

Tabulka 7.19: Výsledky všech modelů ve formátu COCO metrik pro segmentaci (%)

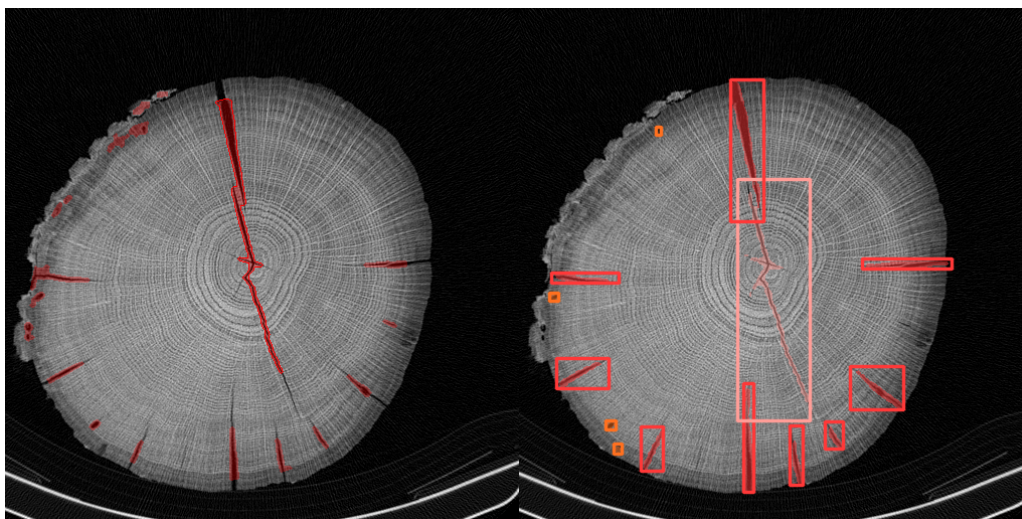
Je však důležité poznamenat, že model YOLO s kombinovanou augmentací není schopen správně detekovat jednu z tříd kvůli jejímu malému výskytu v trénovacích datech. Tento fakt naznačuje, že i přes lepší výsledky celkově, tento model není vhodný pro detekci této konkrétní třídy.

Pokud jde o model Mask R-CNN, přestože má nízké hodnoty AP, je důležité si všimnout, že model YOLO dosahuje podobných výsledků s lepšími hodnotami Recallu. I přes předpokládanou výhodu Mask R-CNN, model YOLO je schopný dosáhnout srovnatelných výsledků, což může být důsledkem nerovnoměrného rozložení defektů mezi třídami. Předpokládám, že nízké hodnoty modelů jsou částečně způsobeny nedostatečným počtem trénovacích dat a také složitostí hledaných defektů.

## 7.6 Srovnání výsledků vlastní metody a neuronové sítě

Má metoda pro segmentaci trhlin dosahuje dobré úspěšnosti detekce, jak je patrné z tabulky 7.1, kde pro IoU 0.25 dosahuje kolem 50% úspěšnosti. Je však důležité poznamenat, že s rostoucím IoU klesá přesnost detekce. Zlepšení přesnosti detekce může být dosaženo zjemněním detekčního procesu. Avšak to může způsobit zvýšení počtu falešně pozitivních detekcí (FP). Proto je vhodné hledat vyvážený přístup mezi těmito parametry, i když to může vést k mírnému snížení celkové přesnosti detekce.

Naopak, neuronové sítě vykazují menší celkovou úspěšnost detekce, ale pokud se zaměřím pouze na detekci trhlin a neberu v úvahu další podtřídy, je jejich přesnost srovnatelná nebo dokonce vyšší, zejména u neuronové sítě YOLO, jak je vidět z tabulky 7.8, kde dosahuje 85% úspěšnosti detekce trhlin. Při klasifikaci trhlin do dvou tříd - dřevové a výsušné - se často u neuronových sítí vyskytovalo zaměňování těchto kategorií. Sjednocení těchto trhlin do jedné třídy by pravděpodobně zvýšilo úspěšnost detekce.



Obrázek 7.2: Ukázka segmentace vlevo a predikce modelu YOLO napravo

Z obrázku 7.2 lze pozorovat, že segmentační metoda vlevo úspěšně detekuje trhliny ve snímku. Nicméně je patrné, že má tendenci generovat relativně vysoký počet falešně pozitivních detekcí v oblasti kůry. Na druhé straně, vpravo je zobrazen výstup natrénovaného modelu, který přináší kvalitní a přesné predikce trhlín. Barevné kódování tříd v pravé části obrázku je následující: červená barva označuje trhliny - výsušné, růžová barva označuje trhliny - dřevné a oranžová barva značí detekci dřevokazného hmyzu. Predikce na obrázku 7.2 je z YOLO modelu popsáném v sekci 7.2.4.

V rámci této bakalářské práce byla provedena analýza CT snímků dřeva s cílem vyvinout spolehlivý systém detekce defektů, který by nahradil manuální proces prováděný lidskou pracovní silou v dřevařském průmyslu. Tato automatizace má potenciál zvýšit výtěžnost suroviny díky přesné lokalizaci defektů.

V práci jsem se zaměřil na tvorbu vlastní segmentační metody pro detekci trhlin v dřevě. Výsledná metoda dokáže detekovat a polygonem segmentovat defekty pomocí tradičních metod počítačového vidění. Využití této segmentační metody umožnilo vytvoření anotací, které byly ručně doplněny a upraveny do formy polygonů ohraničujících defektů a klasifikovány do odpovídajících tříd.

Zpracování datasetů a konverze mezi formáty byly klíčovou součástí práce, neboť byly nezbytné pro kompatibilitu s různými druhy neuronových sítí. Každá síť vyžadovala specifický formát datasetu: Mask R-CNN využívala formát COCO, zatímco síť YOLO používala jednoduchý formát YOLO.

Vytvořené datasety byly následně použity k natrénování, již zmíněných dvou neuronových sítí. Experimenty provedené s různými augmentačními metodami a jejich kombinacemi odhalily, že výsledky neuronových sítí byly mírně nižší, než se původně předpokládalo. Přesto se ukázalo, že síť YOLO dosahuje lepších výsledků než síť Mask R-CNN. Věřím, že s větším množstvím trénovacích dat a optimalizací volby rozsahu tříd pro detekci by mohly neuronové sítě dosáhnout ještě lepších výsledků.

Tato práce představuje důležitý krok v oblasti detekce defektů v dřevařském průmyslu. Doufám, že výsledky a poznatky z této práce přispějí k dalšímu rozvoji a efektivitě dřevařského průmyslu a zároveň poslouží, jako základ pro budoucí rozsáhlejší výzkum v této oblasti.





# Seznam obrázků

2.1	Příklad trhlin ve struktuře dřeva na CT snímku . . . . .	5
2.2	Příklad suků a trhlin dřeva na CT snímku . . . . .	6
2.3	Detail na hmyzí stopy na dřevě . . . . .	7
3.1	Princip CT snímku . . . . .	11
4.1	Porovnání původního obrázku a obrázku po aplikaci Gaussova filtru .	14
4.2	Porovnání detekce hran s a bez použití Gaussova filtrování . . . . .	14
4.3	Gaussův filtr, histogram a Otsuovo prahování . . . . .	15
5.1	Ukázka tvorby anotací z prostředí Supervisely . . . . .	19
5.2	Četnost výskytu tříd v trénovacím datasetu . . . . .	20
5.3	Četnost výskytu tříd v testovacím datasetu . . . . .	21
5.4	Četnost výskytu tříd v augmentovaném datasetu . . . . .	22
6.1	Struktura jednorůchodového modelu YOLO . . . . .	24
6.2	Mask R-CNN framework pro segmentaci objektů. . . . .	25
6.3	Vizualizace překryvu u IoU . . . . .	26
6.4	Proces augmentace od počátečního pozadí (vlevo) přes plánované polygony s umístěnými defekty (uprostřed) až po konečný augmentovaný obrázek (vpravo) . . . . .	29
7.1	Ukázka predikce modelu YOLO s augmentací . . . . .	36
7.2	Ukázka segmentace vlevo a predikce modelu YOLO napravo . . . . .	44



# Seznam tabulek

7.1	Výsledky segmentace trhlin s použitím různých prahů IoU (%) . . . . .	32
7.2	Výsledky modelu YOLO pro bounding boxy (%) . . . . .	32
7.3	Výsledky modelu YOLO pro segmentaci (%) . . . . .	33
7.4	Výsledky modelu YOLO pro bounding boxy s augmentací (%) . . . . .	34
7.5	Výsledky modelu YOLO pro segmentaci s augmentací (%) . . . . .	34
7.6	Výsledky modelu YOLO pro boxy s YOLO augmentací (%) . . . . .	35
7.7	Výsledky modelu YOLO pro segmentaci s YOLO augmentací (%) . . .	35
7.8	Výsledky modelu YOLO pro boxy s kombinovanou augmentací (%) . .	37
7.9	Výsledky modelu YOLO pro segmentaci s kombinovanou augmentací (%)	37
7.10	Shrnutí výsledků detekčních modelů YOLO pro detekci bounding boxů (%) . . . . .	38
7.11	Shrnutí výsledků detekčních modelů YOLO pro segmentaci (%) . . . .	38
7.12	Výsledky Mask R-CNN pro bounding boxy . . . . .	39
7.13	Výsledky Mask R-CNN pro segmentaci (%) . . . . .	40
7.14	Výsledky Mask R-CNN pro bounding boxy s vlastní augmentací (%) .	40
7.15	Výsledky Mask R-CNN pro segmentaci s vlastní augmentací (%) . . . .	40
7.16	Výsledky Mask R-CNN pro bounding boxy s Detectron augmentací (%)	41
7.17	Výsledky Mask R-CNN pro segmentaci s Detectron augmentací (%) . .	41
7.18	Výsledky všech modelů ve formátu COCO metrik pro detekci bounding boxů (%) . . . . .	42
7.19	Výsledky všech modelů ve formátu COCO metrik pro segmentaci (%) .	43



# Seznam zkratek

<i>AP</i>	Average Precision
<i>AR</i>	Average Recall
<i>BA</i>	Boundary Approximation
<i>CNN</i>	Convolutional Neural Network
<i>COCO</i>	Common Objects in Context
<i>CT</i>	Computed Tomography
<i>CUDA</i>	Compute Unified Device Architecture
<i>DICOM</i>	Digital Imaging and Communications in Medicine
<i>FN</i>	False Negative
<i>FP</i>	False Positive
<i>FPN</i>	Feature Pyramid Network
<i>HT</i>	Hough Transform
<i>IoU</i>	Intersect over union
<i>JSON</i>	JavaScript Object Notation
<i>R – CNN</i>	Region-based Convolutional Neural Network
<i>RLE</i>	Run-Length Encoding
<i>SOP</i>	Service Object Pair
<i>TN</i>	True Negative
<i>TP</i>	True Positive
<i>YOLO</i>	You Only Look Once



# Bibliografie

- [Gaz+20] GAZO, Rado; VANEK, Juraj; ABDUL<sub>M</sub>ASSIH, Michel; BENES, Bedrich. A fast pith detection for computed tomography scanned hardwood logs. *Computers and Electronics in Agriculture*. 2020, roč. 170, s. 105107. ISSN 0168-1699. Dostupné z DOI: <https://doi.org/10.1016/j.compag.2019.105107>.
- [GW08] GONZALEZ, Rafael C.; WOODS, Richard E. *Digital Image Processing*. Prentice Hall, 2008. ISBN 9780131687288. Dostupné také z: <https://books.google.cz/books?id=8uG0njRGEzoC>.
- [He+18] HE, Kaiming; GKIOXARI, Georgia; DOLLÁR, Piotr; GIRSHICK, Ross. *Mask R-CNN*. 2018. Dostupné z arXiv: 1703.06870 [cs.CV].
- [Liu+16] LIU, Wei et al. *SSD: Single shot multibox detector*. 2016. Dostupné také z: <https://arxiv.org/abs/1512.02325>.
- [Lon+12] LONGUETAUD, F. et al. Automatic knot detection and measurements from X-ray CT images of wood: A review and validation of an improved algorithm on softwood samples. *Computers and Electronics in Agriculture*. 2012, roč. 85, s. 77–89. ISSN 0168-1699. Dostupné z DOI: <https://doi.org/10.1016/j.compag.2012.03.013>.
- [Mc08] MASON, Darcy; CONTRIBUTOR, pydicom. *Pydicom* [<https://pydicom.github.io/about>]. 2008. [Accessed 04-10-2023].
- [Mis20] MISHRA, Aditya. *Metrics to evaluate your machine learning algorithm*. Towards Data Science, 2020. Dostupné také z: <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>.
- [Ngu15] NGUYEN, Martin. *Brno University of Technology - Vut*. 2015. Dostupné také z: [https://www.vut.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=174714](https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=174714).
- [REJ14] REJŽEK, Lukáš. *Vady a škůdci dřeva [online]*. 2014. Dostupné také z: <https://theses.cz/id/yp4w94/>. SUPERVISOR: PaedDr. Alena Poláčková, Ph.D.

- [Rez+19] REZATOFIGHI, Hamid et al. Generalized Intersection over Union. 2019.
- [17] *Suky a Trhliny*. 2017. Dostupné také z: <https://www.mezistromy.cz/vlastnosti-dreva-a-drevostaveb/vady-dreva-1-cast-suky-a-trhliny/odborny>.
- [Tan22] TAN, Ren Jie. *Breaking down mean average precision (MAP)*. Medium, 2022. Dostupné také z: <https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52>.
- [97] *US Forest Service*. 1997. Dostupné také z: <https://www.fs.usda.gov/t-d/pubs/pdf/97241202.pdf>.



101011000011100010 1100001  
1010110001 10001 10

110100011101101001 101101  
01100001 101101  
111000101011101